The Moses for Localization Open Source Project

Achim Ruopp
Digital Silk Road
Washington, DC 20008
achim@digitalsilkroad.net

Abstract

The open source statistical machine translation toolkit Moses has recently drawn a lot of attention in the localization industry. Companies see the chance to use Moses to leverage their existing translation assets and integrate MT into their localization processes. Due to the academic origins of Moses there are some obstacles to overcome when using it in an industry setting. In this paper we discuss what these obstacles are and how they are addressed by the newly established *Moses for Localization* open source project. We describe the different components of the project and the benefits a company can gain from using this open source project.

1 Motivation

Machine translation has arrived for good in the localization industry; the influential Global Watchtower blog even called it a tidal wave earlier this year.

While research in machine translation has been going on for over 50 years, recently adopted statistical methods make it easier to create customized MT systems for specific language pairs and topics. By leveraging existing translation assets, translation providers and buyers can build well-performing MT systems that, combined with post-editing, are more productive than traditional translation methods alone. This is beginning to lead to wide-spread adoption of statistical MT (SMT) systems throughout the language industry.

Just over a decade ago, the localization industry was transformed by a similar technological shift – the adoption of translation memory, translation management and globalization management systems (TM/TMS/GMS).

The adoption of TM/TMS/GMS systems came at a significant cost, not only in terms of technology investments, but also in the form of changes to established business processes and business models. Today TM/TMS/GMS systems have been widely adopted and their adoption has yielded significant economic benefits.

With the arrival of SMT it is in the best interest of translation providers and translation buyers to integrate the SMT systems into the established TM/TMS/GMS systems with the least disruption possible, while realizing the promised productivity gains.

Commercial providers are already starting to offer integrated solutions. However, given the wide variety of TM/TMS/GMS systems and highly customized workflows, some users require the independence and flexibility to build custom MT systems, integrate MT systems tightly into their processes and generally adapt MT systems to their needs.

Thanks to the efforts of the academic community the open source SMT system Moses (Koehn, et al., 2007) provides an excellent basis for such customized integration projects.

In the upcoming sections we describe typical integration scenarios of translation workflows with Moses. It will become clear that there are some crucial gaps that have to be filled to integrate Moses in localization workflows.

We started the *Moses for Localization* open source project to address these gaps. We outline the *Moses for Localization* plan to address the most

urgent gaps, how you can use the code in your projects and how you can participate in the effort.

2 Translation Workflow MT Integration

We first look at the integration of the Moses MT system into a typical translation process. This assumes an already trained and tuned Moses MT system, later in section 2.3 we get to the process of leveraging existing translation memories to train such a system.

2.1 Offline Integration

Offline integration is the most straightforward integration between a translation management system (TMS) and the Moses MT system with the least number of dependencies. A schema of this integration is shown in Figure 1.

Offline in this context means that source segments, which cannot be matched in the translation memory (TM), are sent for translation in bulk to the MT system. The resulting machine translations are then transferred back to the TMS for editing by a human translator.

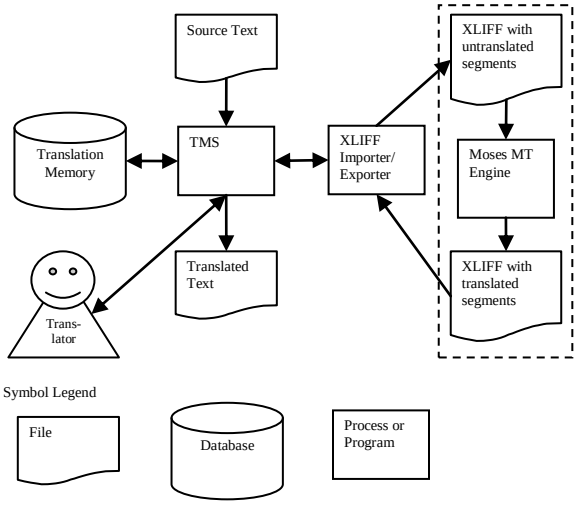

Figure 1: TMS-Moses offline integration with XLIFF

This integration process uses the localization industry standard XLIFF (XML Localization Interchange File Format[1]) to exchange translations between the TMS and the MT system. Many currently available TMS systems provide XLIFF import and export functions.

XLIFF is used in this process because it is able to store translatable text along with meta information about the original source format and inline formatting (e.g. bold/italic phrases). The process is transferable to other standard and non-standard localization formats that meet the same criteria.

Because XLIFF contains meta information like comments and inline formatting, and Moses is mainly designed for the translation of plain text, the sub process in the dashed box in Figure 1 is a bit more complex. The meta information has to be carried through the MT process out-of-band as shown in Figure 2.

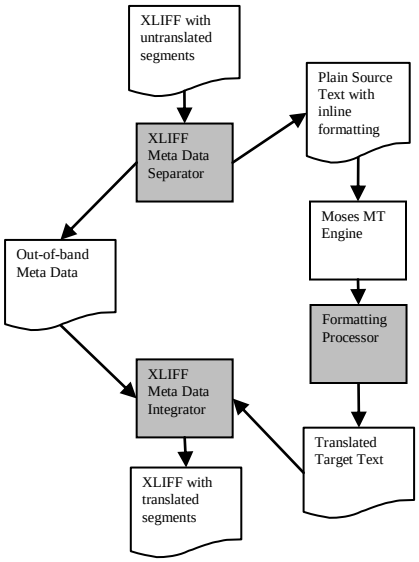

Figure 2: Processing XLIFF Meta Data

The offline integration requires XLIFF meta data separation/integration components and a formatting processor to insert inline formatting information into the translations (shown in gray in Figure 2). It is one of the main goals of the *Moses for Localization* project to provide these missing components.

