Unsupervised Transduction Grammar Induction via Minimum Description Length

Markus Saers and **Karteek Addanki** and **Dekai Wu**
Human Language Technology Center
Dept. of Computer Science and Engineering
Hong Kong University of Science and Technology
{masaers|vskaddanki|dekai}@cs.ust.hk

Abstract

We present a minimalist, unsupervised learning model that induces relatively clean phrasal inversion transduction grammars by employing the minimum description length principle to drive search over a space defined by two opposing extreme types of ITGs. In comparison to most current SMT approaches, the model learns a very parsimonious phrase translation lexicons that provide an obvious basis for generalization to abstract translation schemas. To do this, the model maintains internal consistency by avoiding use of mismatched or unrelated models, such as word alignments or probabilities from IBM models. The model introduces a novel strategy for avoiding the pitfalls of premature pruning in chunking approaches, by incrementally splitting an ITG while using a second ITG to guide this search.

1 Introduction

We introduce an unsupervised approach to inducing parsimonious, relatively clean phrasal inversion transduction grammars or ITGs (Wu, 1997) that employs a theoretically well-founded minimum description length (MDL) objective to explicitly drive two opposing, extreme ITGs towards one minimal ITG. This represents a new attack on the problem suffered by most current SMT approaches of learning phrase translations that require enormous amounts of run-time memory, contain a high degree of redundancy, and fails to provide an obvious basis for generalization to abstract translation schemas. In particular, phrasal SMT models such as Koehn *et al.* (2003) and Chiang (2005) often search for candidate translation segments and transduction rules by committing to a word alignment based on very different assumptions (Brown *et al.*, 1993; Vogel *et al.*, 1996), and heuristically derive lexical segment translations (Och and Ney, 2003). In fact, it is possible to improve the performance by tossing away most of the learned segmental translations (Johnson *et al.*, 2007). In addition to preventing such wastefulness, our work aims to also provide an obvious basis for generalization to abstract translation schemas by driving the search for phrasal rules by simultaneously using two opposing types of ITG constraints that have both individually been empirically proven to match phrase reordering patterns across translations well.

We adopt a more "pure" methodology for evaluating transduction grammar induction than typical system building papers. Instead of embedding our learned ITG in the midst of many other heuristic components for the sake of a short term boost in BLEU, we focus on scientifically understanding the behavior of pure MDL-based search for phrasal translations, divorced from the effect of other variables, even though BLEU is naturally much lower this way. The common practice of plugging some aspect of a learned ITG into either (a) a long pipeline of training heuristics and/or (b) an existing decoder that has been patched up to compensate for earlier modeling mistakes, as we and others have done before—see for example Cherry and Lin (2007); Zhang *et al.* (2008); Blunsom *et al.* (2008, 2009); Haghighi *et al.* (2009); Saers and Wu (2009, 2011); Blunsom and Cohn (2010); Burkett *et al.* (2010); Riesa and Marcu (2010); Saers *et al.* (2010); Neubig *et al.* (2011, 2012)—obscures the specific traits of the induced grammar. Instead, we directly use our learned ITG in translation mode (any transduction grammar also represents a decoder when parsing with the input sentence as a hard constraint) which allows us to see exactly which aspects of correct translation the transduction rules have captured.

Proceedings of the Second Workshop on Hybrid Approaches to Translation, pages 67–73,
Sofia, Bulgaria, August 8, 2013. ©2013 Association for Computational Linguistics

When the structure of an ITG is induced without supervision, it has so far been assumed that smaller rules get clumped together into larger rules. This is a natural way to search, since maximum likelihood (ML) tends to improve with longer rules, which is typically balanced with Bayesian priors (Zhang *et al.*, 2008). Bayesian priors are also used in Gibbs sampling (Blunsom *et al.*, 2008, 2009; Blunsom and Cohn, 2010), as well as other nonparametric learning methods (Neubig *et al.*, 2011, 2012). All of the above evaluate their models by feeding them into mismatched decoders, making it hard to evaluate how accurate the learned models themselves were. In this work we take a radically different approach, and start with the longest rules possible and attempt to segment them into shorter rules iteratively. This makes ML useless, since our initial model maximizes it. Instead, we balance the ML objective with a minimum description length (MDL) objective, which let us escape the initial ML optimum by rewarding *model parsimony*.

