Noisy SMS Machine Translation in Low-Density Languages

Vladimir Eidelman[†], **Kristy Hollingshead**[†], and **Philip Resnik**[†‡]
[†]UMIACS Laboratory for Computational Linguistics and Information Processing
[‡]Department of Linguistics
University of Maryland, College Park
{vlad,hollingk,resnik}@umiacs.umd.edu

Abstract

This paper presents the system we developed for the 2011 WMT Haitian Creole–English SMS featured translation task. Applying standard statistical machine translation methods to noisy real-world SMS data in a low-density language setting such as Haitian Creole poses a unique set of challenges, which we attempt to address in this work. Along with techniques to better exploit the limited available training data, we explore the benefits of several methods for alleviating the additional noise inherent in the SMS and transforming it to better suite the assumptions of our hierarchical phrase-based model system. We show that these methods lead to significant improvements in BLEU score over the baseline.

1 Introduction

For the featured translation task of the Sixth Workshop on Statistical Machine Translation, we developed a system for translating Haitian Creole Emergency SMS messages. Given the nature of the task, translating text messages that were sent during the January 2010 earthquake in Haiti to an emergency response service called Mission 4636, we were not only faced with the problem of dealing with a low-density language, but additionally, with noisy, real-world data in a domain which has thus far received relatively little attention in statistical machine translation. We were especially interested in this task because of the unique set of challenges that it poses for existing translation systems. We focused our research effort on techniques to better utilize the limited available training resources, as well as ways in which we could automatically alleviate and transform the noisy data to our advantage through the use of automatic punctuation prediction, finite-state raw-to-clean transduction, and grammar extraction. All these techniques contributed to improving translation quality as measured by BLEU score over our baseline system.

The rest of this paper is structured as follows. First, we provide a brief overview of our baseline system in Section 2, followed by an examination of issues posed by this task and the steps we have taken to address them in Section 3, and finally we conclude with experimental results and additional analysis.

2 System Overview

Our baseline system is based on a hierarchical phrase-based translation model, which can formally be described as a synchronous context-free grammar (SCFG) (Chiang, 2007). Our system is implemented in cdec, an open source framework for aligning, training, and decoding with a number of different translation models, including SCFGs. (Dyer et al., 2010). [1] SCFG grammars contain pairs of CFG rules with aligned nonterminals, where by introducing these nonterminals into the grammar, such a system is able to utilize both word and phrase level reordering to capture the hierarchical structure of language. SCFG translation models have been shown to produce state-of-the-art translation for most language pairs, as they are capable of both exploiting lexical information for and efficiently computing all possible reorderings using a CKY-based decoder (Dyer et al., 2009).

[1] http://cdec-decoder.org

Proceedings of the 6th Workshop on Statistical Machine Translation, pages 344–350,
Edinburgh, Scotland, UK, July 30–31, 2011.

One benefit of cdec is the flexibility allowed with regard to the input format, as it expects either a string, lattice, or context-free forest, and subsequently generates a hypergraph representing the full translation forest without any pruning. This forest can now be rescored, by intersecting it with a language model for instance, to obtain output translations. These capabilities of cdec allow us to perform the experiments described below, which may have otherwise proven to be quite impractical to carry out in another system.

The set of features used in our model were the rule translation relative frequency $P(e|f)$, a target n-gram language model $P(e)$, lexical translation probabilities $P_{lex}(\overline{e}|\overline{f})$ and $P_{lex}(\overline{f}|\overline{e})$, a count of the total number of rules used, a target word penalty, and a count of the number of times the glue rule is used. The number of non-terminals allowed in a synchronous grammar rule was restricted to two, and the non-terminal span limit was 12 for non-glue grammars. The hierarchical phrase-based translation grammar was extracted using a suffix array rule extractor (Lopez, 2007).

To optimize the feature weights for our model, we used an implementation of the hypergraph minimum error rate training (MERT) algorithm (Dyer et al., 2010; Och, 2003) for training with an arbitrary loss function. The error function we used was BLEU (Papineni et al., 2002), and the decoder was configured to use cube pruning (Huang and Chiang, 2007) with a limit of 100 candidates at each node.

