Vanda
A Statistical Machine Translation Toolkit

Matthias Büchse* Toni Dietze Johannes Osterholzer
Anja Fischer* Linda Leuschner*

Technische Universität Dresden

March 22, 2012

The aim of Machine Translation is to teach computers to translate sentences (and ultimately, texts) from one language into another. The subarea of Statistical Machine Translation (SMT) applies methods from Statistics and Machine Learning to automatically select a translation function that performs well on a sample of existing translations.

More precisely, an engineer devises a class of translation functions (called hypothesis space) and a loss function, and then she executes an algorithm that selects an element of the hypothesis space that has the least loss on the sample. Nowadays, the hypothesis space is often specified via a class of weighted synchronous grammars; i.e., each translation function corresponds to a grammar, and it maps every input sentence to a translation that has a derivation with highest weight.

Current SMT systems such as Hiero [1, 2], Moses [6], Joshua [7], or cdec [5] implement their respective specification directly in thousands of lines of code in languages such as Python, Java or C++. Sometimes the code adds substantial refinement to the specification, in form of poorly documented short-cuts and heuristics, making it hard to grasp the actual workings of the systems.

Moreover, the tasks of (i) preparing the data, (ii) selecting the translation function, and (iii) performing a translation require complex workflows, consisting of several programs to be run, each with its own command-line syntax. Because of the rapid development in the area of SMT and the prototypic nature of academic research, documentation of the workflows is often sub-optimal. In the absence of a workflow management system such as LoonyBin [3, 4], carrying out these workflows is error-prone, regardless of whether it is done manually or by means of a script.

We introduce our new SMT toolkit, Vanda, and the accompanying workflow management system, Vanda Studio. We hope to accomplish two objectives:

1. facilitate specifying core SMT algorithms in an algebraic manner, with a straightforward implementation; to this end, our solution is twofold:

*financially supported by DFG VO 1011/6-1.

    - we make full use of the theory of weighted automata and grammars as well as accompanying constructions and algorithms; for example, product constructions (weighted intersection), reduction constructions (removing useless rules), algorithms finding best derivations;
    - we use the functional programming language Haskell, which directly supports an algebraic way of thinking.

2. facilitate the cycle of "specify – take measurements – evaluate"; to this end, we adopt the concept of hyper-workflows from LoonyBin, i.e.,
    - a workflow is a formal object that can be edited graphically via Vanda Studio;
    - a workflow element (e.g., a program) is a formal object as well, and the user interface provides a specification of every workflow element;
    - a workflow can be transformed into an executable artifact, e.g., a shell script, that incorporates sanity checks to ensure correct interplay between workflow elements;
    - a workflow can be transformed into different executable artifacts depending on the intended platform, e.g., a local workstation vs. a grid of computers;
    - hyper-workflows introduce nondeterminism into workflows, allowing the engineer to specify alternatives; measurements are taken for each of these alternatives, and tables and diagrams for comparison can be generated automatically.

We note that Vanda Studio is tailored to the Vanda toolkit, but it is not limited to it.

In our presentation, we will show that Vanda and Vanda Studio attain our two above-mentioned objectives by means of simple SMT tasks such as parsing and translation.

References

[1] David Chiang. Hierarchical phrase-based translation. *Comp. Ling.*, 33(2):201–228, 2007.

[2] David Chiang, Adam Lopez, Nitin Madnani, Christof Monz, Philip Resnik, and Michael Subotin. The hiero machine translation system: extensions, evaluation, and analysis. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 779–786, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[3] Jonathan H. Clark and Alon Lavie. Loonybin: Keeping language technologists sane through automated management of experimental (hyper)workflows. In *Proceedings LREC 2010*, 2010.

[4] Jonathan H. Clark, Jonathan Weese, Byung Gyu Ahn, Andreas Zollmann, Qin Gao, Kenneth Heafield, and Alon Lavie. The machine translation toolpack for loonybin: Automated management of experimental machine translation hyperworkflows. *The Prague Bulletin of Mathematical Linguistics*, 2010.

[5] Chris Dyer, Adam Lopez, Juri Ganitkevitch, Jonathan Weese, Ferhan Ture, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of the ACL 2010 System Demonstrations*, pages 7–12, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[6] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[7] Zhifei Li, Chris Callison-Burch, Chris Dyer, Juri Ganitkevitch, Sanjeev Khudanpur, Lane Schwartz, Wren N. G. Thornton, Jonathan Weese, and Omar F. Zaidan. Joshua: an open source toolkit for parsing-based machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, StatMT '09, pages 135–139, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.