IBM Chinese-to-English PatentMT System for NTCIR-9

Young-Suk Lee, Bing Xiang, Bing Zhao, Martin Franz, Salim Roukos, Yaser Al-Onaizan
IBM T. J. Watson Research Center
1101 Kitchawan Road
Yorktown Heights, NY 10598
U. S. A
{ysuklee, bxiang, zhaob, franzm, roukos, onaizan}@us.ibm.com

ABSTRACT

We describe IBM statistical machine translation systems for the NTCIR-9 Chinese-to-English PatentMT evaluation. IBM's primary system combines the translation output of three distinct statistical machine translation systems – phrase, direct and syntax-based translation systems – using language model re-scoring on confusion networks. Each translation system differs in terms of translation models and decoding techniques, sharing the same pre-processing, word alignments, post-processing and language models. IBM's Chinese-to-English primary system achieved the second highest BLEU score 36.11 out of all primary systems scored.

Categories and Subject Descriptors

[Natural Language Processing]: Machine Translation

General Terms

Algorithms, Experiments, Languages

Keywords

Statistical Machine Translation, Phrase Translation Model, Direct Translation Model, Syntax Model, System Combination, Word Segmentation, Reordering, Parsing, Language Model

Team Name

IBM

Subtasks/Languages

Chinese-to-English PatentMT

External Resources Used

LDC2007T36. LDC2006E93, LDC2009E89

1. INTRODUCTION

IBM primary system for the NTCIR-9 Chinese-to-English patent translation evaluation is the combination of three types of statistical machine translation systems: phrase translation [13], direct translation [5], and syntax-based translation [20]. In this paper, we present the core techniques of each translation system and the NTCIR-9 PatentMT evaluation results.

An overview of the end-to-end translation process of the IBM primary system is shown in Figure 1. For translation model training, we obtain hidden markov model (HMM) and maximum entropy (ME) word alignments from the pre-processed parallel corpus. Translation models are derived from the two types of word alignments. For decoding, the Chinese input texts are pre-processed in the same way as the translation model training, and translated by the respective decoder of the three translation systems. The translation outputs of the three translation systems are combined, case-restored and de-tokenized to produce the final translation output. A block denotes a phrase translation pair consisting of a source and a target phrase.

This paper is organized as follows: Section 2 describes pre-processing. Section 3 discusses word alignments. Section 4 presents the translation models and decoding techniques of the three statistical machine translation systems. Post-processing, which includes system combination, is discussed in Section 5. Section 6 describes the NTCIR-9 Chinese-to-English PatentMT evaluation and other experimental results. Section 7 concludes the paper.

2. PRE-PROCESSING

Pre-processing consists of word segmentation, entity classing, and parsing.

2.1 Word Segmentation

Word segmentation is carried out in 2 stages: In the first stage, we apply a finite state machine (FSM) segmenter, incorporating 7-gram character-based language model [6] for state transition probabilities. For each FSM-segmented Chinese word, its out-of-vocabulary (OOV) status is checked against the translation vocabulary. If the word is an OOV, another segmenter is applied to further segment the word in the second stage.

The probabilistic FSM segmenter tends to under-segment by joining character sequences not seen in the training data, generating many OOV words for the translation lexicon. The other segmenter refers to a bigram look-up table constructed from the training data. If a two character sequence is not in the bigram table, it splits the two characters. This tends to over-segment due to the numerous missing bigrams in the training data.

Both segmenters are trained on about 1.4 million word corpus, shown in Table 1. The 710k word in-house annotation corpus is derived from patent abstracts, claims and titles.

Table 1. Chinese Word Segmentation Corpus Statistics

Source	Training Set	Development Set
LDC2007T36 (CTB6)	~720k words	37489 words
Patent (in-house)	~710k words	13800 words

The FSM segmenter performance on the CTB6 development set is about 96% and the patent development test set is about 93% in F-measure.

