Phrasal: A Toolkit for Statistical Machine Translation with Facilities for Extraction and Incorporation of Arbitrary Model Features

Daniel Cer, Michel Galley, Daniel Jurafsky and Christopher D. Manning
Stanford University
Stanford, CA 94305, USA

Abstract

We present a new Java-based open source toolkit for phrase-based machine translation. The key innovation provided by the toolkit is to use APIs for integrating new features (/knowledge sources) into the decoding model and for extracting feature statistics from aligned bitexts. The package was used to develop a number of useful features written to these APIs including features for hierarchical reordering, discriminatively trained linear distortion, and syntax based language models. Useful utilities distributed with the toolkit include: a conditional phrase extraction system that builds a phrase table just for a specific dataset; and an implementation of MERT that allows for pluggable evaluation metrics for both training and evaluation with built in support for a variety of metrics (e.g., TERp, BLEU, METEOR).

1 Motivation

Progress in machine translation (MT) depends critically on the development of new and better model features that allow translation systems to better identify and construct high quality machine translations. The popular Moses decoder (Koehn et al., 2007) was designed to allow new features to be defined using factored translation models. In such models, the individual phrases being translated can be factored into two or more abstract phrases (e.g., lemma, POS-tags) that can be translated individually and then combined in a separate generation stage to arrive at the final target translation. While greatly enriching the space of models that can be used for phrase-based machine translation, Moses only allows features that can be defined at the level of individual words and phrases.

The Phrasal toolkit provides easy-to-use APIs for the development of arbitrary new model features. It includes an API for extracting feature statistics from aligned bitexts and for incorporating the new features into the decoding model. The system has already been used to develop a number of innovative new features (Chang et al., 2009; Galley and Manning, 2008; Galley and Manning, 2009; Green et al., 2010) and to build translation systems that have placed well at recent competitive evaluations, achieving second place for Arabic to English translation on the NIST 2009 constrained data track.[1]

We implemented the toolkit in Java because it offers a good balance between performance and developer productivity. Compared to C++, developers using Java are 30 to 200% faster, produce fewer defects, and correct defects up to 6 times faster (Phipps, 1999). While Java programs were historically much slower than similar programs written in C or C++, modern Java virtual machines (JVMs) result in Java programs being nearly as fast as C++ programs (Bruckschlegel, 2005). Java also allows for trivial code portability across different platforms.

In the remainder of the paper, we will highlight various useful capabilities, components and modeling features included in the toolkit.

2 Toolkit

The toolkit provides end-to-end support for the creation and evaluation of machine translation models. Given sentence-aligned parallel text, a new translation system can be built using a single command:

```
java edu.stanford.nlp.mt.CreateModel \
 (source.txt) (target.txt) \
 (dev.source.txt) (dev.ref) (model_name)
```

Running this command will first create word level alignments for the sentences in source.txt and target.txt using the Berkeley cross-EM aligner

[1] http://www.itl.nist.gov/iad/mig/tests /mt/2009/ResultsRelease/currentArabic.html

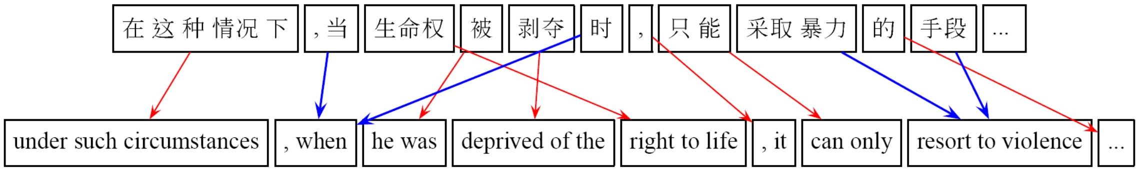

Figure 1: Chinese-to-English translation using discontinuous phrases.

(Liang et al., 2006).[2] From the word-to-word alignments, the system extracts a phrase table (Koehn et al., 2003) and hierarchical reordering model (Galley and Manning, 2008). Two n-gram language models are trained on the target.txt sentences: one over lowercased target sentences that will be used by the Phrasal decoder and one over the original source sentences that will be used for truecasing the MT output. Finally, the system trains the feature weights for the decoding model using minimum error rate training (Och, 2003) to maximize the system's BLEU score (Papineni et al., 2002) on the development data given by dev.source.txt and dev.ref. The toolkit is distributed under the GNU general public license (GPL) and can be downloaded from `http://nlp.stanford.edu/software/phrasal`.

3 Decoder

Decoding Engines The package includes two decoding engines, one that implements the left-to-right beam search algorithm that was first introduced with the Pharaoh machine translation system (Koehn, 2004), and another that provides a recently developed decoding algorithm for translating with discontinuous phrases (Galley and Manning, 2010). Both engines use features written to a common but extensible feature API, which allows features to be written once and then loaded into either engine.

