Demonstration of Joshua: An Open Source Toolkit for Parsing-based Machine Translation[*]

Zhifei Li, Chris Callison-Burch, Chris Dyer[†], Juri Ganitkevitch[+], Sanjeev Khudanpur, Lane Schwartz[⋆], Wren N. G. Thornton, Jonathan Weese, and **Omar F. Zaidan**
Center for Language and Speech Processing, Johns Hopkins University
† Computational Linguistics and Information Processing Lab, University of Maryland
+ Human Language Technology and Pattern Recognition Group, RWTH Aachen University
⋆ Natural Language Processing Lab, University of Minnesota

Abstract

We describe **Joshua** (Li et al., 2009a)[1], an open source toolkit for statistical machine translation. Joshua implements all of the algorithms required for translation via synchronous context free grammars (SCFGs): chart-parsing, n-gram language model integration, beam- and cube-pruning, and k-best extraction. The toolkit also implements suffix-array grammar extraction and minimum error rate training. It uses parallel and distributed computing techniques for scalability. We also provide a demonstration outline for illustrating the toolkit's features to potential users, whether they be newcomers to the field or power users interested in extending the toolkit.

1 Introduction

Large scale parsing-based statistical machine translation (e.g., Chiang (2007), Quirk et al. (2005), Galley et al. (2006), and Liu et al. (2006)) has made remarkable progress in the last few years. However, most of the systems mentioned above employ tailor-made, dedicated software that is not open source. This results in a high barrier to entry for other researchers, and makes experiments difficult to duplicate and compare. In this paper, we describe **Joshua**, a Java-based general-purpose open source toolkit for parsing-based machine translation, serving the same role as Moses (Koehn et al., 2007) does for regular phrase-based machine translation.

[*]This research was supported in part by the Defense Advanced Research Projects Agency's GALE program under Contract No. HR0011-06-2-0001 and the National Science Foundation under grants No. 0713448 and 0840112. The views and findings are the authors' alone.

[1]Please cite Li et al. (2009a) if you use Joshua in your research, and **not** this demonstration description paper.

2 Joshua Toolkit

When designing our toolkit, we applied general principles of software engineering to achieve three major goals: *Extensibility*, *end-to-end coherence*, and *scalability*.

Extensibility: Joshua's codebase consists of a separate Java `package` for each major aspect of functionality. This way, researchers can focus on a single `package` of their choosing. Fuurthermore, extensible components are defined by Java `interface`s to minimize unintended interactions and unseen dependencies, a common hindrance to extensibility in large projects. Where there is a clear point of departure for research, a basic implementation of each `interface` is provided as an `abstract class` to minimize work necessary for extensions.

End-to-end Cohesion: An MT pipeline consists of many diverse components, often designed by separate groups that have different file formats and interaction requirements. This leads to a large number of scripts for format conversion and to facilitate interaction between the components, resulting in untenable and non-portable projects, and hindering repeatability of experiments. Joshua, on the other hand, integrates the critical components of an MT pipeline seamlessly. Still, each component can be used as a stand-alone tool that does not rely on the rest of the toolkit.

Scalability: Joshua, especially the decoder, is scalable to large models and data sets. For example, the parsing and pruning algorithms are implemented with dynamic programming strategies and efficient data structures. We also utilize suffix-array grammar extraction, parallel/distributed decoding, and bloom filter language models.

Joshua offers state-of-the-art quality, having been ranked 4th out of 16 systems in the French-English task of the 2009 WMT evaluation, both in automatic (Table 1) and human evaluation.

System	BLEU-4
google	31.14
lium	26.89
dcu	26.86
joshua	**26.52**
uka	25.96
limsi	25.51
uedin	25.44
rwth	24.89
cmu-statxfer	23.65

Table 1: BLEU scores for top primary systems on the WMT-09 French-English Task from Callison-Burch et al. (2009), who also provide human evaluation results.

2.1 Joshua Toolkit Features

Here is a short description of Joshua's main features, described in more detail in Li et al. (2009a):

- **Training Corpus Sub-sampling:** We support inducing a grammar from a subset of the training data, that consists of sentences needed to translate a particular test set. To accomplish this, we make use of the method proposed by Kishore Papineni (personal communication), outlined in further detail in (Li et al., 2009a). The method achieves a 90% reduction in training corpus size while maintaining state-of-the-art performance.