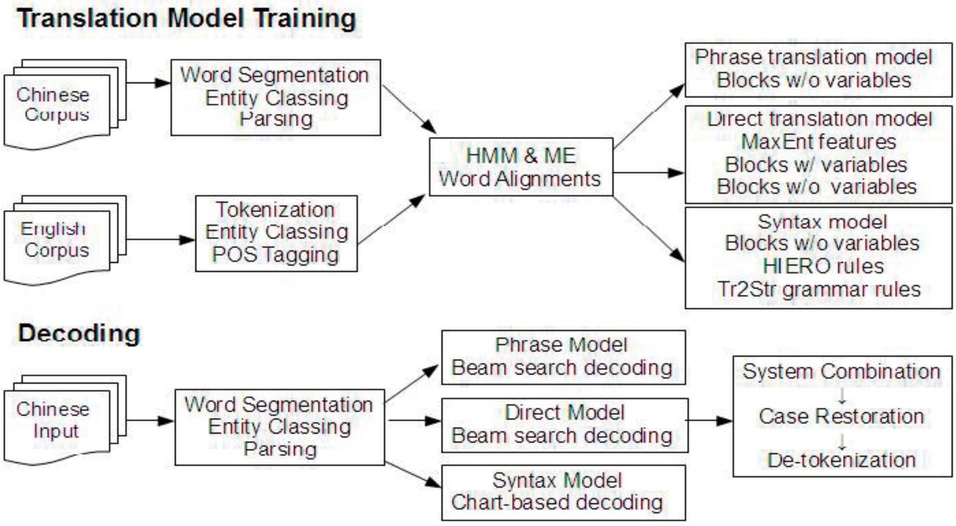


Figure 1. An Overview of the IBM Chinese-to-English Primary System

2.2 Rule-Based Entity Classing

Entity classing is applied to numerical expressions such as DATE, TIME, CARDINALS and ORDINALS and some non-numerical expressions such as URL, EMAIL, based on regular expression pattern matching.

The entity classing has proven useful for patent texts that include measure words such as *mg*, *dl*, *gb*, etc. preceded by Arabic numerals. We treat the combination of Arabic numerals and measure words as a single entity, as shown in Table 2. By treating the Arabic numerals and measure words as a single entity, we reduce the input size in addition to producing accurate translations for the entity classed expression.

Table 2. Entity Classing Examples

根据 该 报告 , 糖尿病 为 表现 出 空腹 血糖 水平 (静脉 血浆 内 的 葡萄糖 浓度) 不 小于 $num_(126mg/dl) , $num_(75g) 口服 葡萄糖 耐量 试验 的 $num_(2) 小时 值 (静脉 血浆 内 的 葡萄糖 浓度) 不 小于 $num_(200mg/dl) 的 病症 。

2.3 Parsing and Tree Transformation

Source language parsing is used for pre-ordering by the phrase translation model, tree-to-string grammar acquisition by the syntax model and feature acquisition by the direct translation model.

We use a maximum entropy Chinese parser [12] trained on CTB6 and an in-house annotated part-of-speech corpus, shown in Table 3.

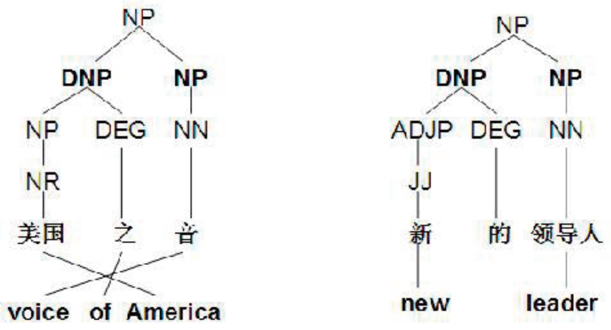
Table 3. Chinese Language Resources for Parsing

Source	Training Set	Annotation Type
LDC2007T36 (CTB6)	~720k words	Tree bank
Patent (in-house)	~240 k words	Part-of-speech

Part-of-speech annotated corpus was parsed using constraint based decoding, and added to the tree bank training data.[1]

The primary purpose of using the parser was to better model the reordering phenomena between Chinese and English with context free grammar (CFG) rules. However, as shown in Figure 2, there are quite a few Chinese constructs whose reordering pattern cannot be easily captured with CTB6-style CFG representations.[2]

Figure 2 illustrates a CFG rule 'NP → DNP NP'. The order of the two children DNP and NP may or may not be swapped to produce the same word order as English translations.



(a) (b)

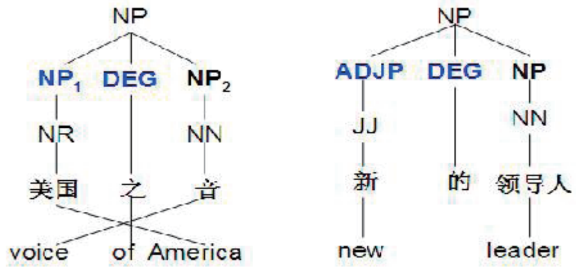
Figure 2. Chinese DNP phrases with reordering ambiguities

Whether or not to swap the order between DNP and NP depends on the part-of-speech of the DNP 's children. If the part-of-speech of a child is an NN or NR, Figure 2-a, NP DNP order is more likely, whereas if the part-of-speech of a child is a JJ, Figure 2-b, the DNP NP order is very likely to remain unchanged.

In order to capture the reordering contexts, we remove the parent node (DNP) that dominates the disambiguation contexts, Figure 2. After DNP node removal, disambiguation contexts are all captured as the children of the same NP parent, Figure 3.

