PanDoRA: A Large-scale Two-way Statistical Machine Translation System for Hand-held Devices

Ying Zhang Stephan Vogel

Language Technologies Institute
School of Computer Sciences
Carnegie Mellon University
5000 Forbes Ave. Pittsburgh, PA 15213
U.S.A.
{joy+, vogel+}@cs.cmu.edu

Abstract

The statistical machine translation (SMT) approach has taken a lead place in the field of Machine Translation for its better translation quality and lower cost in training compared to other approaches. However, due to the high demand of computing resources, an SMT system can not be directly run on hand-held devices. Most existing hand-held translation systems are either interlingua-based, which require non-trivial human efforts to write grammar rules, or using the client/server architecture, which are constrained by the availability of wireless connections. In this paper we present PanDoRA, a two-way phrase-based statistical machine translation system for stand-alone hand-held devices. Powered by special designs such as integerized computation and compact data structure, PanDoRA can translate dialogue speech on off-the-shelf PDAs in real time. PanDoRA uses 64K words vocabulary and millions of phrase pairs for each translation directions. To our knowledge, PanDoRA is the first large-scale SMT system with build-in reordering models running on hand-held devices. We have successfully developed several speech-to-speech translation systems using PanDoRA and our experiments show that PanDoRA's translation quality is comparable to that of the state-of-the-art phrase-based statistical machine translation systems such as Pharaoh and STTK.

Introduction

The world today sees great demands of portable translation devices. Speech translation systems running on hand-held devices are of great interest to international tourism, business and humanitarian aids.

Statistical machine translation (SMT), especially the phrase-based SMT has shown great advantages over other MT approaches in recent years for its translation quality and its ease to be adapted to new language pairs and new domains. As an emerging trend, SMT systems have been used in many areas such as webpage translation, live broadcasting translation, lecture translation and so on.

To use an SMT system on a hand-held device, however, is not that easy. SMT systems use large amount of data to train the statistical models. The resulting models could easily go up to several gigabytes when loaded into the memory. Hand-held devices, such as mobile phones and PDAs, have very limited dynamic memory. In addition, the CPUs on most hand-held devices are weak. Their frequencies are less than 1/4 of those used in the regular PCs and they do not have numerical co-processors, which are critical for calculating various probabilities in SMT systems. All these restrictions make it a great challenge to develop a practical SMT system for hand-held devices.

In this paper, we present PanDoRA, a two-way phrase-based SMT system for hand-held devices (Figure 1). PanDoRA has been successfully applied to PDA-based speech-to-speech translation systems for various language pairs, including Arabic/English, Chinese/English, Japanese/English, Spanish/English, and Thai/English, in tourist, medical aid and force protection domains. PanDoRA uses two translation models, one for each translation direction and each can have millions of phrase pairs[1]. On a typical set up, PanDoRA translates a sentence in about 10ms on a PDA. Its translation quality is comparable to the state-of-the-art SMT systems.

Figure 1: PanDoRA system running on a PDA

Vocabulary	Source Language	Up to 64K
	Target Language	Up to 64K
Translation Model	Src. → Tgt. pairs	Up to 4 billion
	Tgt. → Src. pairs	Up to 4 billion
	Uniq. Src. phrases	Up to 256 million
	Uniq. Tgt. phrases	Up to 256 million
Language Model	Type	3-gram LM
	Size	No limitation[2]

Table 1: Technical Specifications of PanDoRA

[1]The data structure allows 4 billion phrase pairs for each translation direction. This theoretic bound is subject to the capacity limitations of the storage devices.

[2] The size of the language model is subject to the storage capacity limitation of the device running the system.

The remainder of this paper is organized as follows: we first describe the general concepts of phrase-based statistical machine translation systems and then we introduce the PanDoRA system and its major components. We show the performance of PanDoRA system running on a PDA with standard training/testing data sets and discuss the results in the experiments section.

Phrase-based Statistical Machine Translation

In statistical machine translation (SMT), we are given a source language sentence $f_1^J = f_1 \ldots f_j \ldots f_J$, which is to be translated into a target language sentence $e_1^I = e_1 \ldots e_i \ldots e_I$. Among all possible target language sentences, the decoder will choose the one with the highest probability such that the output translation:

$$\begin{aligned} e^* &= \arg\max_e P(e \mid f_1^J) \\ &= \arg\max_e P(f_1^J \mid e) \cdot P(e) \end{aligned} \quad \text{(Eq. 1)}$$

The decomposition of $P(e \mid f)$ in Eq. 1 is based on the source-channel approach which allows us to make use of two types of knowledge sources: translation model (TM) $P(f \mid e)$ and language model (LM) $P(e)$. TM models how likely a source sentence is the translation of the target sentence and LM describes the well-formedness of the generated translation.

