An Open-Source Hierarchical Phrase-Based Translation System

Lane Schwartz
Department of Computer Science & Engineering
University of Minnesota
Minneapolis, MN 55455, USA
lane@cs.umn.edu

Abstract

We present an open source translation system that provides a clean-room implementation of the hierarchical phrase-based statistical translation model introduced in (Chiang, 2005) and refined in (Chiang, 2007). To our knowledge this is the first freely available hierarchical phrase-based translation system which implements cube pruning. We introduce extensions to (Chiang, 2007) to take advantage of multiple source languages.

1 Introduction

While the area of statistical machine translation is very active, there are few generally available tools available to researchers, and even fewer open-source tools. For researchers specifically interested in hierarchical phrase-based statistical translation methods, there is no freely available implementation of Chiang (2007) on which to build. In this paper we provide a brief overview of existing tools, and present the first freely available hierarchical phrase-based statistical translation tool which implements cube-pruning. Finally, we introduce a new technique for improving translation results when multiple source languages are available.

Moses (Koehn et al., 2007) and Phramer (Olteanu et al., 2006) provide open-source re-implementations of the non-hierarchical phrase-based Pharaoh system (Koehn, 2004). Zollmann and Venugopal (2006) present an open source syntax-augmented hierarchical phrase-based system written in C++; their system includes a Chiang (2005) compatibility mode, but does not implement cube pruning (Chiang, 2007). Hiero, the system presented in Chiang (2007), is a hierarchical phrase-based system which implements cube pruning, but is not generally available and is not open source. Cubit (Huang and Chiang, 2007) is an open source reference implementation of just the cube pruning algorithm which requires a separately trained Pharaoh-style phrase table.

Our system is a clean-room implementation of the hierarchical phrase-based statistical translation model introduced in Chiang (2005) and refined in Chiang (2007). The system implements all three language model integration techniques from Chiang (2007), including cube pruning. This system is implemented in Java, and was designed to be easily extended. The software is released under the GNU General Public License (GPL); code and documentation are available at `http://sf.net/projects/nlp-parsers`.

The remainder of this paper is structured as follows. Section 2 briefly reviews the hierarchical translation model originally presented in Chiang (2007). Section 3 describes how decoders which implement this model can produce n-best lists of translations, using the framework introduced in Huang and Chiang (2005). Finally, section 4 presents ongoing research which extends this model by using multiple source languages when translating.

2 Model

Our model is trained on a sentence-aligned parallel corpus. Word alignments are extracted for each parallel sentence in the corpus using GIZA++ (Och and Ney, 2000) and refined using the “final-and” method of Koehn et al. (2003).

Following Chiang (2007), our model is a

$P(\gamma \mid \alpha)$	$P(\alpha \mid \gamma)$	$P(\gamma_w \mid \alpha_w)$	$P(\alpha_w \mid \gamma_w)$
$P_{lm}(\alpha)$	λ_{glue}	λ_{phrase}	λ_{word}

Figure 1: Log-linear features

weighted synchronous context-free grammar where the only nonterminals are X, S, and S′. X is by far the most prevalent nonterminal in the grammar; synchronous context-free rules of the form $X \rightarrow \langle \gamma, \alpha, \sim \rangle$ are extracted automatically from the word-aligned sentence pairs, following the process and restrictions of Chiang (2005). In addition to the extracted rules, the grammar includes the following rules used to combine sub-translations:

$$\begin{aligned} S &\rightarrow \langle S_{\textcircled{1}} X_{\textcircled{2}}, S_{\textcircled{1}} X_{\textcircled{2}} \rangle & (1) \\ S &\rightarrow \langle X_{\textcircled{1}}, X_{\textcircled{1}} \rangle & (2) \\ S' &\rightarrow \langle S_{\textcircled{1}}, \langle s \rangle \; S_{\textcircled{1}} \langle /s \rangle \rangle & (3) \end{aligned}$$

The nonterminal S represents the left-hand side of the two glue rules, (1) and (2), which which can be used to combine partial translations serially rather than hierarchically. S′ is the start symbol; rule (3) is used to enclose a complete translation S with beginning and end of sentence tags.