2.1 Data Preparation

The SMS messages were originally translated by English speaking volunteers for the purpose of providing first responders with information and locations requiring their assistance. As such, in order to create a suitable parallel training corpus from which to extract a translation grammar, a number of steps had to be taken in addition to lowercasing and tokenizing both sides of training data. Many of the English translations had additional notes sections that were added by the translator to the messages with either personal notes or further informative remarks. As these sections do not correspond to any text on the source side, and would therefore degrade the alignment process, these had to be identified and removed. Furthermore, the anonymization of the data resulted in tokens such as *firstname* and *phonenumber* which were prevalent and had to be preserved as they were. Since the total amount of Haitian-English parallel data provided is quite limited, we found additional data and augmented the available set with data gathered by the CrisisCommons group and made it available to other WMT participants. The combined training corpus from which we extracted our grammar consisted of 123,609 sentence pairs, which was then filtered for length and aligned using the GIZA++ implementation of IBM Model 4 (Och and Ney, 2003) to obtain one-to-many alignments in either direction and symmetrized using the grow-diag-final-and method (Koehn et al., 2003).

We trained a 5-gram language model using the SRI language modeling toolkit (Stolcke, 2002) from the English monolingual News Commentary and News Crawl language modeling training data provided for the shared task and the English portion of the parallel data with modified Kneser-Ney smoothing (Chen and Goodman, 1996). We have previously found that since the beginnings and ends of sentences often display unique characteristics that are not easily captured within the context of the model, explicitly annotating beginning and end of sentence markers as part of our translation process leads to significantly improved performance (Dyer et al., 2009).

A further difficulty of the task stems from the fact that there are two versions of the SMS test set, a raw version, which contains the original messages, and a clean version which was post-edited by humans. As the evaluation of the task will consist of translating these two versions of the test set, our baseline system consisted of two systems, one built on the clean data using the 900 sentences in SMS dev clean to tune our feature weights, and evaluated using SMS devtest clean, and one built analogously for the raw data tuned on the 900 sentences in SMS dev raw and evaluated on SMS devtest raw. We report results on these sets as well as the 1274 sentences in the SMS test set.

3 Experimental Variation

The results produced by the baseline systems are presented in Table 1. As can be seen, the clean version performs on par with the French-English trans-

BASELINE			
Version	Set	BLEU	TER
clean	dev	30.36	56.04
	devtest	28.15	57.45
	test	27.97	59.19
raw	dev	25.62	63.27
	devtest	24.09	63.82
	test	23.33	65.93

Table 1: Baseline system BLEU and TER scores

lation quality in the 2011 WMT shared translation task,[2] and significantly outperforms the raw version, despite the content of the messages being identical. This serves to underscore the importance of proper post-processing of the raw data in order to attempt to close the performance gap between the two versions. Through analysis of the raw and clean data we identified several factors which we believe greatly contribute to the difference in translation output. We examine punctuation in Section 3.2, grammar post-processing in Section 3.3, and morphological differences in Sections 3.4 and 3.5.

3.1 Automatic Resource Confidence Weighting

A practical technique when working with a low-density language with limited resources is to duplicate the same trusted resource multiple times in the parallel training corpus in order for the translation probabilities of the duplicated items to be augmented. For instance, if we have confidence in the entries of the glossary and dictionary, we can duplicate them 10 times in our training data to increase the associated probabilities. The aim of this strategy is to take advantage of the limited resources and exploit the reliable ones.

However, what happens if some resources are more reliable than others? Looking at the provided resources, we saw that in the Haitisurf dictionary, the entry for *paske* is matched with *for*, while in glossary-all-fix, *paske* is matched with *because*. If we then consider the training data, we see that in most cases, *paske* is in fact translated as *because*. Motivated by this type of phenomenon, we employed an alternative strategy to simple duplication which allows us to further exploit our prior knowledge.

[2]http://matrix.statmt.org/matrix

First, we take the previously word-aligned baseline training corpus and for each sentence pair and word e_i compute the alignment link count $c(e_i, f_j)$ over the positions j that e_i is aligned with, repeating for $c(f_i, e_j)$ in the other direction. Then, we process each resource we are considering duplicating, and augment its score by $c(e_i, f_j)$ for every pair of words which was observed in the training data and is present in the resource. This score is then normalized by the size of the resource, and averaged over both directions. The outcome of this process is a score for each resource. Taking these scores on a log scale and pinning the top score to associate with 20 duplications, the result is a decreasing number of duplications for each subsequent resources, based on our confidence in its entries. Thus, every entry in the resource receives credit, as long as there is evidence that the entries we have observed are reliable. On our set of resources, the process produces a score of 17 for the Haitisurf dictionary and 183 for the glossary, which is in line with what we would expect. It may be that the resources may have entries which occur in the test set but not in the training data, and thus we may inadvertently skew our distribution in a way which negatively impacts our performance, however, overall we believe it is a sound assumption that we should bias ourselves toward the more common occurrences based on the training data, as this should provide us with a higher translation probability from the *good* resources since the entries are repeated more often. Once we obtain a proper weighting scheme for the resources, we construct a new training corpus, and proceed forward from the alignment process.