The original SMT work described in Brown et al. (1990) models the translation process as a word-to-word mapping. In recent years, various approaches have been developed to use phrase-to-phrase translation models to encapsulate more local context inside the phrases during the translation process (Och et al. 1999; Zhang et al. 2003; Koehn et al. 2003; Vogel 2005). The so-called "phrases" are not linguistically motivated and they could be *n*-grams running across linguistic constituent boundaries such as phrase "*the spokesman said today at*." Phrase-based SMT systems outperform word-based systems and -- despite their lack of linguistic grounding -- have become one of the dominant approaches in machine translation research. Figure 2 shows some examples of Arabic→English phrase translation pairs extracted automatically from the bilingual training data using the PESA method (Vogel 2005).

احتفال المدرسة # school festival is # 0.0034
احتفال المدرسة # school festival is hold # 0.0031
احتفال المدرسة عقد # school festival is hold # 0.9980
احتفال دل # dolls' festival # 0.5431
احتفال دل للفتيات # dolls' festival # 0.3999
إحتفاليا # festive # 0.7535
إحتفاليا تتمناه # no sekku # 0.5081
إحتفاليا تتمناه # sekku # 0.5081
إحتفاليا تتمناه # tango # 0.5081
إحتفاليا تتمناه # tango no # 0.5081
إحتفاليا تتمناه للنمو # no sekku # 0.3066

Figure 2. Phrase Translation Pairs

Another alternative to the classical source-channel approach is the direct modeling of the posterior probability $P(e \mid f)$ using the log-linear model (Och and Ney, 2002):

$$P(e_1^I \mid f_1^J) = \frac{\exp(\sum_{m=1}^{M} \lambda_m \phi_m(e_1^I, f_1^J))}{Z} \quad \text{(Eq. 2)}$$

Each ϕ_m is a feature function that estimates some feature values from (*e, f*). The two knowledge sources used in the classical source-channel approaches can be converted into two feature functions such that:

$$\begin{aligned} \phi_1 &= P(e) \\ \phi_2 &= P(f \mid e) \end{aligned} \quad \text{(Eq. 3)}$$

The denominator *Z* serves as a normalization factor and depends only on the source sentence *f*. For different translation alternatives of *f*, the normalization factors *Z* are all the same. Therefore the decoder only needs to calculate the numerator part in Eq. 2 to search for the optimal translation *e** for *f*.

Under the log-linear model, we could convert the translation model, language model, distortion model, sentence length model and other models into feature functions. This allows us to incorporate more knowledge sources than is the case in the classical source-channel approach. The weights for each feature function are trained using the Minimum Error (MER) optimization on the development set. MER optimizes the feature weights to minimize the errors, or equivalently, maximizing the BLEU/NIST scores (Och, 2003).

PanDoRA System

Based on the general concepts of phrase-based SMT, PanDoRA is engineered from scratch to cope with the limitations on hand-held devices. The code base for PanDoRA is completely different from our phrase-based SMT system for PC platforms.

Compact Data Structure

When running SMT systems on PCs, we usually load all models into memory. The size of phrase-based SMT models can become very large when the training data size increases, or when we consider longer phrases in the translation model. Callison-Burch(2005) estimates that if we consider phrases up to 10 words long, storing all the phrase translation pairs for the NIST-2004 Arabic-English training data[3] would need 30.62 GB in memory. Phrase-based SMT systems are very demanding in resources. Various approaches have been proposed to cut-down memory needs by applying either the delayed phrase construction (Zhang and Vogel, 2005; Callison-Burch, 2005), or phrase table pruning (Eck 2007).

Loading models of several gigabyte into the dynamic memory (SDRAM) is out of question for hand-held devices. Even though the training data for portable speech

[3] The corpus contains 3.75 million sentence pairs and has 127 million words in English, and 106 million words in Arabic.

