A Polynomial-Time Algorithm for Statistical Machine Translation

Dekai Wu
HKUST
Department of Computer Science
University of Science and Technology
Clear Water Bay, Hong Kong
dekai@cs.ust.hk

Abstract

We introduce a polynomial-time algorithm for statistical machine translation. This algorithm can be used in place of the expensive, slow best-first search strategies in current statistical translation architectures. The approach employs the stochastic bracketing transduction grammar (SBTG) model we recently introduced to replace earlier word alignment channel models, while retaining a bigram language model. The new algorithm in our experience yields major speed improvement with no significant loss of accuracy.

1 Motivation

The statistical translation model introduced by IBM (Brown et al., 1990) views translation as a noisy channel process. Assume, as we do throughout this paper, that the input language is Chinese and the task is to translate into English. The underlying generative model, shown in Figure 1, contains a stochastic English sentence generator whose output is "corrupted" by the translation channel to produce Chinese sentences. In the IBM system, the language model employs simple n-grams, while the translation model employs several sets of parameters as discussed below. Estimation of the parameters has been described elsewhere (Brown et al., 1993).

Translation is performed in the reverse direction from generation, as usual for recognition under generative models. For each Chinese sentence $\mathbf{c}$ that is to be translated, the system must attempt to find the English sentence $\mathbf{e}*$ such that:

$$(1) \qquad \mathbf{e}* = \operatorname*{argmax}_{\mathbf{e}} \Pr(\mathbf{e}|\mathbf{c})$$

$$(2) \qquad \phantom{\mathbf{e}*} = \operatorname*{argmax}_{\mathbf{e}} \Pr(\mathbf{c}|\mathbf{e}) \Pr(\mathbf{e})$$

In the IBM model, the search for the optimal $\mathbf{e}*$ is performed using a best-first heuristic "stack search" similar to A* methods.

One of the primary obstacles to making the statistical translation approach practical is slow speed of translation, as performed in A* fashion. This price is paid for the robustness that is obtained by using very flexible language and translation models. The language model allows sentences of arbitrary order and the translation model allows arbitrary word-order permutation. The models employ no structural constraints, relying instead on probability parameters to assign low probabilities to implausible sentences. This exhaustive space, together with massive number of parameters, permits greater modeling accuracy.

But while accuracy is enhanced, translation efficiency suffers due to the lack of structure in the hypothesis space. The translation channel is characterized by two sets of parameters: translation and alignment probabilities.[1] The translation probabilities describe lexical substitution, while alignment probabilities describe word-order permutation. The key problem is that the formulation of alignment probabilities $a(i|j, V, T)$ permits the Chinese word in position j of a length-T sentence to map to any position i of a length-V English sentence. So V^T alignments are possible, yielding an exponential space with correspondingly slow search times.

Note there are no explicit linguistic grammars in the IBM channel model. Useful methods do exist for incorporating constraints fed in from other preprocessing modules, and some of these modules do employ linguistic grammars. For instance, we previously reported a method for improving search times in channel translation models that exploits bracketing information (Wu and Ng, 1995). If any brackets for the Chinese sentence can be supplied as additional input information, produced for example by a preprocessing stage, a modified version of the A*-based algorithm can follow the brackets to guide the search heuristically. This strategy appears to produces moderate improvements in search speed and slightly better translations.

Such linguistic-preprocessing techniques could

[1] Various models have been constructed by the IBM team (Brown et al., 1993). This description corresponds to one of the simplest ones, "Model 2"; search costs for the more complex models are correspondingly higher.

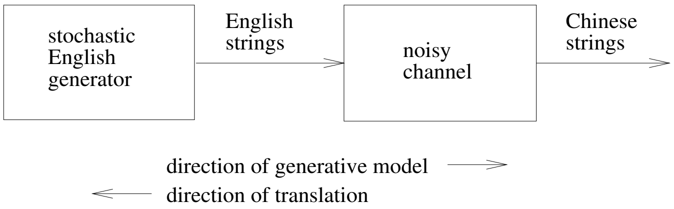

Figure 1: Channel translation model.

also be used with the new model described below, but the issue is independent of our focus here. In this paper we address the underlying assumptions of core channel model itself which does not directly use linguistic structure.