Transduction grammars can also be induced with supervision from treebanks, which cuts down the search space by enforcing external constraints (Galley *et al.*, 2006). This complicates the learning process by adding external constraints that are bound to match the translation model poorly. It does, however, constitute a way to borrow nonterminal categories that help the translation model.

MDL has been used before in monolingual grammar induction (Grünwald, 1996; Stolcke and Omohundro, 1994), as well as to interpret visual scenes (Si *et al.*, 2011). Our work is markedly different in that we (a) induce an ITG rather than a monolingual grammar, and (b) focus on learning the terminal segments rather than the nonterminal categories. Iterative segmentation has also been used before, but only to derive a word alignment as part of a larger pipeline (Vilar and Vidal, 2005).

The paper is structured as follows: we start by describing the MDL principle (Section 2). We then describe the initial ITGs (Section 3), followed by the algorithm that induces an MDL-optimal ITG from them (Section 4). After that we describe the experiments (Section 5), and the results (Section 6). Finally, we offer some conclusions (Section 7).

2 Minimum description length

The minimum description length principle is about finding the optimal balance between the size of a model and the size of some data given the model (Solomonoff, 1959; Rissanen, 1983). Consider the information theoretical problem of encoding some data with a model, and then sending both the encoded data *and* the information needed to decode the data (the model) over a channel; the minimum description length is the minimum number of bits sent over the channel. The encoded data can be interpreted as carrying the information necessary to disambiguate the uncertainties that the model has about the data. The model can *grow in size* and become *more certain* about the data, and it can *shrink in size* and become *more uncertain* about the data. Formally, description length (DL) is:

$$\mathrm{DL}\left(\Phi, D\right) = \mathrm{DL}\left(D|\Phi\right) + \mathrm{DL}\left(\Phi\right)$$

where Φ is the model and D is the data.

In practice, we rarely have complete data to train on, so we need our models to generalize to unseen data. A model that is very certain about the training data runs the risk of not being able to generalize to new data: it is over-fitting. It is bad enough when estimating the parameters of a transduction grammar, and catastrophic when inducing the structure.

The information-theoretic view of the problem gives a hint at the operationalization of description length of a corpus given a grammar. Shannon (1948) stipulates that we can get a lower bound on the number of bits required to encode a specific outcome of a random variable. We thus define description length of the corpus given the grammar to be: $\mathrm{DL}\left(D|\Phi\right) = -\mathrm{lg}P\left(D|\Phi\right)$

Information theory is also useful for the description length of the grammar: if we can find a way to serialize the grammar into a sequence of tokens, we can figure out how that sequence can be optimally encoded. To serialize an ITG, we first need to determine the alphabet that the message will be written in. We need one symbol for every nonterminal, L_0- and L_1-terminal. We will also make the assumption that all these symbols are used in at least one rule, so that it is sufficient to serialize the rules in order to express the entire ITG. We serialize a rule with a type marker, followed by the left-hand side nonterminal, followed by all the right-hand side symbols. The type marker is either $[]$ denoting the start of a straight rule, or $\langle\rangle$ denoting the start of an inverted rule. Unary rules are considered to be straight. We serialize the ITG by concatenating the serialized form of all the rules, assuming that each symbol can be serialized into $-\mathrm{lg}c$ bits where c is the symbol's relative frequency in the serialized form of the ITG.

3 Initial ITGs

To tackle the exponential problem of searching for an ITG that minimizes description length, it is useful to contrast two extreme forms of ITGs. Description length has two components, model length and data length. We call an ITG that minimizes the data at the expense of the model a **long ITG**; we call an ITG that minimizes the model at the expense of the data a **short ITG**.[1] The long ITG simply has all the sentence pairs as biterminals:

$$\begin{aligned} S &\rightarrow A \\ A &\rightarrow e_{0..T_0}/f_{0..V_0} \\ A &\rightarrow e_{0..T_1}/f_{0..V_1} \\ &\dots \\ A &\rightarrow e_{0..T_N}/f_{0..V_N} \end{aligned}$$

where S is the start symbol, A is the nonterminal, N is the number of sentence pairs, T_i is the length of the i^{th} output sentence (making $e_{0..T_i}$ the i^{th} output sentence), and V_i is the length of the i^{th} input sentence (making $f_{0..V_i}$ the i^{th} input sentence). The short ITG is a token-based bracketing ITG:

$$\begin{gathered} S \rightarrow A, \quad A \rightarrow [AA], \quad A \rightarrow \langle AA \rangle, \\ A \rightarrow e/f, \quad A \rightarrow e/\epsilon, \quad A \rightarrow \epsilon/f \end{gathered}$$

where, S is the start symbol, A is the nonterminal symbol, e is an L_0-token, f is an L_1-token, and ϵ is the empty sequence of tokens.