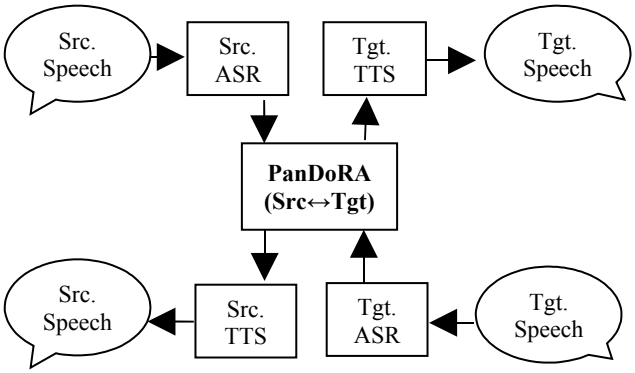

Figure 3. PanDoRA system in a two-way speech translation system for language pair Src. and Tgt.

translation systems is usually limited to domains such as travel and medical, and is usually much smaller than the training data used for the newswire translation, the phrase translation model and language model are still too large to be fit into the dynamic memory of a PDA.

In PanDoRA, we designed a compacted data structure for the translation model and the language model so that:

1. The resulting models are small in size;
2. Decoder can directly access the information without loading the model into the dynamic memory.

Three techniques are developed to achieve these two goals: 1) converting words into integer symbol IDs (vocId); 2) cross-indexing the phrase translation models to reduce the redundancy in model representation; and 3) serializing the model structure to make it directly accessible from disk.

PanDoRA is used mainly for the speech-to-speech translation (SST) systems. It is placed between the Automatic Speech Recognition (ASR) and the Text-to-Speech (TTS) modules in the SST system (Fig. 3). All the input to PanDoRA comes from the ASR output, which means that we operate under a closed vocabulary situation. In other words, there are no unknown word types to the translation system and we can map all the words used in the translation/language models into unique integer vocIds to avoid the hassle of string operations. By doing this, information objects such as *3*-grams and their probabilities, have fixed sizes no matter how many characters each word in the 3-gram has. The fixed object size makes it possible to directly access an object through its index by visiting *start address* + *index* • *sizeof(object)*. During translation, the transcription output from the ASR is first mapped from text to a sequence of vocIds, and vocIds are used throughout the decoding process. After the decoding, vocIds of the translation result are mapped into words of the target language for the Text-To-Speech (TTS) module. Two-byte integers are used for vocIds and the system can handle a vocabulary of 64K words for the source language and 64K words for the target language.

The information in the src→tgt and tgt→src phrase translation tables have a lot of redundancy. Even though the src→tgt table is not symmetric to the tgt→src translation table, most phrases occur in both tables. By cross-indexing, phrases are converted into integer Ids and phrase tables only need to store probability of translating from one phrase ID to another phrase ID.

For hand-held devices the synchronous dynamic random access memory (SDRAM) is usually small (e.g. <64MB). All applications have to share this limited memory. The external storage devices such as CF-cards and SD-cards are much larger in capacity (e.g. 2GB) and can be read/write in reasonably fast rate (about 10MB/second). We serialize the model files in a way such that the decoder can directly access the needed information from the serialized model file on the external storage card. The model does not need to be loaded into the precious SDRAM and the dynamic memory can be saved for the search process during decoding.

For our Thai↔English system, the bilingual training corpus contains 200K sentence pairs, about 1.65 million words on the English side. Using the PESA system, the extracted Thai→English phrase translation table contains 2.6 million entries and the English→Thai direction has 3 million phrase pairs. The two phrase tables are 585 MB when stored on disk in the plain text format. After the cross-indexing and conversion into compact binary format, the two translation models take only 65.27 MB on disk.

Integerized Computation

In standard SMT systems, translation model, language model, and other models use floating points to store probabilities and feature values. On PDAs, there are no built-in floating point co-processors. This means that we have to either use a "soft float" scheme to simulate the floating point calculation which slows down the computation or to integerize all the floating values in the TM/LM.

A pilot study on the PC-based SMT system shows that the overall translation quality does not degrade when all the probabilities are cast from float/double to integers. Table 2 shows both BLEU (Papineni, 2001) and NIST (NIST, 2003) scores of the SMT system using a mid-sized phrase table and tested on the TIDES MT03 Chinese-English evaluation data. Translations from different combinations give almost the same results. This finding corroborate with the study done by Marcellon and Bertoldi (2006), where probabilities are quantized to 2^h number of levels and only h-bits are used to represent a floating point value. Experiments show that quantization with h=8 bits does not affect performance, and gives even slightly better scores. Even when h=4, which corresponds to only 16 bins, the translation quality only loses 1.60% relative in BLEU.