A slightly different model is employed for a word alignment application by Dagan *et al.* (Dagan, Church, and Gale, 1993). Instead of alignment probabilities, offset probabilities $o(k)$ are employed, where k is essentially the positional distance between the English words aligned to two adjacent Chinese words:

$$k = i - (\mathbf{A}(j_{prev}) + (j - j_{prev})N) \tag{3}$$

where j_{prev} is the position of the immediately preceding Chinese word and N is a constant that normalizes for average sentence lengths in different languages. The motivation is that words that are close to each other in the Chinese sentence should tend to be close in the English sentence as well. The size of the parameter set is greatly reduced from the $|i| \times |j| \times |T| \times |V|$ parameters of the alignment probabilities, down to a small set of $|k|$ parameters. However, the search space remains the same.

The A*-style stack-decoding approach is in some ways a carryover from the speech recognition architectures that inspired the channel translation model. It has proven highly effective for speech recognition in both accuracy and speed, where the search space contains no order variation since the acoustic and text streams can be assumed to be linearly aligned. But in contrast, for translation models the stack search alone does not adequately compensate for the combinatorially more complex space that results from permitting arbitrary order variations. Indeed, the stack-decoding approach remains impractically slow for translation, and has not achieved the same kind of speed as for speech recognition.

The model we describe in this paper, like Dagan *et al.*'s model, encourages related words to stay together, and reduces the number of parameters used to describe word-order variation. But more importantly, it makes structural assumptions that eliminate large portions of the space of alignments, based on linguistic motivatations. This greatly reduces the search space and makes possible a polynomial-time optimization algorithm.

2 ITG and BTG Overview

The new translation model is based on the recently introduced *bilingual language modeling* approach. Specifically, the model employs a *bracketing transduction grammar* or BTG (Wu, 1995a), which is a special case of *inversion transduction grammars* or ITGs (Wu, 1995c; Wu, 1995c; Wu, 1995b; Wu, 1995d). These formalisms were originally developed for the purpose of parallel corpus annotation, with applications for bracketing, alignment, and segmentation. This paper finds they are also useful for the translation system itself. In this section we summarize the main properties of BTGs and ITGs.

An ITG consists of context-free productions where terminal symbols come in *couples*, for example x/y, where x is a Chinese word and y is an English translation of x.[2] Any parse tree thus generates two strings, one on the Chinese stream and one on the English stream. Thus, the tree:

(1) [我/I [[拿了/took [一/a 本/ϵ 書/book]$_{\text{NP}}$]$_{\text{VP}}$ [給/for 你/you]$_{\text{PP}}$]$_{\text{VP}}$]$_{\text{S}}$

produces, for example, the mutual translations:

(2) a. [我 [[拿了 [一本書]$_{\text{NP}}$]$_{\text{VP}}$ [給你]$_{\text{PP}}$]$_{\text{VP}}$]$_{\text{S}}$
[Wǒ [[ná le [yī běn shū]$_{\text{NP}}$]$_{\text{VP}}$ [géi nǐ]$_{\text{PP}}$]$_{\text{VP}}$]$_{\text{S}}$
b. [I [[took [a book]$_{\text{NP}}$]$_{\text{VP}}$ [for you]$_{\text{PP}}$]$_{\text{VP}}$]$_{\text{S}}$

An additional mechanism accommodates a conservative degree of word-order variation between the two languages. With each production of the grammar is associated either a *straight* orientation or an *inverted* orientation, respectively denoted as follows:

$$\begin{array}{lcl} \text{VP} & \rightarrow & [\text{VP PP}] \\ \text{VP} & \rightarrow & \langle \text{VP PP} \rangle \end{array}$$