4 Shortening the long ITG

To shorten the long ITG, we will identify good split candidates in the terminal rules by parsing them with the short ITG, and commit to split candidates that give a net gain. A split candidate is an existing long terminal rule, information about where to split its right-hand side, and whether to invert the resulting two rules or not. Consider the terminal rule $A \rightarrow e_{s..t}/f_{u..v}$; it can be split at any point S in L_0 and any point U in L_1, giving the three rules $A \rightarrow [AA]$, $A \rightarrow e_{s..S}/f_{u..U}$ and $A \rightarrow e_{S..t}/f_{U..v}$ when it is split in straight order, and the three rules $A \rightarrow \langle AA \rangle$, $A \rightarrow e_{s..S}/f_{U..v}$ and $A \rightarrow e_{S..t}/f_{u..U}$ when it is split in inverted order. We will refer to the original long rule as r_0, and the resulting three rules as r_1, r_2 and r_3.

To identify the split candidates and to figure out how the probability mass of r_0 is to be distributed to the new rules, we use the short ITG to biparse the right-hand side of r_0. The distribution is derived from the inside probability of the bispans that the new rules are covering in the chart, and we refer to them as λ_1, λ_2 and λ_3, where the index indicates which new rule they apply to. This has the effect of preferring to split a rule into parts that are roughly equally probable, as the size of the data is minimized when the weights are equal.

Algorithm 1 Rule shortening.

```
$G_l$                                      ▷ The long ITG
$G_s$                                      ▷ The short ITG
repeat
  $cands \leftarrow collect\_candidates(G_l, G_s)$
  $\delta \leftarrow 0$
  $removed \leftarrow \{\}$
  repeat
    $score(cands)$
    $sort\_by\_delta(cands)$
    for all $c \in cands$ do
      $r \leftarrow original\_rule(c)$
      if $r \notin removed$ and $\delta_c \leq 0$ then
        $G_l \leftarrow update\_grammar(G_l, c)$
        $removed \leftarrow \{r\} \cup removed$
        $\delta \leftarrow \delta + \delta_c$
      end if
    end for
  until $\delta \geq 0$
until $\delta \geq 0$
return $G_l$
```

To choose which split candidates to commit to, we need a way to estimate their impact on the total MDL score of the model. This breaks down into two parts: the difference in description length of the grammar: $\mathrm{DL}(\Phi') - \mathrm{DL}(\Phi)$ (where Φ' is Φ after committing to the split candidate), and the difference in description length of the corpus given the grammar: $\mathrm{DL}(D|\Phi') - \mathrm{DL}(D|\Phi)$. The two are added up to get the total change in description length.The difference in grammar length is calculated as described in Section 2. The difference in description length of the corpus given the grammar can be calculated by biparsing the corpus, since $\mathrm{DL}(D|\Phi') = -\mathrm{lg}P(D|p')$ and $\mathrm{DL}(D|\Phi) = -\mathrm{lg}P(D|p)$ where p' and p are the rule probability functions of Φ' and Φ respectively. Biparsing is, however, a very costly process that we do not want to carry out for every candidate. Instead, we assume that we have the original corpus probability (through biparsing when generating the can-

[1] Long and short ITGs correspond well to *ad-hoc* and promiscuous grammars in Grünwald (1996).

Table 1: The results of decoding. NIST and BLEU are the translation scores at each iteration, followed by the number of rules in the grammar, followed by the average (as measured by mean and mode) number of English tokens in the rules.