We apply a very simple and straightforward quantization method on the probabilities. We convert all the probabilities into minus log probabilities (cost) and map the costs into integers ranging from 0 to 4095. In other words, the probabilities are quantized to 4096 bins. For those belonging to the same bin, their differences are

ignored. The decoder uses the integer costs to calculate the probability of a hypothesis during decoding. Thus the decoder only needs to use the integer addition and multiplication operations.

TM	LM	BLEU	NIST
Float	Float	19.87	8.03
Int	Float	19.82	7.99
Float	Int	19.94	8.03
Int	Int	19.93	8.04

Table 2. Pilot study on integerize float values in the translation model and language model

Language Model

The *n*-gram language model is converted from its text representation to the binary format. By doing so, each *n*-gram is represented by *n* vocIds and a fixed number of bytes for conditional probabilities and back-off weights. We store all the *n*-grams in a binary file in sorted order. The decoder can directly look up an *n*-gram for its information in the file without loading the LM into the RAM. Similar to the compact translation model, this saves the limited SDRAM for the decoding process, which requires random allocation of memory for dynamic data structures.

Discriminative Language Model

n-gram language models are generative models, i.e., it models the stochastic process of how a sentence is generated. Usually *n*-gram LMs are trained from collections of sentences which are considered to be "correct" and "grammatical". With a trained LM, one can estimate how likely any sentence could be generated given all these "right" examples. The generative n-gram model assumes that any sequence of words could be generated and the total probability sums up to 1, i.e., for any sentence **e** , no matter how bad it is, P(**e**)>0, and for all possible sentences that could be generated.

However, there are certain phenomena in natural languages which we know should never happen. For example, "*the*" should never occur at the end of an English sentence, and particle "*ga*" should never be placed at the beginning of a Japanese sentence. In the case of the generative *n*-gram LM, it has never seen such events in the training data since it is trained only from those "*correct*" examples. *n*-gram LM uses various smoothing techniques to estimate the probability for such unseen events and hopefully it will assign low probabilities to them. But assigning a low probability to "*the </s>*" or "*<s> ga*" can not prevent them from being generated by the SMT decoder. In PanDoRA, language model plays an important role in deciding the correct reordering pattern during decoding, we need to explicitly model the "negative examples" to prevent those ungrammatical n-grams from being generated. In addition to the standard *n*-gram language model, we use a discriminative language model to alleviate the limitations of the generative LM.

Discriminative training has been shown to improve the translation quality (Liang et al., 2006). The idea of using an "anti-language model" has also been tried in speech recognition (Stolcke et al., 2000). We use the perceptron algorithm as described in (Collins, 2002) to train a discriminative language model. Given the current translation model and generative language model, we translate the source side of the bilingual training corpus *f* into *e'*. Unlike Example-based Machine Translation (EBMT) systems, SMT systems usually can not reproduce the same translation as used in the training data, thus the target side of the training corpus *e* is usually different from *e'*. We enumerate all the *n*-grams from the union of *e* and *e'*. For each *n*-gram, we increase the *n*-gram's weight if its frequency in *e'* is less than its frequency in *e* and decrease its weight if it has been over generated. The adjusted weights for *n*-grams are then used as a feature function in the log-linear model (Eq. 2) for the next iteration of decoding.

In other words, we iteratively adjust the weight of each *n*-gram in the discriminative language model to push the generated translation results towards the reference translation.

Decoding

Given a testing sentence, PanDoRA applies the translation model on the sentence and builds a translation lattice. The decoder then searches in this lattice for the optimal path as the output translation for the input sentence.

PanDoRA implements two types of search method in its decoder: a left-to-right monotone decoding and a bottom-up CKY-parsing using the Inverted Transduction Grammar (ITG, Wu 1997).

Monotone Decoding

The monotone decoding in the PanDoRA system is a beam search decoder based on the idea described in Vogel et al. (2003). Once the complete translation lattice has been built, a best-first search through this lattice is performed. In addition to the translation costs, the language model costs are added and the path which minimizes the combined cost is returned. Starting with a special begin-of-sentence hypothesis attached to the first node in the translation lattice, hypotheses are expanded over all outgoing edges from the current node.