[1] We later learned that 1,688 sentences (22%) and 5,172 tokens (2.7%) of the part-of-speech annotated data had the issue of merging more than one part-of-speech tagged word into a single token, as in 复合_JJ 材料_NN, 多_CD 层_M ,同_DT 一_CD.

[2] Refer to [8] for the limitations of CFG representations in English-to-Japanese translations.



(a) (b)

Figure 3. DNP-node removal to capture reordering contexts

We learn the constituents undergoing transformation with probabilistic tree-to-string grammars [20], informally described below:

- Word align a parallel corpus and parse the source language corpus
- Train a probabilistic tree-to-string grammar
- Identify the constituents whose children sequence is highly ambiguous between monotone and non-monotone word order
- For the constituent identified in the previous step, remove the node that dominates the disambiguation contexts for reordering

For the DNP NP sequence in Figure 2, tree-to-string grammar rules assign probability 0.51 to the non-monotone order, Figure 2-a, and 0.49 to the monotone word order, Figure 2-b, indicating that it is highly ambiguous between monotone and non-monotone word order. However, after DNP node removal, tree-to-string grammar rules assign the re-ordering probability 0.77 to the NP_1 DEG NP_2 sequence in Figure 3-a, and the monotone order probability 0.74 to the ADJ DEG NP sequence in Figure 3-b.

We train the parser after applying the tree transformation (i.e. constituent node removal) to the original tree bank. We use the parser trained on the transformed tree bank.

3. WORD ALIGNMENT

All systems combine blocks derived from an unsupervised HMM [15] and a supervised ME [4] word alignments to train translation models. Throughout this paper, we use the term *block* (*b*) to denote a phrase translation pair consisting of a source ($\bar{f}$) and a target phrase ($\bar{e}$).

3.1 Viterbi HMM Alignments

HMM alignment utilizes two conditional probability models: State transition and word-to-word translation probability models.

Assuming the source word sequence (f_1, f_2, … f_J) to be the observation and the target words (e_1, e_2, …, e_I) to be states, state transition probabilities are obtained according to (1):

$$(1)\quad p(i|i', f_{i'} I) = \frac{cnt\,|\,i - i'\,|}{\sum_{l=1}^{I} cnt\,|\,l - i'\,|}$$

I denotes the target sentence length. $f_{i'}$ is the source word generated from the previous state i'. i is the the state generating the current source word f_i. State transition probability is the count of the jump width between i and i' divided by the total count of the jump width between i' and any current state from 1 to I.

Word-to-word translation probability is obtained according to (2): f' is the source word given target word e.

$$(2)\quad p(f \mid e) = \frac{cnt\,(f, e)}{\sum_{f'} cnt\,(f', e)}$$

The optimal path is found by dynamic programming with a recursive formula (3), where $Q(i, j)$ is a partial probability function that generates the observation sequence ranging from the first source word to the current source word f_j in state i that corresponds to the target word e_i.

$$(3)\quad Q(i,j) = p(f_j \mid e_i) \max_{i'=1,\ldots,I} [p(i \mid i', f_{j-1,} I) \cdot Q(i', j-1)]$$

For training, we initialize the word to word translation model with an IBM model 1. State transition probabilities are initialized with a prior model that assigns higher probabilities to smaller jumps where any jump width from 0 to 19 are assigned positive probabilities and any jump width greater than 19 are assigned zero probability. Once the optimal path for a sentence pair is found, we update the alignment and obtain the translation and jump counts from the updated alignment in the E-step and re-normalize the translation and state transition probabilities in the M-step.

3.2 Maximum Entropy Alignments

We use the maximum entropy word aligner described in [4]. We use the resources in Table 4 to train the aligner. 48 patent sentence pairs are selected from the 1 million parallel training corpus [22] for in-house annotations.

Table 4. Linguistic Resources for ME Aligner Training

Source	Quantity
LDC2006E93	14938 sentence pairs
LDC2009E89	14931 sentence pairs
Patent descriptions	48 sentence pairs
Total	29917 sentence pairs

Similar to HMM aligner, the ME aligner consists of two models: observation and transition models. Word alignment is carried out on the basis of the scoring function (4) for each word pair:

(4) AlignmentScore = [log(transitionScore) x λ] + [(1–λ) x log((meScore x α + (1–α) x m1Score))]

In (4), λ is the transition model weight and 1–λ is the observation model weight. α is the ME score weight and (1–α) is the IBM Model 1 score weight.

For observation model training, we use the same features as in [4]: source and target word lexical features, WordNet and spelling features In addition, we use source character and target word features, drawing on the fact that each Chinese character has its own meaning and often corresponds to an English word.

From the human annotated data, we obtain about 15.5k source and 14.8k target vocabulary, which is highly limited compared with about 171k source and 204k target vocabulary acquired from the parallel training data. Therefore, we smooth the observation model score with an IBM Model 1 estimate, as shown in the (4) as (1–α) x m1Score.