- **Suffix-array Grammar Extraction:** Grammars extracted from large training corpora are often far too large to fit into available memory. Instead, we follow Callison-Burch et al. (2005) and Lopez (2007), and use a source language suffix array to extract only rules that will actually be used in translating a particular test set. Direct access to the suffix array is incorporated into the decoder, allowing rule extraction to be performed for each input sentence individually, but it can also be executed as a standalone pre-processing step.

- **Grammar formalism:** Our decoder assumes a probabilistic synchronous context-free grammar (SCFG). It handles SCFGs of the kind extracted by Hiero (Chiang, 2007), but is easily extensible to more general SCFGs (as in Galley et al. (2006)) and closely related formalisms like synchronous tree substitution grammars (Eisner, 2003).

- **Pruning:** We incorporate beam- and cube-pruning (Chiang, 2007) to make decoding feasible for large SCFGs.

- **k-best extraction:** Given a source sentence, the chart-parsing algorithm produces a *hypergraph* representing an exponential number of derivation hypotheses. We implement the extraction algorithm of Huang and Chiang (2005) to extract the k most likely derivations from the hypergraph.

- **Oracle Extraction:** Even within the large set of translations represented by a hypergraph, some desired translations (e.g. the references) may not be contained due to pruning or inherent modeling deficiency. We implement an efficient dynamic programming algorithm (Li and Khudanpur, 2009) for finding the *oracle translations*, which are most similar to the desired translations, as measured by a metric such as BLEU.

- **Parallel and distributed decoding:** We support *parallel decoding* and a *distributed language model* that exploit multi-core and multi-processor architectures and distributed computing (Li and Khudanpur, 2008).

- **Language Models:** We implement three local n-gram language models: a straightforward implementation of the n-gram scoring function in Java, capable of reading standard ARPA backoff n-gram models; a native code bridge that allows the decoder to use the SRILM toolkit to read and score n-grams[2]; and finally a Bloom Filter implementation following Talbot and Osborne (2007).

- **Minimum Error Rate Training:** Joshua's MERT module optimizes parameter weights so as to maximize performance on a development set as measured by an automatic evaluation metric, such as BLEU. The optimization consists of a series of line-optimizations using the efficient method of Och (2003). More details on the MERT method and the implementation can be found in Zaidan (2009).[3]

[2]The first implementation allows users to easily try the Joshua toolkit without installing SRILM. However, users should note that the basic Java LM implementation is not as scalable as the SRILM native bridge code.

[3]The module is also available as a standalone application, *Z-MERT*, that can be used with other MT systems.

- **Variational Decoding:** *spurious ambiguity* causes the probability of an output string among to be split among many derivations. The goodness of a string is measured by the total probability of its derivations, which means that finding the best output string is computationally intractable. The standard Viterbi approximation is based on the most probable derivation, but we also implement a variational approximation, which considers all the derivations but still allows tractable decoding (Li et al., 2009b).

3 Demonstration Outline

The purpose of the demonstration is 4-fold: 1) to give newcomers to the field of statistical machine translation an idea of the state-of-the-art; 2) to show actual, live, end-to-end operation of the system, highlighting its main components, targeting potential users; 3) to illustrate, through visual aids, the underlying algorithms, for those interested in the technical details; and 4) to explain how those components can be extended, for potential *power users* who want to be familiar with the code itself.

The first component of the demonstration will be an interactive user interface, where arbitrary user input in a source language is entered into a web form and then translated into a target language by the system. This component specifically targets newcomers to SMT, and demonstrates the current state of the art in the field. We will have trained multiple systems (for multiple language pairs), hosted on a remote server, which will be queried with the sample source sentences.

Potential users of the system would be interested in seeing an actual operation of the system, in a similar fashion to what they would observe on their own machines when using the toolkit. For this purpose, we will demonstrate three main modules of the toolkit: the rule extraction module, the MERT module, and the decoding module. Each module will have a separate terminal window executing it, hence demonstrating both the module's expected output as well as its speed of operation.