In the case of a production with straight orientation, the right-hand-side symbols are visited left-to-right for both the Chinese and English streams. But for a production with inverted orientation, the

[2] Readers of the papers cited above should note that we have switched the roles of English and Chinese here, which helps simplify the presentation of the new translation algorithm.

f	BTG	all matchings	ratio
0	1	1	1.000
1	1	1	1.000
2	2	2	1.000
3	6	6	1.000
4	22	24	0.917
5	90	120	0.750
6	394	720	0.547
7	1806	5040	0.358
8	8558	40320	0.212
9	41586	362880	0.115
10	206098	3628800	0.057
11	1037718	39916800	0.026
12	5293446	479001600	0.011
13	27297738	6227020800	0.004
14	142078746	87178291200	0.002
15	745387038	1307674368000	0.001
16	3937603038	20922789888000	0.000

Figure 2: Number of legal word alignments between sentences of length f, with and without the BTG restriction.

right-hand-side symbols are visited left-to-right for Chinese and right-to-left for English. Thus, the tree:

(3) [我/I ⟨[給/for 你/you]$_{\mathrm{PP}}$ [拿了/took [一/a 本/ϵ 書/book]$_{\mathrm{NP}}$]$_{\mathrm{VP}}$ ⟩$_{\mathrm{VP}}$]$_{\mathrm{S}}$

produces translations with different word order:

(4) a. [我 [[給你]$_{\mathrm{PP}}$ [拿了 [一本書]$_{\mathrm{NP}}$]$_{\mathrm{VP}}$]$_{\mathrm{VP}}$]$_{\mathrm{S}}$
b. [I [[took [a book]$_{\mathrm{NP}}$]$_{\mathrm{VP}}$ [for you]$_{\mathrm{PP}}$]$_{\mathrm{VP}}$]$_{\mathrm{S}}$

In the special case of BTGs which are employed in the model presented below, there is only one undifferentiated nonterminal category (aside from the start symbol). Designating this category A, this means all non-lexical productions are of one of these two forms:

$$\begin{array}{lll} A & \rightarrow & [A\, A \cdots A] \\ A & \rightarrow & \langle A\, A \cdots A \rangle \end{array}$$

The degree of word-order flexibility is the critical point. BTGs make a favorable trade-off between efficiency and expressiveness: constraints are strong enough to allow algorithms to operate efficiently, but without so much loss of expressiveness as to hinder useful translation. We summarize here; details are given elsewhere (Wu, 1995b).

With regard to efficiency, Figure 2 demonstrates the kind of reduction that BTGs obtain in the space of possible alignments. The number of possible alignments, compared against the unrestricted case where any English word may align to any Chinese position, drops off dramatically for strings longer than four words. (This table makes the simplification of counting only 1-1 matchings and is merely representative.)

With regard to expressiveness, we believe that almost all variation in the order of arguments in a syntactic frame can be accommodated.[3] Syntactic frames generally contain four or fewer subconstituents. Figure 2 shows that for the case of four subconstituents, BTGs permit 22 out of the 24 possible alignments. The only prohibited arrangements are "inside-out" transformations (Wu, 1995b), which we have been unable to find any examples of in our corpus. Moreover, extremely distorted alignments can be handled by BTGs (Wu, 1995c), without resorting to the unrestricted-alignment model.

The translation expressiveness of BTGs is by no means perfect. They are nonetheless proving very useful in applications and are substantially more feasible than previous models. In our previous corpus analysis applications, any expressiveness limitations were easily tolerable since degradation was graceful. In the present translation application, any expressiveness limitation simply means that certain translations are not considered.