Table 2 presents the BLEU and TER results of the standard strategy of duplication against the confidence weighting scheme outlined above. As can be

		CONF. WT.		X10	
Version	Set	BLEU	TER	BLEU	TER
clean	dev	30.79	55.71	30.61	55.31
	devtest	27.92	57.66	28.22	57.06
	test	27.97	59.65	27.74	59.34
raw	dev	26.11	62.64	25.72	62.99
	devtest	24.16	63.71	24.18	63.71
	test	23.66	65.69	23.06	66.78

Table 2: Confidence weighting versus x10 duplication

seen, the confidence weighting scheme substantially outperforms the duplication for the dev set of both versions, but these improvements do not carry over to the clean devtest set. Therefore, for the rest of the experiments presented in the paper, we will use the confidence weighting scheme for the raw version, and the standard duplication for the clean version.

3.2 Automatic Punctuation Prediction

Punctuation does not usually cause a problem in text-based machine translation, but this changes when venturing into the domain of SMS. Punctuation is very informative to the translation process, providing essential contextual information, much as the aforementioned sentence boundary markers. When this information is lacking, mistakes which would have otherwise been avoided can be made. Examining the data, we see there is substantially more punctuation in the clean set than in the raw. For example, there are 50% more comma's in the clean dev set than in the raw. A problem of lack of punctuation has been studied in the context of spoken language translation, where punctuation prediction on the source language prior to translation has been shown to improve performance (Dyer, 2007). We take an analogous approach here, and train a hidden 5-gram model using SRILM on the punctuated portion of the Haitian side of the parallel data. We then applied the model to punctuate the raw dev set, and tuned a system on this punctuated set. However, the translation performance did not improve. This may have been do to several factors, including the limited size of the training set, and the lack of in-domain punctuated training data. Thus, we applied a self-training approach. We applied the punctuation model to the SMS training data, which is only available in the raw format. Once punctuated, we re-trained our punctuation prediction model, now including the automatically punctuated SMS data as part of the punctuation language model training data. We use this second punctuation prediction model to predict punctuation for the tuning and evaluation sets. We continue by creating a new parallel training corpus which substitutes the original SMS training data with the punctuated version, and build a new translation system from it. The results from using the self-trained punctuation method are presented in Table 3. Future experiments on the raw version are performed using this punctuation.

AUTO-PUNC			
Version	Set	BLEU	TER
raw	dev	26.09	62.84
	devtest	24.38	64.26
	test	23.59	65.91

Table 3: Automatic punctuation prediction results

3.3 Grammar Filtering

Although the grammars of a SCFG model permit high-quality translation, the grammar extraction procedure extracts many rules which are formally licensed by the model, but are otherwise incapable of helping us produce a good translation. For example, in this task we know that the token *firstname* must always translate as *firstname*, and never as *phonenumber*. This refreshing lack of ambiguity allows us to filter the grammar after extracting it from the training corpus, removing any grammar rule where these conditions are not met, prior to decoding. Filtering removed approximately 5% of the grammar rules.[3] Table 4 shows the results of applying grammar filtering to the raw and clean version.

GRAMMAR			
Version	Set	BLEU	TER
clean	dev	30.88	54.53
	devtest	28.69	56.21
	test	28.29	58.78
raw	dev	26.41	62.47
	devtest	24.47	63.26
	test	23.96	65.82

Table 4: Results of filtering the grammar in a post-processing step before decoding

3.4 Raw-Clean Segmentation Lattice

As noted above, a major cause of the performance degradation from the clean to the raw version is related to the morphological errors in the messages. Figure 1 presents a segmentation lattice with two versions of the same sentence; the first being from

[3]We experimented with more aggressive filtering based on punctuation and numbers, but translation quality degraded rapidly.

the raw version, and the second from the clean. We can see that that *Ilavach* has been broken into two segments, while *ki sou* has been combined into one.