4. STATISTICAL MACHINE TRANSLATION SYSTEMS

We present the translation model acquisition and decoding techniques of phrase, direct and syntax-based translation systems.

All translation models are trained on the 1 million parallel training corpus. A large 7-gram language model (LM) is trained on about 14 billion word English patent data published between 1993 and 2005. A smaller 5-gram language model is trained on the English side of the 1 million sentence pair parallel training corpus containing 45 million words, [22].

4.1 Phrase Translation System

IBM phrase translation system is an extension of the phrase translation system detailed in [13].

4.1.1 Three Types of Translation Model

Phrase translation model probabilities are learned from the combination of blocks derived from an HMM word alignment and blocks extracted from an ME word alignment.

For block acquisition from an HMM alignment, we word-align a parallel corpus bi-directionally: one from a source word position to a target word position, (A_1: $f \rightarrow e$) and the other from a target word position to a source word position (A_2: $e \rightarrow f$). Given two types of alignment $A_P = A_1 \cap A_2$ and $A_R = A_1 \cup A_2$, blocks are derived according to the projection and extension algorithms [7, 13], and we filter out blocks containing non-consecutive source word sequences.

For block acquisition from an ME alignment, we use the high recall alignment A_R, extracting blocks consisting of consecutive source and target word sequences with the constraint that source and target sequences be aligned only to the words within the block.

Three types of phrase translation model probabilities are derived from the combined blocks, as in (5)¬(7), where $\bar{e}$ is the source phrase, $\bar{f}$ the target phrase and b=($\bar{e}$, $\bar{f}$).

(5) Direct Model

$$p(\bar{e} \mid \bar{f}) = \frac{\boldsymbol{count}(\bar{e}, \bar{f})}{\sum_{\bar{e}'} \boldsymbol{count}(\bar{e}', \bar{f})}$$

(6) Source-channel Model

$$p(\bar{f} \mid \bar{e}) = \frac{\boldsymbol{count}(\bar{f}, \bar{e})}{\sum_{\bar{f}'} \boldsymbol{count}(\bar{f}', \bar{e})}$$

(7) Unigram Model

$$p(b) = \frac{count(b)}{\sum_{b'} count(b')}$$

4.1.2 Decoding

The phrase decoder uses a beam search strategy and a scoring function that incorporates 6 types of manually weighted features listed below:

- phrase translation models
- modified IBM model scores derived from blocks consisting of one source and one target word
- word count penalty
- block count penalty
- lexicalized inbound and outbound distortion models [1]
- a mixture language model with a large 7-gram and a small 5-gram LM's

We expand the search space, by increasing the beam size, i.e. cardinality pruning threshold, up to 5000,[3] and the coverage pruning threshold up to 75.[4] To capture the relatively high degree of distortion between Chinese and English, we set the skip size 5[5] [14].

4.1.3 Parsing-based Pre-ordering

The baseline phrase translation system incorporating lexicalized distortion models is inadequate for capturing the non-local distortion between Chinese and English [3, 16]. To improve word order accuracy, we apply parsing-based pre-ordering for translation model training and decoding [8].

We use the Chinese maximum entropy parser trained on the tree bank with the tree transformation described in Section 2.3. 1-best reordering rules are learned from the tree-to-string grammar rules. Top 5 most frequently occurring reordering rules are given in Table 5, where the reordering pattern 2 1 0 indicates that the source phrase order is reversed.

Table 5. Most frequent Reordering Rules

Frequency	Source Constituent	Reordering Pattern
25797	NP → NP_0 DEG_1 NP_2	2 1 0
15749	NP-OBJ → NP_0 DEG_1 NP_2	2 1 0
11849	NP-SBJ → NP_0 DEG_1 NP_2	2 1 0
7281	PP-LOC → P_0 NP_1 LC_2	2 0 1
3704	NP→NP_0 DEG_1 $ADJP_2$ NP_3	2 3 1 0

We used 144 such rules, which occur more than 10 times in the 1 million sentence pair training corpus.

4.2 Direct Translation Model

The Direct Translation Model (DTM) is a special maximum-entropy (ME) model [5]. The model has the following form:

$$(8)\ \ p(\bar{e}, j \mid \bar{f}) = \frac{p_0(\bar{e}, j \mid \bar{f})}{Z} \exp \sum_i \lambda_i \phi_i(\bar{e}, j, \bar{f})$$

$\bar{f}$ is a source phrase, and $\bar{e}$ is a target phrase. j is the jump distance from the previously translated source word to the current source word. During training j can vary widely due to automatic word alignment in the parallel corpus. To limit the sparseness created by long jumps, j is capped to a window of source words (-5 to 5 words) around the last translated source word. Jumps outside the window are treated as the maximum jump allowed. $p_0(\bar{e}, j \mid \bar{f})$ is a prior distribution, Z is a normalizing term, and $\phi_i(\bar{e}, j, \bar{f})$ are the features of the model, each being a binary question asking about the source and target streams. The feature weights λ_i are estimated with the Improved Iterative Scaling (IIS) algorithm.