For the remainder of the paper, we take advantage of a convenient normal-form theorem (Wu, 1995a) that allows us to assume without loss of generality that the BTG only contains the binary-branching form for the non-lexical productions.[4]

3 BTG-Based Search for the Original Models

A first approach to improving the translation search is to limit the allowed word alignment patterns to those permitted by a BTG. In this case, Equation (2) is kept as the objective function and the translation channel can be parameterized similarly to Dagan *et al.* (Dagan, Church, and Gale, 1993). The effect of the BTG restriction is just to constrain the *shapes* of the word-order distortions. A BTG rather than ITG is used since, as we discussed earlier, pure channel translation models operate without explicit grammars, providing no constituent categories around which a more sophisticated ITG could be structured. But the structural constraints of the BTG can improve search efficiency, even without differentiated constituent categories. Just as in the baseline system, we rely on the language and translation models to take up the slack in place of an explicit grammar. In this approach, an $O(T^7)$ algorithm similar to the one described later can be constructed to replace A* search.

[3] Note that these points are not directed at free word-order languages. But in such languages, explicit morphological inflections make role identification and translation easier.

[4] But see the conclusion for a caveat.

However we do not feel it is worth preserving offset (or alignment or distortion) parameters simply for the sake of preserving the original translation channel model. These parameterizations were only intended to crudely model word-order variation. Instead, the BTG itself can be used directly to probabilistically rank alternative alignments, as described next.

4 Replacing the Channel Model with a SBTG

The second possibility is to use a stochastic bracketing transduction grammar (SBTG) in the channel model, replacing the translation model altogether. In a SBTG, a probability is associated with each production. Thus for the normal-form BTG, we have: The translation lexicon is encoded in productions of

$$
\begin{array}{llll}
A & \xrightarrow{a^{[]}} & [A\,A] & \\
A & \xrightarrow{a^{\langle\rangle}} & \langle A\,A\rangle & \\
A & \xrightarrow{b(x,y)} & x/y & \text{for all } x,y \text{ lexical translations} \\
A & \xrightarrow{b(x,\epsilon)} & x/\epsilon & \text{for all } x \text{ Chinese vocabulary} \\
A & \xrightarrow{b(\epsilon,y)} & \epsilon/y & \text{for all } y \text{ English vocabulary}
\end{array}
$$

the third kind. The latter two kinds of productions allow words of either Chinese or English to go unmatched.

The SBTG assigns a probability $\Pr(\mathbf{c},\mathbf{e},\mathbf{q})$ to all generable trees $\mathbf{q}$ and sentence-pairs. In principle it can be used as the translation channel model by normalizing with $\Pr(\mathbf{e})$ and integrating out $\Pr(\mathbf{q})$ to give $\Pr(\mathbf{c}|\mathbf{e})$ in Equation (2). In practice, a strong language model makes this unnecessary, so we can instead optimize the simpler Viterbi approximation

$$
(4) \qquad \mathbf{e}* \;=\; \operatorname*{argmax}_{\mathbf{e}} \Pr(\mathbf{c},\mathbf{e},\mathbf{q}) \Pr(\mathbf{e})
$$

To complete the picture we add a bigram model $g_{e_{j-1}e_j} = g(e_j|e_{j-1})$ for the English language model $\Pr(\mathbf{e})$.

Offset, alignment, or distortion parameters are entirely eliminated. A large part of the implicit function of such parameters—to prevent alignments where too many frame arguments become separated—is rendered unnecessary by the BTG's structural constraints, which prohibit many such configurations altogether. Another part of the parameters' purpose is subsumed by the SBTG's probabilities $a^{[]}$ and $a^{\langle\rangle}$, which can be set to prefer straight or inverted orientation depending on the language pair. As in the original models, the language model heavily influences the remaining ordering decisions.

Matters are complicated by the presence of the bigram model in the objective function (which word-alignment models, as opposed to translation models, do not need to deal with). As in our word-alignment model, the translation algorithm optimizes Equation (4) via dynamic programming, similar to chart parsing (Earley, 1970) but with a probabilistic objective function as for HMMs (Viterbi, 1967). But unlike the word-alignment model, to accommodate the bigram model we introduce indexes in the recurrence not only on subtrees over the source Chinese string, but also on the delimiting words of the target English substrings.