Discontinuous phrases provide a mechanism for systematically translating grammatical constructions. As seen in Fig. 1, using discontinuous phrases allows us to successfully capture that the Chinese construction 当 **X** 的 can be translated as *when* **X**.

Multithreading The decoder has robust support for multithreading, allowing it to take full advantage of modern hardware that provides multiple CPU cores. As shown in Fig. 2, decoding speed scales well when the number of threads being used is increased from one to four. However, increasing the threads past four results in only marginal additional gains as the cost of managing the resources shared between the threads is starting to overwhelm the value provided by each additional thread. Moses also does not run faster with more than 4-5 threads.[3]

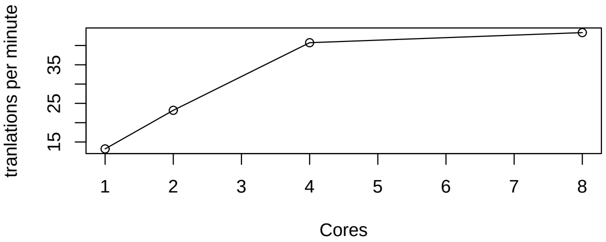

Figure 2: Multicore translations per minute on a system with two Intel Xeon L5530 processors running at 2.40GHz.

Feature API The feature API was designed to abstract away complex implementation details of the underlying decoding engine and provide a simple consistent framework for creating new decoding model features. During decoding, as each phrase that is translated, the system constructs a *Featurizable* object. As seen in Table 1, *Featurizable* objects specify what phrase was just translated and an overall summary of the translation being built. Code that implements a feature inspects the *Featurizable* and returns one or more named feature values. Prior to translating a new sentence, the sentence is passed to the active features for a decoding model, so that they can perform any necessary preliminary analysis.

Comparison with Moses Credible research into new features requires baseline system performance that is on par with existing state-of-the-art systems. Seen in Table 2, Phrasal meets the performance of Moses when using the exact same decoding model feature set as Moses and outperforms Moses significantly when using its own default feature set.[4]

[2] Optionally, GIZA++ (Och and Ney, 2003) can also be used to create the word-to-word alignments.

[3] `http://statmt.org/moses /?n=Moses.AdvancedFeatures (April 6, 2010)`

[4] Phrasal was originally written to replicate Moses as it was implemented in 2007 (release 2007-05-29), and the current ver-

Featurizable
Last Translated Phrase Pair
Source and Target Alignments
Partial Translation
Source Sentence
Current Source Coverage
Pointer to Prior Featurizable

Table 1: Information passed to features in the form of a *Featurizable* object for each translated phrase.

System	Features	MT06 (tune)	MT03	MT05
Moses	Moses	34.23	33.72	32.51
Phrasal	Moses	34.25	33.72	32.49
Phrasal	Default	35.02	34.98	33.21

Table 2: Comparison of two configurations of Phrasal to Moses on Chinese-to-English. One Phrasal configuration uses the standard Moses feature set for single factor phrase-based translation with distance and phrase level msd-bidirectional-fe reordering features. The other uses the default configuration of Phrasal, which replaces the phrase level msd-bidirectional-fe feature with a hierarchical reordering feature.

4 Features

The toolkit includes the basic eight phrase-based translation features available in Moses as well as Moses' implementation of lexical reordering features. In addition to the common Moses features, we also include innovative new features that improve translation quality. One of these features is a hierarchical generalization of the Moses lexical reordering model. Instead of just looking at the reordering relationship between individual phrases, the new feature examines the reordering of blocks of adjacent phrases (Galley and Manning, 2008) and improves translation quality when the material being reordered cannot be captured by a single phrase. This hierarchical lexicalized reordering model is used by default in Phrasal and is responsible for the gains shown in Table 2 using the default features.

To illustrate how Phrasal can effectively be used to design rich feature sets, we present an overview of various extensions that have been built upon the Phrasal feature API. These extensions are currently not included in the release:

Target Side Dependency Language Model The n-gram language models that are traditionally used to capture the syntax of the target language do a poor job of modeling long distance syntactic relationships. For example, if there are a number of intervening words between a verb and its subject, n-gram language models will often not be of much help in selecting the verb form that agrees with the subject. The target side dependency language model feature captures these long distance relationships by providing a dependency score for the target translations produced by the decoder. This is done using an efficient quadratic time algorithm that operates within the main decoding loop rather than in a separate reranking stage (Galley and Manning, 2009).