In addition to demonstrating the functionality of each module, we will also provide accompanying visual aids that illustrate the underlying algorithms and the technical operational details. We will provide visualization of the search graph and the 1-best derivation, which would illustrate the functionality of the decoder, as well as alternative translations for phrases of the source sentence, and where they were learned in the parallel corpus, illustrating the functionality of the grammar rule extraction. For the MERT module, we will provide figures that illustrate Och's efficient line search method.

(Software and documentation at: `http://cs.jhu.edu/~ozaidan/zmert`.)

4 Demonstration Requirements

The different components of the demonstration will be spread across at most 3 machines (Figure 1): one for the live "instant translation" user interface, one for demonstrating the different components of the system and algorithmic visualizations, and one designated for technical discussion of the code. We will provide the machines ourselves and ensure the proper software is installed and configured. However, we are requesting that large LCD monitors be made available, if possible, since that would allow more space to demonstrate the different components with clarity than our laptop displays would provide. We will also require Internet connectivity for the live demonstration, in order to gain access to remote servers where trained models will be hosted.

References

Chris Callison-Burch, Colin Bannard, and Josh Schroeder. 2005. Scaling phrase-based statistical machine translation to larger corpora and longer phrases. In *Proceedings of ACL*.

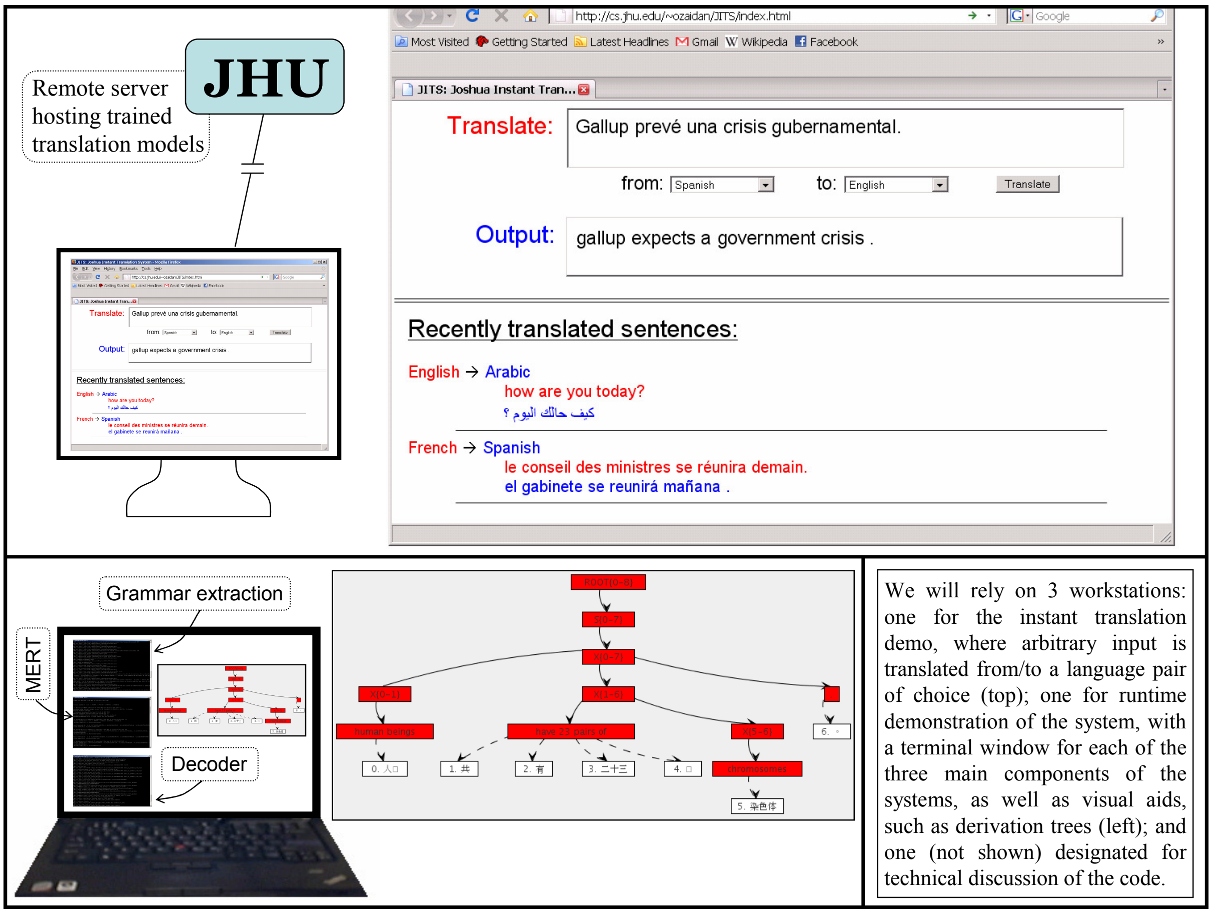
Chris Callison-Burch, Philipp Koehn, Christof Monz, and Josh Schroeder. 2009. Findings of the 2009 Workshop on Statistical Machine Translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 1–28, Athens, Greece, March. Association for Computational Linguistics.