The decoder allows for recombination of hypotheses in a flexible way. It is important to keep hypotheses apart if the partial translations end in different words, as this will result in different scores from the language model during the next expansion step. In addition, we can distinguish hypotheses if the length of the translation generated so far is different. This comes into effect when a sentence length model is applied at the sentence end.

The search space becomes very large for long sentences and when there are many alternative translations for each matching source phrase, heavy pruning is enforced during the decoding to make the search space reasonably small to minimize memory usage. Our monotone decoder realizes a standard beam search, where a best hypothesis is stored based on the features used for hypothesis recombination,

and all hypotheses which are worse by some margin are deleted.

With these PDA-specific designs in the decoder, translating one sentence takes less than 10 ms in the monotone decoding mode.

ITG Reordering Decoding

The monotone translation mode works reasonably well for language pairs which have very similar word orders, for example, Spanish and English, but it works poorly for language pairs which have very different word orders. Translating Japanese to English monotonically can result in sentences such as "*An entry visa do I need a?*" and "*In a taxi I left my bag*." To cope with this type of long-distance reordering phenomena, we implement the ITG-style reordering in PanDoRA decoder.

The Stochastic Inversion Transduction Grammar (ITG) introduced by Wu (1997) is a transduction grammar which assumes that a pair of source/target sentences are simultaneously generated in a context-free manner. At each step, a non-terminal X can generate its span in two ways: either straight:

$$X \rightarrow < f_1 f_2, e_1 e_2 > ,$$

or inverted:

$$X \rightarrow < f_1 f_2, e_2 e_1 > ,$$

where e_1 is the translation for f_1 and e_2 for f_2.

Even though it is quite simple and straightforward, ITG has been shown to have high expressiveness. In other words, most of the reordering patterns in natural language translation can be expressed by ITG.

The ITG-style decoding in the PanDoRA system is a CKY parser with beam search. The idea of translating by parsing is similar to the approach used in the Hiero system (Chiang, 2005). Given a source sentence f, the decoder finds the best derivation that generates $<f, e>$ for some e. Unlike the monotone decoder which works on the source sentence from left to right, the CKY parser works bottom-up starting with spans of length 1. While moving up the parsing chart, the decoder adds new partial hypotheses to cell $[j_1, j_2]$ in the chart table if:

1. there is an entry in the translation table where the source phrase is $f_{j_1}^{j_2}$, then add the corresponding translation as a partial hypothesis; or,
2. there exist a partial hypothesis h_1 covering the subspans (j_1, k) and h_2 covering $(k+1, j_2)$, create a new hypothesis h_1h_2 according to the "straight" combination rule and h_2h_1 according to the "inverted" combination rule and add them into the parsing chart.

The number of partial hypotheses grows fast when we move up in the parsing chart. Pruning has to be applied on each chart cell to keep the search space in a reasonable size. Only a few good hypotheses will be kept in the chart for future expansion.

Since we are not decoding from left-to-right, the sentence start symbol are not available to the partial hypotheses until the whole sentence is decoded. Language model probabilities without sentence start are calculated for each partial hypothesis. When a new partial hypothesis is created from combining two shorter ones, the language model probability of the new hypothesis can be estimated from the LM probabilities of the shorter ones with some adjustment based on the words across the boundaries. This makes the language model probability estimation efficient compared to the naive way of calculating the LM probability for all the words in the hypothesis when a new hypothesis is created.

With ITG-style reordering decoding, the qualities are significantly improved for Japanese↔English translation as shown in the next section.

Experiments

We evaluate the performance of PanDoRA on the IWSLT 2005 (Eck and Hori, 2005) Arabic→English and the Japanese↔English test sets. Both BLEU (Papineni, 2001) and NIST (NIST, 2003) metrics are used to evaluate the translation quality.

PanDoRA runs on a HP iPAQ hx2700 series Pocket PC. hx2700 models are powered by the Intel PXA270 processor with a frequency at 624 MHz. The system has 256 MB total memory (192 MB ROM and 64 MB SDRAM) that includes up to 144 MB user available persistent storage memory. We used one 1GB SD card to store the TM/LM models.

Arabic-English Experiments

For Arabic (A) → English (E) system, the training data is from the Basic Travel Expression Corpus (BTEC), which contains 20,000 Arabic/English sentence pairs for the travel domain (Table 3.). The development data (500 Arabic sentences) and testing data (506 sentences) are drawn from the same domain, each with 16 reference translations.