The weight of an extracted synchronous context-free rule is defined as a log-linear combination of weighted features:

$$w(X \rightarrow \langle \gamma, \alpha \rangle) = \sum_i \phi_i(X \rightarrow \langle \gamma, \alpha \rangle) \times \lambda_i \quad (4)$$

We use the feature set defined in Chiang (2005), listed in figure 1. We estimate the rule-specific feature values ϕ for each rule using relative frequency estimation. Each log-linear feature has a corresponding weight λ. All feature values are in log domain. Rules (2) and (3) are each defined to have log-domain weight of zero. The log-domain weight of glue rule (1) is defined to be $-\lambda_{glue}$.

Given a set of synchronous context-free rules, our decoder uses a variant CKY algorithm to parse input sentences. The parse chart then represents a shared forest of all possible translations that the decoder could produce from the rule set. Each complete derivation in the parse chart represents a parse tree from that shared forest. The weight of a derivation is the sum of the weights for each rule used in the derivation:

$$w(D) = \sum_{r \in D} w(r) \quad (5)$$

The best derivation is obtained by simply selecting the derivation with the highest weight, of those derivations which completely span the source language input:

$$\hat{D} = \arg\max_{D} \; w(D) \quad (6)$$

Because our rules are synchronous, during parsing we must store the target language right-hand side of each rule as it is applied. The target language translation for a given derivation can then be extracted by tracing through the rule applications used to construct a derivation.

3 Training

The model above requires meaningful log-linear feature weights. The quality of translations resulting from the decoder is highly dependent on the log-linear feature weights. Meaningful feature weights can be obtained by performing minimum error rate training (Och, 2003). Minimum error rate training attempts to optimize the BLEU score of a development set of sentences by tuning the log-linear feature weights of the model.

The minimum error rate training process requires that the decoder be capable of producing an n-best list of translations for each input sentence. Huang and Chiang (2005) show how an n-best list of derivations can be obtained from a weighted, directed hypergraph. By organizing our parse chart as a hypergraph, we are able to use the efficient algorithm 2 of Huang and Chiang (2005) to extract an n-best list of translations for each source sentence.

We therefore view the parse chart produced by our decoder as a weighted, directed hypergraph. Each chart cell entry is a vertex in the hypergraph. Vertices are connected via hyperarcs, where each hyperarc corresponds to a rule application. The weight of a hyperarc is the weight of the rule associated with that hyperarc, plus the weights of the tail vertices of the hyperarc. The weight of a vertex node is