Iteration	NIST	BLEU	Rules	Mean	Mode
1	2.7015	11.97	43,704	7.20	6
2	4.0116	14.04	42,823	6.30	6
3	4.1654	16.58	41,867	5.68	2
4	**4.3723**	17.43	40,953	5.23	1
5	4.2032	**18.78**	40,217	4.97	1
6	4.1329	17.28	39,799	4.84	1
7	4.0710	17.31	39,587	4.79	1
8	4.0437	17.10	39,470	4.75	1

didates), and estimate the new corpus probability from it (in closed form). The new rule probability function p' is identical to p, except that:

$$\begin{aligned} p'(r_0) &= 0 \\ p'(r_1) &= p(r_1) + \lambda_1 p(r_0) \\ p'(r_2) &= p(r_2) + \lambda_2 p(r_0) \\ p'(r_3) &= p(r_3) + \lambda_3 p(r_0) \end{aligned}$$

We assume the probability of the corpus given this new rule probability function to be:

$$P(D|p') = P(D|p)\,\frac{p'(r_1)\,p'(r_2)\,p'(r_3)}{p(r_0)}$$

This gives the following description length difference:

$$\begin{gathered} \mathrm{DL}(D|\Phi') - \mathrm{DL}(D|\Phi) = \\ -\mathrm{lg}\tfrac{p'(r_1)p'(r_2)p'(r_3)}{p(r_0)} \end{gathered}$$

We will commit to all split candidates that are estimated to lower the DL, restricting it so that any original rule is split only in the best way (Algorithm 1).

5 Experimental setup

To test whether minimum description length is a good driver for unsupervised inversion transduction induction, we implemented and executed the method described above. We start by initializing one long and one short ITG. The parameters of the long ITG cannot be adjusted to fit the data better, but the parameters of the short ITG can be tuned to the right-hand sides of the long ITG. We do so with an implementation of the cubic time algorithm described in Saers *et al.* (2009), with a beam width of 100. We then run the introduced algorithm.

As training data, we use the IWSLT07 Chinese–English data set (Fordyce, 2007), which contains 46,867 sentence pairs of training data, and 489 Chinese sentences with 6 English reference translations each as test data; all the sentences are taken from the traveling domain. Since the Chinese is written without whitespace, we use a tool that tries to clump characters together into more "word like" sequences (Wu, 1999).

After each iteration, we use the long ITG to translate the held out test set with our in-house ITG decoder. The decoder uses a CKY-style parsing algorithm (Cocke, 1969; Kasami, 1965; Younger, 1967) and cube pruning (Chiang, 2007) to integrate the language model scores. The decoder builds an efficient hypergraph structure which is scored using both the induced grammar and a language model. We use SRILM (Stolcke, 2002) for training a trigram language model on the English side of the training corpus. To evaluate the resulting translations, we use BLEU (Papineni *et al.*, 2002) and NIST (Doddington, 2002).

We also perform a combination experiment, where the grammar at different stages of the learning process (iterations) are interpolated with each other. This is a straight-forward linear interpolation, where the probabilities of the rules are added up and the grammar is renormalized. Although it makes little sense from an MDL point of view to increase the size of the grammar so indiscriminately, it does make sense from an engineering point of view, since more rules typically means better coverage, which in turn typically means better translations of unknown data.

6 Results

As discussed at the outset, rather than burying our learned ITG in many layers of unrelated heuristics just to push up the BLEU score, we think it is more

Table 2: The results of decoding with combined grammars. NIST and BLEU are the translation scores for each combination, followed by the number of rules in the grammar, followed by the average (as measured by mean and mode) number of English tokens in the rules.

Combination	NIST	BLEU	Rules	Mean	Mode
1–2 (2 grammars)	4.2426	15.28	74,969	6.69	6
3–4 (2 grammars)	4.5087	18.75	54,533	5.41	3
5–6 (2 grammars)	4.1897	18.19	44,264	4.86	1
7–8 (2 grammars)	4.0953	17.40	40,785	4.79	1
1–4 (4 grammars)	**4.9234**	19.98	109,183	6.19	5
5–8 (4 grammars)	4.1089	17.86	47,504	4.84	1
1–8 (8 grammars)	4.8649	**20.41**	124,423	5.92	3

important to illuminate scientific understanding of the behavior of pure MDL-driven rule induction without interference from other variables. Directly evaluating solely the ITG in translation mode—instead of (a) deriving word alignments from it by committing to only the one-best parse, but then discarding any trace of structure and/or (b) evaluating it through a decoder that has been patched up to compensate for deficiencies in disparate aspects of translation—allows us to see exactly how accurate the learned transduction rules are.