Discriminative Distortion The standard distortion cost model used in phrase-based MT systems such as Moses has two problems. First, it does not estimate the future cost of known required moves, thus increasing search errors. Second, the model penalizes distortion linearly, even when appropriate reorderings are performed. To address these problems, we used the Phrasal feature API to design a new discriminative distortion model that predicts word movement during translation and that estimates future cost. These extensions allow us to triple the distortion limit and provide a statistically significant improvement over the baseline (Green et al., 2010).

Discriminative Reordering with Chinese Grammatical Relations During translation, a source sentence can be more accurately reordered if the system knows something about the syntactic relationship between the words in the phrases being reordered. The discriminative reordering with Chinese grammatical relations feature examines the path between words in a source-side dependency tree and uses it to evaluate the appropriateness of candidate phrase reorderings (Chang et al., 2009).

5 Other components

Training Decoding Models The package includes a comprehensive tool set for training decoding models. It supports MERT training using coordinate descent, Powell's method, line search along random search directions, and downhill Simplex. In addition to the BLEU metric, models can be trained to optimize other popular evaluation metrics such

sion still almost exactly replicates this implementation when using only the baseline Moses features. To ensure this configuration of the decoder is still competitive, we compared it against the current Moses implementation (release 2009-04-13) and found that the performance of the two systems is still close. The current Moses implementation obtains slightly lower BLEU scores, respectively 33.98 and 32.39 on MT06 and MT05.

as METEOR (Lavie and Denkowski, 2009), TERp (Snover et al., 2009), mWER (Nießen et al., 2000), and PER (Tillmann et al., 1997). It is also possible to plug in other new user-created evaluation metrics.

Conditional Phrase Table Extraction Rather than first building a massive phrase table from a parallel corpus and then filtering it down to just what is needed for a specific data set, our toolkit supports the extraction of just those phrases that might be used on a given evaluation set. In doing so, it dramatically reduces the time required to build the phrase table and related data structures such as reordering models.

Feature Extraction API In order to assist in the development of new features, the toolkit provides an API for extracting feature statistics from a word-aligned parallel corpus. This API ties into the conditional phrase table extraction utility, and thus allows for the extraction of just those feature statistics that are relevant to a given data set.

6 Conclusion

Phrasal is an open source state-of-the-art Java-based machine translation system that was designed specifically for research into new decoding model features. The system supports traditional phrase-based translation as well as translation using discontinuous phrases. It includes a number of new and innovative model features in addition to those typically found in phrase-based translation systems. It is also packaged with other useful components such as tools for extracting feature statistics, building phrase tables for specific data sets, and MERT training routines that support a number of optimization techniques and evaluation metrics.

Acknowledgements

The Phrasal decoder has benefited from the helpful comments and code contributions of Pi-Chuan Chang, Spence Green, Karthik Raghunathan, Ankush Singla, and Huihsin Tseng. The software presented in this paper is based on work work was funded by the Defense Advanced Research Projects Agency through IBM. The content does not necessarily reflect the views of the U.S. Government, and no official endorsement should be inferred.

References

Thomas Bruckschlegel. 2005. Microbenchmarking C++, C#, and Java. *C/C++ Users Journal*.

P. Chang, H. Tseng, D. Jurafsky, and C.D. Manning. 2009. Discriminative reordering with Chinese grammatical relations features. In *SSST Workshop at NAACL*.

Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *EMNLP*.

Michel Galley and Christopher D. Manning. 2009. Quadratic-time dependency parsing for machine translation. In *ACL*.

Michel Galley and Christopher Manning. 2010. Improving phrase-based machine translation with discontiguous phrases. In *NAACL*.

Spence Green, Michel Galley, and Christopher D. Manning. 2010. Improved models of distortion cost for statistical machine translation. In *In NAACL*.

Philipp Koehn, Franz Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *NAACL*.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL*.

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

Alon Lavie and Michael J. Denkowski. 2009. The METEOR metric for automatic evaluation of machine translation. *Machine Translation*, 23.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *NAACL*.

Sonja Nießen, Franz Josef Och, and Hermann Ney. 2000. An evaluation tool for machine translation: Fast evaluation for MT research. In *LREC*.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *ACL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*.

Geoffrey Phipps. 1999. Comparing observed bug and productivity rates for java and C++. *Softw. Pract. Exper.*, 29(4):345–358.

M. Snover, N. Madnani, B.J. Dorr, and R. Schwartz. 2009. Fluency, adequacy, or HTER?: exploring different human judgments with a tunable MT metric. In *SMT workshop at EACL*.

C. Tillmann, S. Vogel, H. Ney, A. Zubiaga, and H. Sawaf. 1997. Accelerated DP based search for statistical translation. In *In Eurospeech*.