Since we do not necessarily know in advance which segmentation is the correct one for a better quality translation, it may be of use to be able to utilize both segmentations and allow the decoder to learn the appropriate one. In previous work, word segmentation lattices have been used to address the problem of productive compounding in morphologically rich languages, such as German, where morphemes are combined to make words but the orthography does not delineate the morpheme boundaries. These lattices encode alternative ways of segmenting compound words, and allow the decoder to automatically choose which segmentation is best for translation, leading to significantly improved results (Dyer, 2009). As opposed to building word segmentation lattices from a linguistic morphological analysis of a compound word, we propose to utilize the lattice to encode all alternative ways of segmenting a word as presented to us in either the clean or raw versions of a sentence. As the task requires us to produce separate clean and raw output on the test set, we tune one system on a lattice built from the clean and raw dev set, and use the single system to decode both the clean and raw test set separately. Table 5 presents the results of using segmentation lattices.

3.5 Raw-to-Clean Transformation Lattice

As can be seen in Tables 1, 2, and 3, system performance on clean text greatly outperforms system performance on raw text, with a difference of almost 5 BLEU points. Thus, we explored the possibility of automatically transforming raw text into clean text, based on the "parallel" raw and clean texts that were provided as part of the task.

One standard approach might have been to train a Haitian-to-Haitian MT system to "translate" from raw text to clean text. However, since the training set was only available as raw text, and only the dev and devtest datasets had been cleaned, we clearly did not have enough data to train a raw-to-clean translation system. Thus, we created a finite-state transducer (FST) by aligning the raw dev text to the clean dev text, on a sentence-by-sentence basis. These raw-to-clean alignments were created using a simple minimum edit distance algorithm; substitution costs were calculated according to orthographic match.

SEG-LATTICE			
Version	Set	BLEU	TER
raw	dev	26.17	61.88
	devtest	24.64	62.53
	test	23.89	65.27

Table 5: Raw-Clean segmentation lattice tuning results

FST-LATTICE			
Version	Set	BLEU	TER
raw	dev	26.20	62.15
	devtest	24.21	63.45
	test	22.56	67.79

Table 6: Raw-to-clean transformation lattice results

One option would be to use the resulting raw-to-clean transducer to greedily replace each word (or phrase) in the raw input with the predicted transformation into clean text. However, such a destructive replacement method could easily introduce cascading errors by removing text that might have been translated correctly. Fortunately, as mentioned in Section 2, and utilized in the previous section, the cdec decoder accepts lattices as input. Rather than replacing raw text with the predicted transformation into "clean" text, we add a path to the input lattice for each possible transform, for each word and phrase in the input. We tune a system on a lattice built from this approach on the dev set, and use the FST developed from the dev set in order to create lattices for decoding the devtest and test sets. An example is shown in Figure 3.4. Note that in this example, the transformation technique correctly inserted new paths for *ilavach* and *ki sou*, correctly retained the single path for *zile*, but overgenerated many (incorrect) options for *nan*. Note, though, that the original path for *nan* remains in the lattice, delaying the ambiguity resolution until later in the decoding process. Results from creating raw-to-clean transformation lattices are presented in Table 6.

By comparing the results in Table 6 to those in Table 5, we can see that the noise introduced by the finite-state transformation process outweighed the

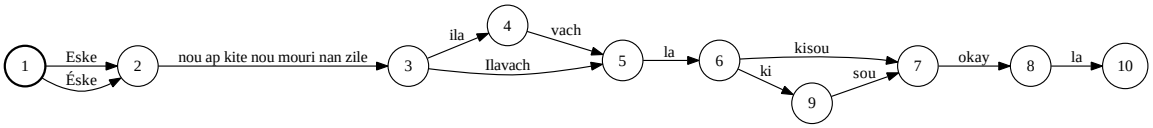

Figure 1: Partial segmentation lattice combining the raw and clean versions of the sentence: *Are you going to let us die on Ile à Vaches which is located close the city of Les Cayes.*