The results from the individual iterations (Table 1) show that we learn very parsimonious models that far outperforms the only other result we are aware of where an ITG is tested exactly as it was learned without altering the model itself: Saers *et al.* (2012) induce a pure ITG by iteratively chunking rules, but they report significantly lower translation quality (8.30 BLEU and 0.8554 NIST) despite a significantly larger ITG (251,947 rules). The average rule length also decreases as smaller reusable spans are found. The English side of the training data has a mean of 8.45 and a mode of 7 tokens per sentence, and these averages drop steadily during training. It is very encouraging to see the mode drop to one so quickly, as this indicates that the learning algorithm finds translations of individual English words. Not only are the rules getting fewer, but they are also getting shorter.

The results from the combination experiments (Table 2) corroborate the engineering intuition that more rules give better translations at the expense of a larger model. Using all eight grammars gives a BLEU score of 20.41, at the expense of approximately tripling the size of the grammar. All individual iterations benefit from being combined with other iterations—but for the very best iterations more additional data is needed to get this improvement; the fifth iteration, which excelled at BLEU score needs to be combined with all other iterations to see an improvement, whereas the first and second iterations only need each other to see an improvement.

7 Conclusions

We have presented a minimalist, unsupervised learning model that induces relatively clean phrasal ITGs by iteratively splitting existing rules into smaller rules using a theoretically well-founded minimum description length objective. The resulting translation model is very parsimonious and provide an obvious foundation for generalization to more abstract transduction grammars with informative nonterminals.

8 Acknowledgements

This material is based upon work supported in part by the Defense Advanced Research Projects Agency (DARPA) under BOLT contract no. HR0011-12-C-0016, and GALE contract nos. HR0011-06-C-0022 and HR0011-06-C-0023; by the European Union under the FP7 grant agreement no. 287658; and by the Hong Kong Research Grants Council (RGC) research grants GRF620811, GRF621008, and GRF612806. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA, the EU, or RGC.

References

Phil Blunsom and Trevor Cohn. Inducing synchronous grammars with slice sampling. In *HLT/NAACL2010*, pages 238–241, Los Angeles, California, June 2010.

Phil Blunsom, Trevor Cohn, and Miles Osborne. Bayesian synchronous grammar induction. In *Proceedings of NIPS 21*, Vancouver, Canada, December 2008.

Phil Blunsom, Trevor Cohn, Chris Dyer, and Miles Osborne. A gibbs sampler for phrasal synchronous grammar induction. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 782–790, Suntec, Singapore, August 2009.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The Mathematics of Machine Translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1993.

David Burkett, John Blitzer, and Dan Klein. Joint parsing and alignment with weakly synchronized grammars. In *HLT/NAACL'10*, pages 127–135, Los Angeles, California, June 2010.

Colin Cherry and Dekang Lin. Inversion transduction grammar for joint phrasal translation modeling. In *Proceedings of SSST'07*, pages 17–24, Rochester, New York, April 2007.

David Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL'05*, pages 263–270, Ann Arbor, Michigan, June 2005.

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

John Cocke. *Programming languages and their compilers: Preliminary notes*. Courant Institute of Mathematical Sciences, New York University, 1969.

George Doddington. Automatic evaluation of machine translation quality using n-gram cooccurrence statistics. In *Proceedings of HLT'02*, pages 138–145, San Diego, California, 2002.

C. S. Fordyce. Overview of the IWSLT 2007 evaluation campaign. In *Proceedings IWSLT'07*, pages 1–12, 2007.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. Scalable inference and training of context-rich syntactic translation models. In *Proceedings COLING/ACL'06*, pages 961–968, Sydney, Australia, July 2006.

Peter Grünwald. A minimum description length approach to grammar inference in symbolic. *Lecture Notes in Artificial Intelligence*, (1040):203–216, 1996.

Aria Haghighi, John Blitzer, John DeNero, and Dan Klein. Better word alignments with supervised itg models. In *Proceedings of ACL/IJCNLP'09*, pages 923–931, Suntec, Singapore, August 2009.