	Arabic	English
Word Tokens	130K	154K
Word Types	18K	6.9K
Sentences	20K	20K
Avg. Sent. Len.	6.5 words	7.7 words

Table 3. Statistics of the BTEC Ar./En training data

We used tools provided by Pharaoh (Koehn, 2004) to extract the phrase translation pairs from the corpus. The Arabic to English phrase table has about 155K translation pairs (Table 4).

Ar/En Pairs	155,825
Uniq. Arabic Phrases	137,836
Uniq. English Phrases	122,460

Table 4. Arabic to English phrase translation model

The English language model is a 3-gram LM trained from the English side of the bilingual corpus using the SRI-LM toolkit (Stolcke, 2002). All the models are converted into compact data structure as described in the previous section. The complete model is of 6.2MB when stored on disk.

Table 5. shows the translation results and speed comparison between Pharaoh and PanDoRA for the Arabic system. The two systems are compared using the same translation model and language model. Feature weights are optimized on the dev-test using MER. Pharaoh runs on a Linux machine with a CPU of 3.2G Hz and PanDoRA runs on the iPaq with a slow CPU at the frequency of 624M Hz. Because of the data structure designed in PanDoRA, the loading time is negligible. Even though the PDA's CPU is not that fast, PanDoRA translates 500 sentences in less than 4.5 seconds, less than 10 ms per sentence.

		Pharaoh		PanDoRA
		Mono.	Reorder	(Mono.)
CPU(Hz)		3.2G		624M
Time	Decoding	0.65s	8.5s	4.3s
	+Model Loading	7.00s	14.0s	4.4s
Dev	BLEU4	47.41	49.05	46.73
	NIST	8.87	8.93	8.30
Test	BLEU4	48.19	47.93	47.06
	NIST	8.81	8.84	8.13

Table 5. Translation results and speed comparisons of Pharaoh and PanDoRA (monotonic decoding) on Arabic/English test set.

The translation quality of PanDoRA is reasonably good compared to the state-of-the-art SMT decoder Pharaoh. The simplification and the heavy punning in the decoder algorithm sacrificed some translation qualities for efficiency.

The reordering model in Pharaoh improves the BLEU score on the dev-test set for about 1.5 points, however the BLEU score for the test set decreased slightly from the monotone decoding. The effect of the reordering model, as is used in the Pharaoh system, is not consistent in this experiment. It is clear that allowing words to be reordered slows down the decoding speed by a factor of 8. This justifies our decision of using monotone decoding for Arabic↔English in PanDoRA because speed is more important than gaining 1 or 2 BLEU points in speech-to-speech translation.

Japanese-English Experiments

The Japanese (J) ↔English (E) system is trained from the JE BTEC corpus for the tourist/medical domain (Table 6.) PESA (Vogel, 2005) is used to align and extract JE and EJ phrase tables from the bilingual corpus. About 4.6 million phrase pairs are extracted for the Japanese-to-English direction and 4.8 million pairs for English-to-Japanese (Table 7). After converting the models to their compact format, the two-way translation model is about 76.11MB on disk.

	Japanese	English
Word Tokens	1.2M	1.0M
Word Types	18K	13K
Sentences	162K	162K
Avg. Sent. Len.	7.32 words	6.18 words

Table 6. Statistics of the BTEC Jp/En training data

J→E Phrase Pairs	4,648,018
E→J Phrase Pairs	4,871,862
Uniq. Japanese Phrases	1,396,719
Uniq. English Phrases	1,015,821

Table 7. Japanese↔English phrase translation model.

We compare the translation performance of PanDoRA with the performance of the Statistical Translation Tool Kit (STTK, Vogel et al. 2003) running on a PC. The reordering model used in STTK is different from PanDoRA's ITG-style reordering. STTK allows for local reordering by leaving a gap and jumping to a distant node in the translation lattice during the decoding time. To restrict reordering, STTK uses position alignment probabilities; specifically, the jump probabilities as estimated in the HMM alignment. Another feature that is different in STTK is its language model. In this experiment, the STTK decoder was used with either the SRI *n*-gram LM, or the Suffix-Array Language Model (SALM, Zhang 2006) which allows arbitrarily long history in estimating the language model probabilities.