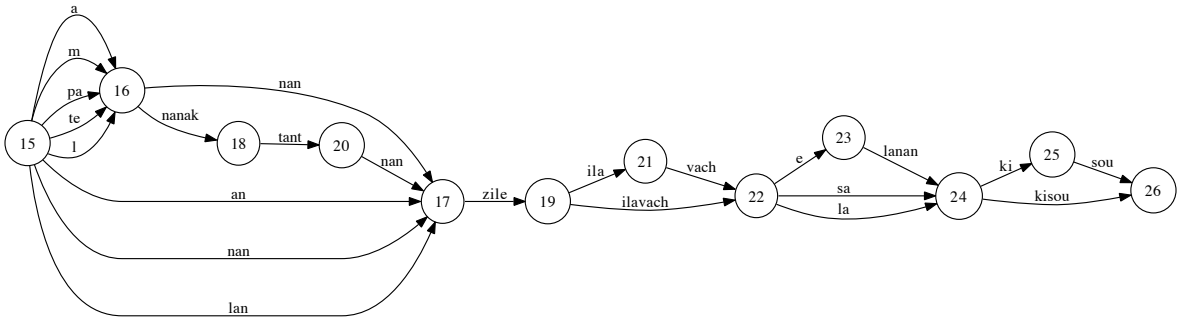

Figure 2: Partial input lattice for sentence in Figure 3.4, generated using the raw-to-clean transform technique described in Section 3.5.

gains of adding new phrases for tuning.

4 System Comparison

Table 7 shows the performance on the devtest set of each of the system variations that we have presented in this paper. From this table, we can see that our best-performing system on clean data was the GRAMMAR system, where the training data was multiplied by ten as described in Section 3.1, then the grammar was filtered as described in Section 3.3. Our performance on clean test data, using this system, was 28.29 BLEU and 58.78 TER. Table 7 also demonstrates that our best-performing system on raw data was the SEG-LATTICE system, where the training data was confidence-weighted (Section 3.1), the grammar was filtered (Section 3.3), punctuation was automatically added to the raw data as described in Section 3.2, and the system was tuned on a lattice created from the raw and clean dev dataset. Our performance on raw test data, using this system, was 23.89 BLEU and 65.27 TER.

5 Conclusion

In this paper we presented our system for the 2011 WMT featured Haitian Creole–English translation task. In order to improve translation quality of low-density noisy SMS data, we experimented with a number of methods that improve performance on both the clean and raw versions of the data, and help close the gap between the post-edited and real-world data according to BLEU and TER evaluation. The methods employed were developed to specifically address shortcomings we observed in the data, such as segmentation lattices for morphological ambiguity, confidence weighting for resource utilization, and punctuation prediction for lack thereof. Overall, this work emphasizes the feasibility of adapting existing translation technology to as-yet underexplored domains, as well as the shortcomings that need to be addressed in future work in real-world data.

System	clean		raw	
	BLEU	TER	BLEU	TER
BASELINE	28.15	57.45	24.09	63.82
CONF. WT.	27.92	57.66	24.16	63.71
X10	28.22	57.06	24.18	63.71
GRAMMAR	**28.69**	**56.21**	24.47	63.26
AUTO-PUNC	–	–	24.38	64.26
SEG-LATTICE	–	–	**24.64**	**62.53**
FST-LATTICE	–	–	24.21	63.45

Table 7: Comparison of all systems' performance on devtest set

6 Acknowledgments

The authors gratefully acknowledge partial support from the DARPA GALE program, No. HR0011-06-2-001. In addition, the first author was supported by the NDSEG Fellowship. Any opinions or findings do not necessarily reflect the view of the sponsors.

References

Stanley F. Chen and Joshua Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 310–318.

David Chiang. 2007. Hierarchical phrase-based translation. In *Computational Linguistics*, volume 33(2), pages 201–228.

Chris Dyer, Hendra Setiawan, Yuval Marton, and Philip Resnik. 2009. The University of Maryland statistical machine translation system for the Fourth Workshop on Machine Translation. In *Proceedings of the EACL-2009 Workshop on Statistical Machine Translation*.

Chris Dyer, Adam Lopez, Juri Ganitkevitch, Jonathan Weese, Ferhan Ture, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of ACL System Demonstrations*.

Chris Dyer. 2007. The University of Maryland Translation system for IWSLT 2007. In *Proceedings of IWSLT*.

Chris Dyer. 2009. Using a maximum entropy model to build segmentation lattices for MT. In *Proceedings of NAACL-HLT*.

Liang Huang and David Chiang. 2007. Forest rescoring: Faster decoding with integrated language models. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 144–151.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54.

Adam Lopez. 2007. Hierarchical phrase-based translation with suffix arrays. In *Proceedings of EMNLP*, pages 976–985.

Franz Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. In *Computational Linguistics*, volume 29(21), pages 19–51.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Intl. Conf. on Spoken Language Processing*.