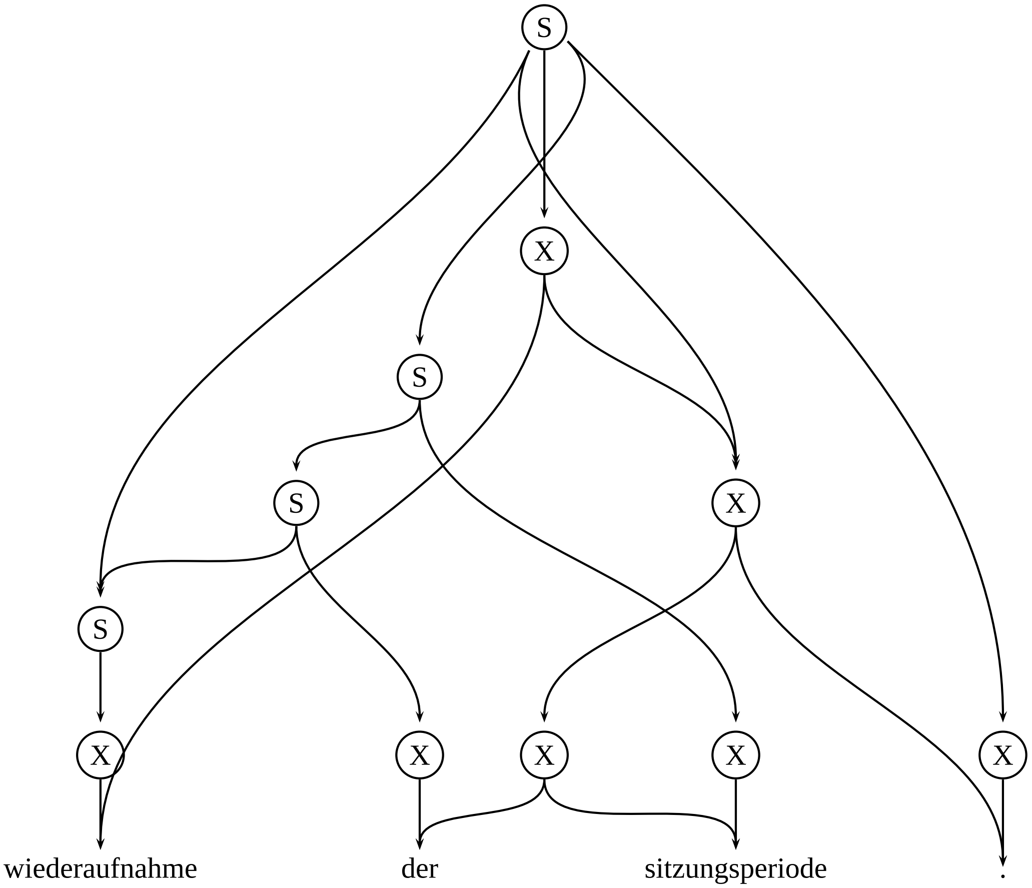

Figure 2: Sample parse chart viewed as a hypergraph. Each node in the hypergraph corresponds to a chart cell entry in the parse chart. Each hyperarc corresponds to a single rule application in the parse chart. The head node of a hyperarc is the left-hand side of the rule associated with the rule application. The tail nodes of the hyperarc are the elements of the source language right-hand for that rule. Note that a given node may have potentially many hyperarcs for which that node is the head.

the weight of the highest-weighted hyperarc headed at that vertex. In order to allow a translation to be extracted from a derivation, we store the target language right-hand side in the relevant hyperarc whenever we apply a rule in the parse chart. Figure 2 shows an example parse chart for a short sentence viewed as a hypergraph.

When considering a given span during parsing, there may be many rules with the same source language right-hand side, but different target language right-hand sides. In such cases, each rule will be applied separately, resulting in numerous hyperarcs with the same head and same tail nodes, but with different target language right-hand sides stored in each hyperarc. Figure 3 shows a simple partial hypergraph to illustrate this phenomenon.

Once feature values are calculated for each rule, we train feature weights for our system using minimum error rate training, using the open source MERT implementation presented in Olteanu et al. (2006). Minimum error rate training attempts to optimize the BLEU score of a development set of sentences by tuning the log-linear feature weights of the model.

We present the above implementation in the hope that such a freely available system may help stimulate further research. We now briefly introduce our ongoing research based on this system.

4 Translation using multiple source languages

Nearly all existing machine translation techniques assume a single source language and a single target language. However, governments and large busi-

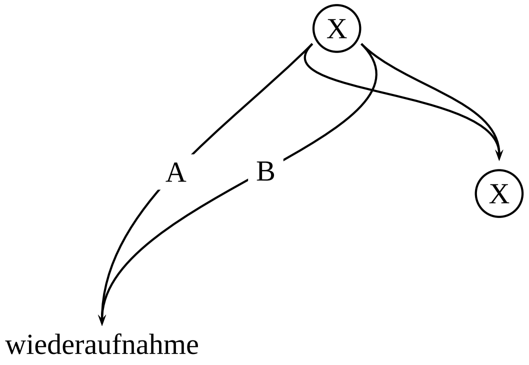

Figure 3: Partial hypergraph showing an portion of the parse chart from figure 2. This figure shows that there may be multiple hyperarcs with the same span, same head, and same tail nodes. Because each hyperarc corresponds to a single rule application in the parse chart, each hyperarc has a target language string associated with it that corresponds to the target language right-hand side of the rule application. In this example, hyperarc A represents an application of rule $X \rightarrow \langle$wiederaufnahme $\mathrm{X}_{①}$, resumption $\mathrm{X}_{①}\rangle$ and hyperarc B represents an application of rule $X \rightarrow \langle$wiederaufnahme $\mathrm{X}_{①}$, restarting $\mathrm{X}_{①}\rangle$. In real parse charts it is very common for an X node to have a large number of outgoing hyperarcs.

nesses often encounter situations where documents must be translated into a large number of languages. The proceedings of the European Union parliament is one notable example. In such situations, machine translation systems which take advantage of multilingual resources may be of use. Using relatively few resources, source documents can be manually translated from a single source into a small number of target languages, effectively resulting in multiple synchronous source languages for each document to be translated.