Phrase Table	LM	Reorder	STTK	PanDoRA
Pruning	SRI 3-gram	No	46.2	45.93
		Yes	52.4	54.59
	SALM	Yes	53.6	
No Pruning	SRI 3-gram	No	50.3	49.96
		Yes		58.64
	SALM	Yes	59.1	

Table 8. Translation results of the PanDoRA system on the Japanese to English task, compared with the performance of STTK.

Running on the same iPaq hx2700 PDA with the reordering mode is much slower than the monotone decoding mode. On average it takes 0.5 second to translate one sentence with reordering whereas the monotone translation needs only 10ms. However, for language pairs such as Japanese and English, word reordering makes a big difference in translation quality. Table 8 shows the BLEU scores of different conditions. Word reordering accounts for about 10 BLEU points' differences in translation qualities. For all conditions, PanDoRA system has achieved very close performance as STTK.

We also trained a discriminative language models (DLM) to correct errors which can not be captured by the generative language model. As described in the previous section, we use the current PanDoRA model to translate the source side of the training data and use the perceptron algorithm to adjust the weight of an n-gram generated by the SMT such that it will be preferred in the next iteration if it is under-generated compared to the reference translations, or less preferred if it is over generated.

	Training		Test.	
	w/o DLM	with DLM	w/o DLM	with DLM
J→E	58.90	60.01	58.64	58.13
E→J	59.40	60.51	46.40	47.01

Table 9. Performance of the DLM

Table 9 shows the performance of using the discriminative language model for both Japanese to English and English to Japanese translation directions. On the training data set DLM pushes the generated translation towards the reference translation (the target side of the bilingual corpus) and the BLEU scores are improved. However, DLM slightly over-fits the training and does not show the same improvement over the testing data. On the other hand, when we subjectively evaluate the translations generated with/without the DLM, human subjects prefer the translation generated using the DLM. One explanation to this is that BLEU score is not so sensitive to phenomena such as Japanese particles occur at the beginning of the sentence, but correcting such errors make the sentence much more readable to humans.

Relevant Work

Zhou et al. (2004) described a two-way speech translation on an off-the-shelf hand-held device. The translation module in this system uses a statistical natural language understanding (NLU) and a statistical natural language generation (NLG) module. The NLU module is based on a statistical parser which utilizes statistical decision-tree models to determine the meaning and structure of the input utterance. The parser assigns a hierarchical tree structure to the reorganized sentence as predicted by the statistical model. Next, high level semantic translation is performed by the NLU module. The system uses a bilingual dictionary for the domain. To decrease the memory requirement, the size of the dictionary is cut to 9K entries from English to Chinese and 15K entries from Chinese to English.

Zhou et al. (2006) introduces FOLSOM system: a phrase-based statistical machine translation system using weighted finite-state transducers (WFST). FOLSOM is applied in real-time speech translation on scalable computing devices.

Yamabana et al. (2003) used a client-server approach for a mobile speech to speech translation system. The hand-held device is treated as a client, and it sends the compressed speech via Wi-Fi (IEEE 802.11b) to the translation server. The entire speech-to-speech translation process is conducted on the server side, and the translated speech in the target language is later sent back to the client.

Waibel et al. (2003) developed an interlingua-based two way translation system on a consumer PDA that translates between English and Egyptian Arabic. The developed prototype is limited. It was aimed at medical interviews, and dealt with only a few hundred sentence types.

Conclusion

We present PanDoRA, a two-way phrase-based statistical machine translation system for hand-held devices. Various PDA-specific designs have made PanDoRA a practical SMT system that generates translations comparable to the state-of-the-art phrase-based SMT systems in real-time. PanDoRA has been successfully applied in a PDA-based two-way speech-to-speech translation system for several language pairs.

We have been using the PanDoRA system to other PDA-based language applications such as the Automatic Sign Translation (Chang et al., 2007) and hope to see more impact of language technologies on people's daily life.

Bibliographical References

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational. Linguistics*, 16(2):79-85.

Chris Callison-Burch, Colin Bannard, and Josh Schroeder. 2005. Scaling phrase-based statistical machine translation to larger corpora and longer phrases. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 255-262, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Yi Chang, Ying Zhang, Stephan Vogel, and Jie Yang. 2007. Enhancing image-based arabic document translation using a noisy channel correction model. In *Proceedings of MT Summit XI*, Copenhagen, Denmark, September.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL 2005*, pages 263-270, Ann Arbor, MI, June 2005. ACL.

Michael Collins. 2002. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In EMNLP'02: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 1-8, Morristown, NJ, USA. ACL.