Another feature of the algorithm is that segmentation of the Chinese input sentence is performed in parallel with the translation search. Conventional architectures for Chinese NLP generally attempt to identify word boundaries as a preprocessing stage.[5] Whenever the segmentation preprocessor prematurely commits to an inappropriate segmentation, difficulties are created for later stages. This problem is particularly acute for translation, since the decision as to whether to regard a sequence as a single unit depends on whether its components can be translated compositionally. This in turn often depends on what the target language is. In other words, the Chinese cannot be appropriately segmented except with respect to the target language of translation—a *task-driven* definition of correct segmentation.

The algorithm is given below. A few remarks about the notation used: $\mathbf{c}_{s..t}$ denotes the subsequence of Chinese tokens $\mathbf{c}_{s+1}, \mathbf{c}_{s+2}, \ldots, \mathbf{c}_t$. We use $E(s..t)$ to denote the set of English words that are translations the Chinese word created by taking all tokens in $\mathbf{c}_{s..t}$ together. $E(s,t)$ denotes the set of English words that are translations of any of the Chinese words anywhere within $\mathbf{c}_{s..t}$. Note also that we assume the explicit sentence-start and sentence-end tokens $\mathbf{c}_0 =$ <s> and $\mathbf{c}_{T+1} =$ </s>, which makes the algorithm description more parsimonious. Finally, the argmax operator is generalized to vector notation to accomodate multiple indices.

1. Initialization

$$
\delta^0_{stYY}(i) \;=\; b_i(\mathbf{c}_{s..t}/Y), \qquad \begin{array}{c} 0 \le s < t \le T \\ Y \in E(s..t) \end{array}
$$

2. Recursion For all s,t,y,z such that $\begin{cases} -1 \le s < t \le T+1 \\ y \in E(s,t) \\ z \in E(s,t) \end{cases}$

$$
\delta_{styz} \;=\; \max[\delta^{[]}_{styz}, \delta^{\langle\rangle}_{styz}, \delta^{0}_{styz}]
$$

$$
\theta_{styz} \;=\; \begin{cases} [] & \text{if } \delta^{[]}_{styz} > \delta^{\langle\rangle}_{styz} \text{ and } \delta^{[]}_{styz} > \delta^{0}_{styz} \\ \langle\rangle & \text{if } \delta^{\langle\rangle}_{styz} > \delta^{[]}_{styz} \text{ and } \delta^{\langle\rangle}_{styz} > \delta^{0}_{styz} \\ 0 & \text{otherwise} \end{cases}
$$

[5] Written Chinese contains no spaces to delimit words; any spaces in the earlier examples are artifacts of the parse tree brackets.

Category	Original A*	Bracket A*	BTG-Channel
Correct	67.5	69.8	68.2
Incorrect	32.5	30.2	31.8

Figure 3: Translation accuracy (percentage correct).

where

$$\delta^{[]}_{styz} = \max_{\substack{s<S<t \\ Y\in E(s,S) \\ Z\in E(S,t)}} a^{[]}\, \delta_{sSyY}\, \delta_{StZz}\, g_{YZ}$$

$$\begin{bmatrix} \sigma^{[]}_{styz} \\ \psi^{[]}_{styz} \\ \omega^{[]}_{styz} \end{bmatrix} = \operatorname*{argmax}_{\substack{s<S<t \\ Y\in E(s,S) \\ Z\in E(S,t)}} a^{[]}\, \delta_{sSyY}\, \delta_{StZz}\, g_{YZ}$$

$$\delta^{\langle\rangle}_{styz} = \max_{\substack{s<S<t \\ Y\in E(S,t) \\ Z\in E(s,S)}} a^{\langle\rangle}\, \delta_{sSZz}\, \delta_{StyY}\, g_{YZ}$$

$$\begin{bmatrix} \sigma^{\langle\rangle}_{styz} \\ \psi^{\langle\rangle}_{styz} \\ \omega^{\langle\rangle}_{styz} \end{bmatrix} = \operatorname*{argmax}_{\substack{s<S<t \\ Y\in E(S,t) \\ Z\in E(s,S)}} a^{\langle\rangle}\, \delta_{sSZz}(j)\, \delta_{StyY}(k)\, g_{YZ}$$