4.2.1 Block Extraction

Blocks are extracted from the HMM word alignment ($f \rightarrow e$ direction only) and the ME word alignment. The "Projection Constraint", which requires that the source and target sequences be aligned only to the words within the block, is then checked to

[3] Typically set to 250.

[4] Typically set to 25.

[5] Typically set to 1 or 2 for optimal performances.

ensure that the phrase pair is consistent. A slight relaxation is made to the traditional phrase blocks in that a variable is allowed at the source or target side to accommodate the fork-style alignments [5].

4.2.2 Features

Large number of features utilized in the ME model [18] fall into the following categories:

- Lexical features that examine source word, target word and jump;
- Lexical context features that examine the previous and next source words, and also the previous two target words;
- Segmentation features based on morphological analysis. In this work, they are features based on Chinese characters;
- Part-of-speech (POS) features that collect the syntactic information from the source and target words;
- Source parse tree features that examine the source parse labels, sibling labels and coverage;
- Coverage features that examine the coverage status of the source words to the left and to the right. They fire only if the left source is open (un-translated) or the right source is closed.

The total number of features used in the system is around 15M.

4.2.3 Decoding

A beam search decoder similar to the phrase-based systems is used to translate the Chinese sentences into English. These decoders have two parameters that control their search strategy: (a) the skip length (how many positions are allowed to be un-translated) and (b) the window width, which controls how many words are allowed to be considered for translation.

The primary difference between a DTM decoder and standard phrase-based decoders is that the maximum entropy model provides a cost estimate of producing this translation using the features described above. Another difference is that the DTM decoder handles blocks with variables. When such a block is proposed, the initial target sequence is first output, and the source word position is marked as being partially visited. Then an index into which segment was generated is kept for completing the visit at a later time. Subsequent extensions of this path can either complete this visit or visit other source words. On a search path, we make a further assumption that only one source position can be in a partially visited state at any point. This greatly reduces the search task and suffices to handle the type of blocks encountered in Chinese to English translation.

Decoder parameters are optimized with the simplex algorithm in [23].

4.3 Syntax Model

Our syntax-based translation system is a chart-based decoder as described in [20].

For this patent evaluation, we applied a variation of the source tree to target string grammar (tree-to-string). We applied a few tree transformation operations to handle fragment of tree (or tree-sequence) [19], binarizations, lexicalizations, and flattening multi-level trees to extract PSCFG style reordering rules. Practically, we extract at least one tree-to-string rule for each aligned phrase-pair (block). By applying such approach, we significantly improved the grammar coverage, and also enable our decoding process to be less sensitive to the parse tree errors.

4.3.1 Grammar Acquisitions

Grammar set extraction is divided into two stages. Given the source parsed, word-aligned parallel data, we first extract the same set of blocks from the training data using the standard approach, as shared by phrase-decoder system. In this stage, we allow blocks with source side of length up to 12 tokens, and max target side of length 15 tokens.

After blocks are extracted, we walk through every aligned pair of phrases. For each source phrase, we get the immediate common parent for the source span, and then we retrieve the tree fragment which cover the span. From this tree-fragment, we identify the frontier nodes, which can be generalized into non-terminals, and necessary lexicalizations which can model the fork-style alignment.

Then we consider tree-transformations, to restructure the trees via synchronous binarizations, flatten the trees by removing the interior nodes, and add additional markups to indicate the transformed trees. One important additional markups are sentence-begin and sentence-end, as at which positions, we generally do not expect the dramatic re-orderings to happen. More details can be found in [21].

After all grammar rules are extracted, we prune the tree-to-string rules with a hard threshold of frequency. Any rules that occurred less than 3 times are dropped.

Note that the tree transformation applied to the tree bank for the parser training, cf. Section 2.3, is analogous to the flattening operator in [21]. However, the tree transformation applied to the tree bank is primarily to capture the disambiguation contexts for re-ordering within the CFG framework, i.e., all CFG rules are encoded in a tree of depth 2, whereas the tree transformations by the syntax model are largely motivated to increase the grammar coverage. Constituent node removal applied to the tree bank is targeted on the nodes that dominate disambiguation contexts, and the transformation enables the CFG to capture the disambiguation contexts by flattening a tree of depth 3 or more to a tree of depth 2. On the other hand, the tree transformations by the syntax model is largely determined by the alignment decisions. Empirically, the tree transformations applied to the tree bank and those performed by the syntax model may or may not overlap.