Matthias Eck and Chiori Hori. 2005. Overview of the iwslt 2005 evaluation campaign. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT 2005)*, pages 11-32, Pittsburgh, PA, Oct. 24-25.

Marcello Federico and Nicola Bertoldi. 2006. How many bits are needed to store probabilities for phrase-based translation? In *Proceedings on the Workshop on Statistical Machine Translation*, pages 94-101, New York City, June. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proc. of HLT/NAACL 2003*, Edmonton, Canada, May 27-June 1.

Philipp Koehn. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *Proceedings of the 6th Conference of the Association for Machine Translation in the Americas*, Georgetown University, Washington DC, September 28 - October 2.

Percy Liang, Alexandre Bouchard-Cote, Dan Klein, and Ben Taskar. 2006. An end-to-end discriminative approach to machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 761-768, Sydney, Australia, July. Association for Computational Linguistics.

NIST. 2003. Automatic evaluation of machine translation quality using n-gram cooccurrence statistics. *Technical report*, NIST, http://www.nist.gov/speech/tests/mt/doc/ngram-study.pdf.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 295-302, Philadelphia, PA, July.

Franz Josef Och, Christoph Tillmann, and Hermann Ney. 1999. Improved alignment models for statistical machine translation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20-28, University of Maryland, College Park, MD, June.

Franz Josef Och. 2003. Minimum classifiation error training for statistical machine translation. In *ACL 2003: Proc. of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, July.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. *Technical Report RC22176(W0109-022), IBM Research Division*, Thomas J. Watson Research Center.

A. Stolcke, H. Bratt, J. Butzberger, H. Franco, V. R. Rao Gadde, M. Plauche, C. Richey, K. Sonmez E. Shriberg, F. Weng, and J. Zheng. 2000. The SRI march 2000 hub-5 conversational speech transcription system. In *Proceedings of the 2000 Speech Transcription Workshop*, University College Conference Center University of Maryland, May 16-19. NIST.

A. Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proc. Intl. Conf. on Spoken Language Processing*, volume 2, pages 901-904,Denver, CO.

Stephan Vogel, Ying Zhang, Fei Huang, Alicia Tribble, Ashish Venogupal, Bing Zhao, and Alex Waibel. 2003. The CMU statistical translation system. In *Proceedings of MT Summit IX*, NewOrleans, LA, September.

Stephan Vogel. 2005. PESA: phrase pair extraction as sentence splitting. In *Proceedings of the tenth Machine Translation Summit*, pages 251-258, Phuket, Thailand, September.

Alex Waibel, Ahmed Badran, Alan W. Black, Robert Frederking, Donna Gates, Alon Lavie, Lori Levin, Kevin Lenzo, Laura Mayfield Tomokiyo, Jurgen Reichert, Tanja Schultz, Dorcas Wallace, Monika Woszczyna, and Jing Zhang. 2003. Speechalator: two-way speech-to-speech translation on a consumer PDA. In *Proc. of Eurospeech 2003*, pages 369-372, Geneva, Switzerland, September 1-4.

Kiyoshi Yamabana, Ken Hanazawa, Ryosuke Isotani, Seiya Osada, Akitoshi Okumura, and Takao Watanabe. 2003. A speech translation system with mobile wireless clients. In The Companion Voluem to the *Proceedings of 41st Annual Meeting of the Association for Computational Linguistics*, pages 133-136, Sapporo, Hokkaido, Japan, July.

Ying Zhang, Stephan Vogel, and Alex Waibel. 2003. Integrated phrase segmentation and alignment algorithm for statistical machine translation. In *Proceedings of International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE'03)*, Beijing, China, October.

Ying Zhang. 2006. Suffix array and its applications in empirical natural language processing. *Technical Report CMU-LTI-06-010*, Language Technologies Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, Dec.

Bowen Zhou, Daniel Dechelotte, and Yuqing Gao. 2004. Two-way speech-to-speech translation on hand-held devices. In *Proceedings of International Conference of Spoken Language Processing (ICSLP)*, Jeju, Korea, Oct.

Bowen Zhou, Stanley F. Chen, and Yuqing Gao. 2006. Folsom: A fast and memory-efficient phrase-based approach to statistical machine translation. In *Proceedings of the IEEE/ACL 2006 Workshop on Spoken Language Technology*, pages 226-229, Palm Beach, Aruba, December 10-13.