3. Reconstruction Initialize by setting the root of the parse tree to $q_0 = (-1, T-1, \texttt{<s>}, \texttt{</s>})$. The remaining descendants in the optimal parse tree are then given recursively for any $q = (s, t, y, z)$ by:

$$\textsc{Left}(q) = \begin{cases} \text{NIL} & \text{if } t-s \le 1 \\ (s, \sigma^{[]}_q, y, \psi^{[]}_q) & \text{if } \theta_q = [\,] \\ (s, \sigma^{\langle\rangle}_q, \omega^{\langle\rangle}_q, z) & \text{if } \theta_q = \langle\rangle \\ \text{NIL} & \text{otherwise} \end{cases}$$

$$\textsc{Right}(q) = \begin{cases} \text{NIL} & \text{if } t-s \le 1 \\ (\sigma^{[]}_q, t, \omega^{[]}_q, z) & \text{if } \theta_q = [\,] \\ (\sigma^{\langle\rangle}_q, t, y, \psi^{\langle\rangle}_q) & \text{if } \theta_q = \langle\rangle \\ \text{NIL} & \text{otherwise} \end{cases}$$

Assume the number of translations per word is bounded by some constant. Then the maximum size of $E(s,t)$ is proportional to $t-s$. The asymptotic time complexity for the translation algorithm is thus bounded by $O(T^7)$. Note that in practice, actual performance is improved by the sparseness of the translation matrix.

An interesting connection has been suggested to direct parsing for ID/LP grammars (Shieber, 1984), in which word-order variations would be accommodated by the parser, and related ideas for generation of free word-order languages in the TAG framework (Joshi, 1987). Our work differs from the ID/LP work in several important respects. First, we are not merely parsing, but translating with a bigram language model. Also, of course, we are dealing with a probabilistic optimization problem. But perhaps most importantly, our goal is to constrain as tightly as possible the space of possible transduction relationships between two languages with fixed word-order, making no other language-specific assumptions; we are thus driven to seek a kind of language-universal property. In contrast, the ID/LP work was directed at parsing a single language with free word-order. As a consequence, it would be necessary to enumerate a specific set of linear-precedence (LP) relations for the language, and moreover the immediate-dominance (ID) productions would typically be more complex than binary-branching. This significantly increases time complexity, compared to our BTG model. Although it is not mentioned in their paper, the time complexity for ID/LP parsing rises exponentially with the length of production right-hand-sides, due to the number of permutations. ITGs avoid this with their restriction to inversions, rather than permutations, and BTGs further minimize the grammar size. We have also confirmed empirically that our models would not be feasible under general permutations.

5 Results

The algorithm above was tested in the SILC translation system. The translation lexicon was largely constructed by training on the HKUST English-Chinese Parallel Bilingual Corpus, which consists of governmental transcripts. The corpus was sentence-aligned statistically (Wu, 1994); Chinese words and collocations were extracted (Fung and Wu, 1994; Wu and Fung, 1994); then translation pairs were learned via an EM procedure (Wu and Xia, 1995). The resulting English vocabulary is approximately 6,500 words and the Chinese vocabulary is approximately 5,500 words, with a many-to-many translation mapping averaging 2.25 Chinese translations per English word. Due to the unsupervised training, the translation lexicon contains noise and is only at about 86% percent weighted precision.