4.3.2 Decoding

During decoding, we first populate a chart (a vector of cells) with the relevant reordering rules.

We walk through the given parse tree for the test sentence, and at every cell, we retrieve the immediate common parent for that source span, and correspondingly the tree fragment for that span. Then we apply every predefined operator to transform the tree, and see if the transformed tree matches some grammar rules in our library. If there is no tree-to-string rule matching the span, we back off to Hiero-style [3] unlabeled rules, and if there is no Hiero rule, we further back off to monotone ITG-style [17] glue rule for each span. We collect all the matched rules for each cell, and expand them one by one from bottom up to fill up the chart until we reach the final translation.

Different from the DTM decoder, which can use millions of features during decoding, we used 19 feature functions, including relative frequencies in both directions, IBM model-1 score in both directions, rule count for blocks, glue rule, Hiero-style rule, and tree-to-string rule, brevity penalty, interpolated IBM model-1 score, ngram LM scores, and several feature functions to handle function/content word mismatches in both directions. The feature weights are optimized with the simplex algorithm in [23].

Pruning needs to be done to improve the speed and memory requirement. We kept at most 200 partial hypotheses per rule, and at most 400 final expanded hypotheses per cell. During the populating step, we kept at most 20 rules per cell, ranked by their frequencies. Glue rule is not encouraged for short source spans with less than 10 tokens, if we see tree-to-string or Hiero-style rule for that span. We merge the hypotheses which share the same state but different cost. We also trigger early stopping when the search space is over the maximum amount of memory predefined, and the decoding process will reach final translation quicker using a restricted the search space at that time.

5. POST-PROCESSING

We combine the translation outputs from the three statistical machine translation systems, restore case and perform de-tokenization[6] to produce the final translation output.

5.1 System Combination

The system combination component is based on an incremental alignment approach, using inversion transduction grammars (ITGs) for aligning system outputs [9, 10].

More specifically, 1-best output of the Syntax Model is used as the “skeleton”, to which the 1-best outputs of the other translation models are aligned, one by one, using ITGs. The resulting confusion network is re-scored using a combination of two language models (LMs): 7-gram LM trained on 14 billion word corpus, and 5-gram LM trained on the English side of the parallel corpus. Values of the free parameters (relative weight of the LM, skeleton system choice, and the word insertion/deletion penalty) are selected using development and tuning sets, containing 500 and 1500 and sentences.

As shown in Table 6, the system combination yields increase of 0.76 cased and 0.91 lower-cased BLEU over the best performing translation system.

5.2 Case Restoration

We treat case restoration as a translation task that translates lower-cased English words to true-cased counterparts.

We use the translation model to identify all possible case variations of a lower-cased word, and use a 5-gram language model score to select the best alternative. The 5-gram language model is trained on the English side of the 1 million sentence pair training corpus.[7]

Table 6. Cased vs. Lower-cased BLEU scores

Systems	Cased BLEU	Lower-cased BLEU
System Combination	36.12	37.58
Phrase Model	32.56	33.87
Direct Model	33.60	34.87
Syntax Model	35.36	36.67

The gap between lower-cased and cased BLEU scores are shown in Table 6 for the systems submitted to NTCIR-9 PatentMT evaluations. Cased BLEU scores are lower than lower-cased BLEU scores by 1.27 to 1.46 BLEU points.

5.3 De-tokenization

For de-tokenization, we use look-up tables that lists tokens to be merged to the preceding and following tokens, shown in Table 7.

Table 7 De-tokenization Look-up Table

No space before tokens	- ; , : ! ?) . % ~] / ' ↓↑ 's 't 'm 'll
No space after tokens	- ~ ($ [/ ↓↑

Tokens that require no space before tokens are merged to the preceding tokens, and includes contractual forms such as *'s*, *'t*, *'m*, *'ll* as well as punctuations and symbols. Tokens that require no space after tokens are merged to the following tokens, and mostly includes symbols.

6. EXPERIMENTS & EVALUATIONS

We present the impact of various techniques on phrase translation system, NTCIR-9 Chinese-to-English translation evaluation results and discuss the impact of contextual data on our submission systems.

6.1 Impact of Techniques

In Table 8, we show the impact of various modules and techniques on the phrase translation system, using the NTCIR-9 development test data set. System performances are shown in lower-cased BLEU with up to 4-grams for modified precision computation [11].

Table 8. Impact of various techniques on phrase translation system performances

Systems	BLEUr1n4
Baseline with HMM blocks	28.96
Pre-processing + ME blocks	32.20
7-gram LM w/ 14 billion words	33.84
Expanded search space	34.97
Parsing-based pre-ordering	36.00

The key properties of the baseline system that differs from the final submission system are listed below:

- Use only the smaller 5-gram language model trained on 45 million words.
- For pre-processing, use only the FSM word segmenter and an entity spans over exactly one token.
- Use blocks derived from HMM word alignment only.
- Use a restricted search space with the beam size 250, skip size 1 and the maximum translations per source phrase 20.