Our system exploits this multiplicity of source languages by training a translation model for each source-target language pair. When decoding a given sentence, we begin with the sentence, in original source language f_1. The original source sentence is manually or semi-automatically translated from f_1 into a small number of pseudo-source languages, $f_2 \ldots f_n$. Now, the modeling task becomes:

$$\arg\max_{e} P(e \mid f_1, f_2, \ldots, f_n) = \arg\max_{e,x} P(e \mid f_x) \quad (7)$$

We take the straightforward strategy of choosing the translation with the highest probability from any of the available sources. This approach is motivated by the observation that different language pairs present different ambiguities under different conditions. By starting with the same sentence in multiple source languages, it may be possible to find a better translation than if only one language pair is considered.

We translate from each source language f_1 and pseudo-source language $f_2 \ldots f_n$ into a common target language e, using our decoder and the appropriate source-target translation model. We would like our system to simply choose the translation with the highest score as the final result. But, because the log-linear scores produced by the decoder are not directly comparable, we must first normalize the score of each translation by the inside probability of the relevant parse tree.

A translation or partial translation can be uniquely identified in a parse tree by a chart cell entry $node$ and a rule application arc rooted at $node$. The inside probability, β, of a subtree is defined as the probability of a (partial) translation:

$$\begin{aligned} P(arc) &= exp(w(arc)) && (8) \\ \beta(arc) &= P(arc) \prod_{\substack{node \in \\ tail(arc)}} \beta(node) && (9) \\ \beta(node) &= \sum_{\substack{arc \in \\ bs(node)}} \beta(arc) && (10) \\ \beta(node_{term}) &= 1 && (11) \end{aligned}$$

Given a rule application arc that spans the entire parse tree, and the corresponding chart cell entry $node$ at which arc is rooted, the probability of the corresponding translation can be defined in terms of the weight of arc and the inside probability of $node$:

$$P(node) = \frac{exp(w(arc))}{\beta(node)} \quad (12)$$

The normalized probabilities of each translation can now be compared. The translation with the highest probability is selected as the final result.

Our current research applies this technique to the highly multi-lingual Europarl corpus (Koehn, 2005). We expect this will yield consistently higher scores than the Chiang (2007) baseline, as the best translation can be chosen from among multiple decoders.

5 Conclusion

Statistical hierarchical phrase-based translation is an active and very promising area of research. This paper introduces the first freely available implementation of such a translation system which also implements the cube pruning language model integration technique.

Governments and large businesses often encounter situations where documents must be translated into a large number of languages. In such situations, the use of translation techniques which exploit multiple parallel source languages may be able to improve the quality of translation results.

References

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proc. ACL*, pages 263–270.

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

Liang Huang and David Chiang. 2005. Better k-best parsing. In *Proc. IWPT*.

Liang Huang and David Chiang. 2007. Forest rescoring: Faster decoding with integrated language models. In *Proc. ACL*.

Philipp Koehn, Franz Joseph Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT-NAACL 2003*, pages 127–133.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. ACL 2007 Demo and Poster Sessions*.

Philipp Koehn. 2004. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *Proc. AMTA*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit X*.

Franz Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proc. ACL*, pages 440–447.

Franz Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. ACL*, pages 160–167.

Marian Olteanu, Chris Davis, Ionut Volosen, and Dan Moldovan. 2006. Phramer - an open source statistical phrase-based translator. In *Proc. Workshop on Statistical Machine Translation*, pages 146–149.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proc. Workshop on Statistical Machine Translation*, pages 138–141.