With regard to accuracy, we merely wish to demonstrate that for statistical MT, accuracy is not significantly compromised by substituting our efficient optimization algorithm. It is not our purpose here to argue that accuracy can be increased with our model. No morphological processing has been used to correct the output, and until now we have only been testing with a bigram model trained on extremely limited samples. A coarse evaluation of

Input: 香港的安定繁榮是我們生活方式的支柱。
(Xiāng gǎng de ān dìng fǎn róng shì wǒ mén shēng huó fāng shì de zhī zhù.)
Output: *Hong Kong's stabilize boom is us life styles's pillar.*
Corpus: Our prosperity and stability underpin our way of life.

Input: 本港的經濟前景與中國, 特別是廣東省的經濟前景息息相關。
(Běn gǎng de jīng jì qían jǐng yǔ zhōng guó, tè bié shì guǎng dōng shěng de jīng jì qián jǐng xī xī xiāng guān.)
Output: *Hong Kong's economic foreground with China, particular Guangdong province's economic foreground vitally interrelated.*
Corpus: Our economic future is inextricably bound up with China, and with Guangdong Province in particular.

Input: 我完全支持他的意見。
(Wǒ wán quán zhī chí tā de yì jiàn.)
Output: *I absolutely uphold his views.*
Corpus: I fully support his views.

Input: 這些安排可加強我們日後維持金融穩定的能力。
(Zhè xiē ān pái kě jiā qiáng wǒ mén rì hòu wéi chí jīn róng wěn dìng de néng lì.)
Output: *These arrangements can enforce us future kept financial stabilization's competency.*
Corpus: These arrangements will enhance our ability to maintain monetary stability in the years to come.

Input: 不過, 我現在可以肯定的說, 我們將會提供為達到各項主要目標所需的經費。
(Bù guò, wǒ xiàn zài kě yǐ kěn dìng de shuō, wǒ mén jiāng huì tí gōng wèi dá dào gè xiàng zhǔ yào mù biāo suǒ xū de jīng fèi.)
Output: *However, I now can certainty's say, will provide for us attain various dominant goal necessary's current expenditure.*
Corpus: The consultation process is continuing but I can confirm now that the necessary funds will be made available to meet the key targets.

Figure 4: Example translation outputs.

translation accuracy was performed on a random sample drawn from Chinese sentences of fewer than 20 words from the parallel corpus, the results of which are shown in Figure 3. We have judged only whether the correct meaning (as determined by the corresponding English sentence in the parallel corpus) is conveyed by the translation, paying particular attention to word order, but otherwise ignoring morphological and function word choices. For comparison, the accuracies from the A*-based systems are also shown. There is no significant difference in the accuracy. Some examples of the output are shown in Figure 4.

On the other hand, the new algorithm has indeed proven to be much faster. At present we are unable to use direct measurement to compare the speed of the systems meaningfully, because of vast implementational differences between the systems. However, the order-of-magnitude improvements are immediately apparent. In the earlier system, translation of single sentences required on the order of hours (Sun Sparc 10 workstations). In contrast the new algorithm generally takes less than one minute—usually substantially less—with no special optimization of the code.

6 Conclusion

We have introduced a new algorithm for the runtime optimization step in statistical machine translation systems, whose polynomial-time complexity addresses one of the primary obstacles to practicality facing statistical MT. The underlying model for the algorithm is a combination of the stochastic BTG and bigram models. The improvement in speed does not appear to impair accuracy significantly.

We have implemented a version that accepts ITGs rather than BTGs, and plan to experiment with more heavily structured models. However, it is important to note that the search complexity rises exponentially rather than polynomially with the size of the grammar, just as for context-free parsing (Barton, Berwick, and Ristad, 1987). This is not relevant to the BTG-based model we have described since its grammar size is fixed; in fact the BTG's minimal grammar size has been an important advantage over more linguistically-motivated ITG-based models.

We have also implemented a generalized version that accepts arbitrary grammars not restricted to normal form, with two motivations. The pragmatic benefit is that structured grammars become easier to write, and more concise. The expressiveness benefit is that a wider family of probability distributions can be written. As stated earlier, the normal form theorem guarantees that the same set of shapes will be explored by our search algorithm, regardless of whether a binary-branching BTG or an arbitrary BTG is used. But it may sometimes be useful to place probabilities on n-ary productions that vary with n in a way that cannot be expressed by composing binary productions; for example one might wish to encourage longer straight productions. The generalized version permits such strategies.