Enhanced pre-processing and addition of blocks derived from ME word alignment improves the performance by 3.24 BLEU points. Pre-processing improvement includes 2-stage word segmentation, improved non-Chinese word tokenization, entity classing bug fix,[8]

[6] Merge punctuations, symbols and/or contracted forms to the preceding or the following tokens

[7] The scores are computed with the original in-house BLEU script, different from mteval-v13a.pl, using the reference translation provided by the PatentMT evaluation organizers.

[8] Previously if the entity content is of the format 36), it used to strip off the right parenthesis, e.g., as 36. This bug was fixed by

and the entity classing extension that can span over more than one word token.

Using a mixture language model that includes the 7-gram language model trained on the 14 billion word corpus − in addition to the smaller 5-gram LM − leads to statistically significant 1.64 BLEU gain.

Expansion of the search space – beam size increase from 250 to 5000, coverage vector pruning threshold increase from 25 to 75, skip size increase from 1 to 5 and the maximum translations per source phrase from 20 to 50 – improves the performance by 1.13 BLEU points.

Parsing-based pre-ordering for both translation model training and decoding improves the performance by 1.03 BLEU point.

The overall performance improvement of the final system over the baseline is 7.04 BLEU points.

6.2 NTCIR-9 Evaluation Results

IBM primary system performance in the NTCIR-9 Chinese-to-English PatentMT evaluation is shown in Table 9.

Table 9. IBM Primary System Performance in NTCIR-9

Metrics	Segments Scored	Scores
BLEUr1n4c	2000	36.11
NIST	2000	8.51
RIBES	2000	0.80
Adequacy Score	300	3.39

BLEU score 36.11 and RIBES score 0.8 are the second highest, and the NIST score 8.51 is the third highest among all the primary systems scored.

The Pearson correlation coefficients between the BLEU scores and the human evaluation adequacy scores of the 23 systems scored is 0.915 with R2 = 0.837, indicating that the correlation between BLEU and human evaluations is fairly high. MT scores of the 23 systems are shown in Table 10.

Table 10. MT Evaluation Scores of 23 Systems Scored

Systems	BLEU	HE	Systems	BLEU	HE
G1	39.44	4.03	G3	26.49	3.30
G6	36.11	3.39	G15	26.38	3.13
G17	35.69	3.42	G4	25.97	3.05
G12	34.76	3.40	G13	25.84	2.85
G10	32.76	3.34	G9	25.36	3.04
G14	32.29	3.51	G11	17.80	2.41
G7	31.97	3.30	BS1	30.72	3.29
G5	31.46	3.34	BS2	29.32	2.89
G18	30.74	3.29	ONLINE	25.69	2.97
G16	30.26	3.23	RBMT1	10.75	2.27
G8	29.27	3.19	RBMT2	12.80	2.66
G2	27.79	3.11			

In Table 10, BS1 denotes BASELINE1, BS2 BASELINE2 and ONLINE denotes ONLINE1. HE stands for human evaluation adequacy scores.

6.3 Impact of Contextual Data

During the evaluation period, we derived 23,005 sentence pairs of additional parallel corpus from the 103 document pairs of the development test set contextual data (about 670k word tokens in Chinese and 709k word tokens in English). We automatically sentence aligned them with an in-house sentence aligner.

We added this parallel corpus for translation model training of the submission systems. Table 11 illustrates the impact of the contextual data on system performances.

Table 11. Impact of contextual data on system performance

Systems	w/ context data	w/o context data
System Combination	36.11	35.00
Phrase Model	32.56	32.42
Direct Model	33.60	33.48
Syntax Model	35.55	34.42

While the contextual data was helpful for all of the systems, it was particularly effective for the syntax model, improving the BLEU score by 1.13 points. We see the similar gap of 1.11 BLEU point for system combination, namely 36.11 with contextual data and 35.0 without contextual data.

7. Conclusions

In this paper, we described IBM primary system for the NTCIR-9 Chinese-to-English patent translation evaluation. IBM primary system is a combination of three types of statistical machine translation systems: phrase translation, direct translation, and syntax-based translation.

All translation models are trained on the 1 million parallel training corpus. All systems use a mixture language model consisting of a large 7-gram language model is trained on about 14 billion word English patent data published between 1993 and 2005 and a smaller 5-gram language model is trained on the English side of the 1 million sentence pair parallel training corpus containing 45 million words [22]. All systems share the same word segmentation, entity classing, Chinese parsing, case restoration and de-tokenization.