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

Jason Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. In *Proceedings of ACL*.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the ACL/Coling*.

Liang Huang and David Chiang. 2005. Better k-best parsing. In *Proceedings of the International Workshop on Parsing Technologies*.


Remote server hosting trained translation models
JHU
http://cs.jhu.edu/~ozaidan/JITS/index.html
Most Visited Getting Started Latest Headlines Gmail Wikipedia Facebook
JITS: Joshua Instant Tran...
Grammar extraction
MERT
Decoder
ROOT{0-8}
S{0-7}
X{0-7}
X{0-1}
X{1-6}
.
human beings
have 23 pairs of
X{5-6}
6. 。
0. 人口
1. 共
2. 有
3. 二十三
4. 对
chromosomes
5. 染色体

Translate: Gallup prevé una crisis gubernamental.

from: Spanish to: English Translate

Output: gallup expects a government crisis .

Recently translated sentences:

English → Arabic
how are you today?
كيف حالك اليوم ؟

French → Spanish
le conseil des ministres se réunira demain.
el gabinete se reunirá mañana .

We will rely on 3 workstations: one for the instant translation demo, where arbitrary input is translated from/to a language pair of choice (top); one for runtime demonstration of the system, with a terminal window for each of the three main components of the systems, as well as visual aids, such as derivation trees (left); and one (not shown) designated for technical discussion of the code.

Figure 1: Proposed setup of our demonstration. When this paper is viewed as a PDF, the reader may zoom in further to see more details.

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 *Proceedings of the ACL-2007 Demo and Poster Sessions*.

Zhifei Li and Sanjeev Khudanpur. 2008. A scalable decoder for parsing-based machine translation with equivalent language model state maintenance. In *Proceedings Workshop on Syntax and Structure in Statistical Translation*.

Zhifei Li and Sanjeev Khudanpur. 2009. Efficient extraction of oracle-best translations from hypergraphs. In *Proceedings of NAACL*.

Zhifei Li, Chris Callison-Burch, Chris Dyer, Juri Ganitkevitch, Sanjeev Khudanpur, Lane Schwartz, Wren Thornton, Jonathan Weese, and Omar Zaidan. 2009a. Joshua: An open source toolkit for parsing-based machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 135–139, Athens, Greece, March. Association for Computational Linguistics.

Zhifei Li, Jason Eisner, and Sanjeev Khudanpur. 2009b. Variational decoding for statistical machine translation. In *Proceedings of ACL*.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment templates for statistical machine translation. In *Proceedings of the ACL/Coling*.

Adam Lopez. 2007. Hierarchical phrase-based translation with suffix arrays. In *Proceedings of EMNLP-CoLing*.

Franz Josef Och. 2003. Minimum error rate training for statistical machine translation. In *Proceedings of ACL*.

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal smt. In *Proceedings of ACL*.

David Talbot and Miles Osborne. 2007. Randomised language modelling for statistical machine translation. In *Proceedings of ACL*.

Omar F. Zaidan. 2009. Z-MERT: A fully configurable open source tool for minimum error rate training of machine translation systems. *The Prague Bulletin of Mathematical Linguistics*, 91:79–88.