Currently we are evaluating robustness extensions of the algorithm that permit words suggested by the language model to be inserted in the output sentence, which the original A* algorithms permitted.

Acknowledgements

Thanks to an anonymous referee for valuable comments, and to the SILC group members: Xuanyin Xia, Eva Wai-Man Fong, Cindy Ng, Hong-sing Wong, and Daniel Ka-Leung Chan. Many thanks also to Kathleen McKeown and her group for discussion, support, and assistance.

References

Barton, G. Edward, Robert C. Berwick, and Eric Sven Ristad. 1987. *Computational Complexity and Natural Language.* MIT Press, Cambridge, MA.

Brown, Peter F., John Cocke, Stephen A. DellaPietra, Vincent J. DellaPietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):29–85.

Brown, Peter F., Stephen A. DellaPietra, Vincent J. DellaPietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Dagan, Ido, Kenneth W. Church, and William A. Gale. 1993. Robust bilingual word alignment for machine aided translation. In *Proceedings of the Workshop on Very Large Corpora*, pages 1–8, Columbus, OH, June.

Earley, Jay. 1970. An efficient context-free parsing algorithm. *Communications of the Association for Computing Machinery*, 13(2):94–102.

Fung, Pascale and Dekai Wu. 1994. Statistical augmentation of a Chinese machine-readable dictionary. In *Proceedings of the Second Annual Workshop on Very Large Corpora*, pages 69–85, Kyoto, August.

Joshi, Aravind K. 1987. Word-order variation in natural language generation. In *Proceedings of AAAI-87, Sixth National Conference on Artificial Intelligence*, pages 550–555.

Shieber, Stuart M. 1984. Direct parsing of ID/LP grammars. *Linguistics and Philosophy*, 7:135–154.

Viterbi, Andrew J. 1967. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, 13:260–269.

Wu, Dekai. 1994. Aligning a parallel English-Chinese corpus statistically with lexical criteria. In *Proceedings of the 32nd Annual Conference of the Association for Computational Linguistics*, pages 80–87, Las Cruces, New Mexico, June.

Wu, Dekai. 1995a. An algorithm for simultaneously bracketing parallel texts by aligning words. In *Proceedings of the 33rd Annual Conference of the Association for Computational Linguistics*, pages 244–251, Cambridge, Massachusetts, June.

Wu, Dekai. 1995b. Grammarless extraction of phrasal translation examples from parallel texts. In *TMI-95, Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation*, volume 2, pages 354–372, Leuven, Belgium, July.

Wu, Dekai. 1995c. Stochastic inversion transduction grammars, with application to segmentation, bracketing, and alignment of parallel corpora. In *Proceedings of IJCAI-95, Fourteenth International Joint Conference on Artificial Intelligence*, pages 1328–1334, Montreal, August.

Wu, Dekai. 1995d. Trainable coarse bilingual grammars for parallel text bracketing. In *Proceedings of the Third Annual Workshop on Very Large Corpora*, pages 69–81, Cambridge, Massachusetts, June.

Wu, Dekai and Pascale Fung. 1994. Improving Chinese tokenization with linguistic filters on statistical lexical acquisition. In *Proceedings of the Fourth Conference on Applied Natural Language Processing*, pages 180–181, Stuttgart, October.

Wu, Dekai and Cindy Ng. 1995. Using brackets to improve search for statistical machine translation. In *PACLIC-10, Pacific Asia Conference on Language, Information and Computation*, pages 195–204, Hong Kong, December.

Wu, Dekai and Xuanyin Xia. 1995. Large-scale automatic extraction of an English-Chinese lexicon. *Machine Translation*, 9(3–4):285–313.