IBM Chinese-to-English patent translation system demonstrated a highly competitive performance in the NTCIR-9 PatentMT evaluation by achieving the second highest BLEU score 36.11 among all primary systems scored.

Acknowledgement

We would like to thank IBM Statistical Machine Translation and Realtime Translation teams for the various software tools used in the evaluations.

References

[1] Y. Al-Onaizan and K. Papineni. 2006. Distortion models for statistical machine translation. *Proceedings of ACL-COLING. P*ages 529-536.

[2] P. Brown, V. Della Pietra, S. Della Pietra, and R. Mercer. 1993. The mathematics of statistical machine translation:

correctly generating the entity content as 36).

parameter estimation. *Computational Linguistics*, 19(2):263−311.

[3] D. Chiang. 2005. A Hierarchical Phrase-Based Model for Statistical Machine Translation. *Proceedings of ACL*. Pages 263-270.

[4] A. Ittycheriah and Salim Roukos. 2005. A Maximum Entropy Word Aligner for Arabic-English Machine Translation. *Proceedings of Human Language Technology.* Pages 89-96, Vancouver, October 2005.

[5] A. Ittycheriah and S. Roukos. 2007. Direct translation model 2. *Proceedings of NAACL-HLT.* Pages 57-64, Rochester, NY, April.

[6] Y-S. Lee, K. Papineni, S. Roukos, O. Emam and H. Hassan. 2003. Language Model Based Arabic Word Segmentation. *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics.* Pages 399-406. Sapporo, July 2003.

[7] Y-S. Lee. 2006. IBM Arabic-to-English translation for IWSLT 2006. *Proceedings of the International Workshop on Spoken Language Translation (IWSLT 2006).* Pages 45-52. Kyoto. November 2006.

[8] Y-S. Lee, B. Zhao and X. Luo. 2010. Constituent Reordering and Syntax Models for English-to-Japanese Statistical Machine Translation. *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010).* Pages 626-634. Beijing, August 2010.

[9] J. Olive, C. Christianson and J. McCary (Editors.). 2011. *Handbook of Natural Language Processing and Machine Translation, DARPA Global Autonomous Language Exploitation*, ISBN 978-1-4419-7712-0, 352-360

[10] D. Karakos, J. Eisner, S. Khudanpur, and M. Dreyer. 2008. Machine translation system combination using ITG-based alignments. *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics and Human Language Technologies: Short Papers (HLT-Short '08).* Pages 81-84.

[11] K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. *Proceedings of ACL*. Pages 311–318.

[12] A. Ratnaparkhi. 1999. Learning to Parse Natural Language with Maximum Entropy Models. *Machine Learning: Vol. 34*. Pages 151-178.

[13] C. Tillmann. 2003. A Projection Extension Algorithms for Statistical Machine Translation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2003)*, Sapporo, July 2003.

[14] C. Tillmann and H. Ney. 2003. Word Reordering and a Dynamic Programming Beam-Search Algorithm for Statistical Machine Translation. *Computational Linguistics: Vol 29, No. 1.* Pages 97-133. March 2003.

[15] S. Vogel, H. Ney and C. Tillmann. 1996. HMM-Based Word Alignment in Statistical Machine Translation. *Proceedings of the 16th International Conference on Computational Linguistics (Coling 1996).* Pages 836-841, Copenhagen, August 1996.

[16] C. Wang, M. Collins, and P. Koehn. 2007. Chinese Syntactic Reordering for Statistical Machine Translation. *Proceedings of EMNLP-CoNLL*. Pages 737-745.

[17] D. Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics 23(3)*:377-404.

[18] B. Xiang and A. Ittycheriah. 2011. Discriminative feature-tied mixture modeling for statistical machine translation. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: short papers*. Pages 424-428, Portland, June 2011.

[19] H. Zhang, M. Zhang, H/ Li, A. Aw, and C. L. Tan. 2009. Forest-based tree sequence to string translation model. *Proceedings. of ACL*. Pages 172–180.

[20] B. Zhao, Y. Al-Onaizan. 2008. Generalizing Local and Non-Local Word-Reordering Patterns for Syntax-Based Machine Translation. *Proceedings of EMNLP*'. Pages 572~581

[21] B. Zhao, Y-S. Lee, X. Luo, L. Li. 2011. Learning to Transform and Select Elementary Trees for Improved Syntax-based Machine Translations. *Proceedings of ACL*. Pages 846–855.

[22] I. Goto, B. Lu, K. P. Chow, E. Sumita and B. K. Tsou. 2011. Overview of the Patent Machine Translation Task at the NTCIR-9 Workshop. NTCIR-9, Tokyo, December 2011.

[23] B. Zhao and S. Y. Chen. 2009. A Simplex Armijo Downhill Algorithm for Optimizing Statistical Machine Translation Decoding Parameters (short paper). *Proceedings of NAACL-HLT.*