Howard Johnson, Joel Martin, George Foster, and Roland Kuhn. Improving translation quality by discarding most of the phrasetable. In *Proceedings EMNLP/CoNLL'07*, pages 967–975, Prague, Czech Republic, June 2007.

Tadao Kasami. An efficient recognition and syntax analysis algorithm for context-free languages. Technical Report AFCRL-65-00143, Air Force Cambridge Research Laboratory, 1965.

Philipp Koehn, Franz Joseph Och, and Daniel Marcu. Statistical Phrase-Based Translation. In *Proceedings of HLT/NAACL'03*, volume 1, pages 48–54, Edmonton, Canada, May/June 2003.

Graham Neubig, Taro Watanabe, Eiichiro Sumita, Shinsuke Mori, and Tatsuya Kawahara. An unsupervised model for joint phrase alignment and extraction. In *Proceedings of ACL/HLT'11*, pages 632–641, Portland, Oregon, June 2011.

Graham Neubig, Taro Watanabe, Shinsuke Mori, and Tatsuya Kawahara. Machine translation without words through substring alignment. In *Proceedings of ACL'12*, pages 165–174, Jeju Island, Korea, July 2012.

Franz Josef Och and Hermann Ney. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51, 2003.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL'02*, pages 311–318, Philadelphia, Pennsylvania, July 2002.

Jason Riesa and Daniel Marcu. Hierarchical search for word alignment. In *Proceedings of ACL'10*, pages 157–166, Uppsala, Sweden, July 2010.

Jorma Rissanen. A universal prior for integers and estimation by minimum description length. *The Annals of Statistics*, 11(2):416–431, June 1983.

Markus Saers and Dekai Wu. Improving phrase-based translation via word alignments from Stochastic Inversion Transduction Grammars. In *Proceedings of SSST'09*, pages 28–36, Boulder, Colorado, June 2009.

Markus Saers and Dekai Wu. Principled induction of phrasal bilexica. In *Proceedings of EAMT'11*, pages 313–320, Leuven, Belgium, May 2011.

Markus Saers, Joakim Nivre, and Dekai Wu. Learning stochastic bracketing inversion transduction grammars with a cubic time biparsing algorithm. In *Proceedings of IWPT'09*, pages 29–32, Paris, France, October 2009.

Markus Saers, Joakim Nivre, and Dekai Wu. Word alignment with stochastic bracketing linear inversion transduction grammar. In *Proceedings of HLT/NAACL'10*, pages 341–344, Los Angeles, California, June 2010.

Markus Saers, Karteek Addanki, and Dekai Wu. From finite-state to inversion transductions: Toward unsupervised bilingual grammar induction. In *Proceedings of COLING 2012: Technical Papers*, pages 2325–2340, Mumbai, India, December 2012.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, July, October 1948.

Zhangzhang Si, Mingtao Pei, Benjamin Yao, and Song-Chun Zhu. Unsupervised learning of event and-or grammar and semantics from video. In *Proceedings of the 2011 IEEE ICCV*, pages 41–48, November 2011.

Ray J. Solomonoff. A new method for discovering the grammars of phrase structure languages. In *IFIP Congress*, pages 285–289, 1959.

Andreas Stolcke and Stephen Omohundro. Inducing probabilistic grammars by bayesian model merging. In R. C. Carrasco and J. Oncina, editors, *Grammatical Inference and Applications*, pages 106–118. Springer, 1994.

Andreas Stolcke. SRILM – an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, pages 901–904, Denver, Colorado, September 2002.

Juan Miguel Vilar and Enrique Vidal. A recursive statistical translation model. In *ACL-2005 Workshop on Building and Using Parallel Texts*, pages 199–207, Ann Arbor, Jun 2005.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. HMM-based Word Alignment in Statistical Translation. In *Proceedings of COLING-96*, volume 2, pages 836–841, 1996.

Dekai Wu. Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. *Computational Linguistics*, 23(3):377–403, 1997.

Zhibiao Wu. LDC Chinese segmenter, 1999.

Daniel H. Younger. Recognition and parsing of context-free languages in time n^3. *Information and Control*, 10(2):189–208, 1967.

Hao Zhang, Chris Quirk, Robert C. Moore, and Daniel Gildea. Bayesian learning of non-compositional phrases with synchronous parsing. In *Proceedings of ACL/HLT'08*, pages 97–105, Columbus, Ohio, June 2008.