2.2 Online Integration

With the advent of free online translation services like Microsoft Translator[2] and Google Translate[3] many translation environments now provide translators with the option to pre-translate text with these services. This integration is implemented via free web service APIs that most online translation services provide.

[1] http://www.oasis-open.org/committees/xliff

[2] http://www.microsofttranslator.com/

[3] http://translate.google.com/

In some translation environments additional MT services can be configured, which can be used to integrate customized Moses MT systems as shown in Figure 3.

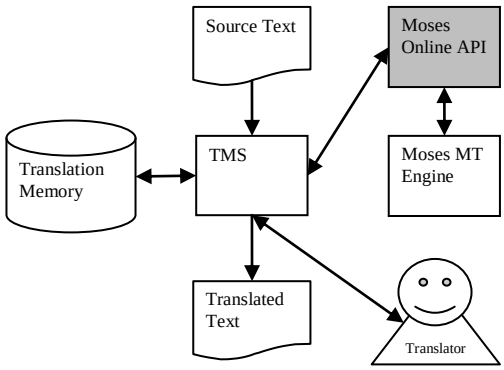

Figure 3: TMS-Moses online integration with API

The Moses tool chain already contains such a web service API; however, this component requires the SRILM language model package (Stolcke, 2002), which is not open source. In the *Moses for Localization* project we aim to provide a web service API to Moses that supports IRSTLM (Marcello, Bertoldi, & Cettolo, 2008) and other open source language model packages.

On-demand translation is another context in which an online web service API is beneficial. In an on-demand translation scenario translation has to be instant and MT output quality is often sufficient without editing, e.g. in online chat. In this case a standard web service API to a customized Moses MT system can provide a more appropriate chat experience than standard translations from a large online translation provider. For instance a food related chat room can benefit from an MT system that is tuned to terminology on cooking and recipe ingredients.

2.3 MT Training with Translation Memories

Over the years many translation providers and translation buyers have built up considerable assets in the form of translation memories. Lately initiatives like the TAUS Data Association[4] encourage sharing of these assets among industry participants.

Translation memories are very valuable for the training of Moses MT systems customized to certain topic areas. Translation memory systems provide the option to export TMs in the industry standard TMX format[5].

To use TMX data for training and tuning Moses MT systems, they need to be converted to plain text parallel corpora.

Moses for Localization provides scripts for conversion from TMX to parallel corpora and vice versa. Scripts to create training, tuning and evaluation sets from parallel corpora are also available.

3 Moses for Localization

Companies that are already using Moses have all duplicated investments to overcome the same issues described in the previous sections. What we are proposing with the *Moses for Localization* project is to combine investments to efficiently leverage the potential that Moses promises for localization.

In the near term this means to work on the following goals:

1. Enable the use of standard translation memory TMX data as training, tuning and evaluation data (code already available)
2. Enable the translation of XLIFF files (and other file formats containing meta information and inline formatting)
3. Automate the batch translation of XLIFF files (potentially with the Moses experiment.perl framework)
4. Provide a REST translation server (Fielding, 2000) for online integration with various translation environments

4 Benefits of Open Source

A collaborative effort to fill the gaps with open source code has several significant advantages for the industry. In addition to the desired flexibility for integration and interoperability, open source solutions provide vendor independence, lower costs than in-house development, and a community for continued support and development.

Outside the language industry the model of shared open source development enabling the use of software that originated in the academic world in a commercial setting has been successfully applied in several projects. A prime example is the PostgreSQL database.

[4] http://www.tausdata.org/

[5] http://www.lisa.org/tmx/

Because the *Moses for Localization* project uses the Apache License 2.0[6], companies can integrate the code worry-free into their projects, whether they are open-source or proprietary.

For engineers in the localization industry *Moses for Localization* makes complex Moses systems more approachable and manageable; the code serves as a training aid for the subject matter unfamiliar to most localization engineers.

Even if an MT user decides to purchase a proprietary MT solution, open source can offer a second source alternative.

5 Participation

There are opportunities for everyone to contribute, individuals and large organizations, technical, linguistic and business oriented participants. Options for participation range from reporting bugs and providing language expertise, over writing documentation and supporting the community, to contributing code and providing funding. Information on *Moses for Localization* and a discussion forum can be found on http://groups.google.com/group/m4loc.

6 Future Work

Of course the challenges of fully integrating MT into a human translation process to its full potential go beyond the mere integration into existing translation environments. Issues like training data selection, training data cleaning, evaluation of MT output and the post-editing of machine translations have to be considered. Language-specific issues like morphologically rich languages and word segmentation for East Asian languages need to be addressed. On the operational level things like installation and usability need improvement.

Building on continued Moses development of Moses in the academic community and combined with other open source projects, *Moses for Localization* can serve as a crucial building block in a true end-to-end open source machine translation solution for the localization industry.

[6] http://opensource.org/licenses/apache2.0.php

References

Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures.* Irvine: University of California.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Frederico, M., Bertoldi, N., et al. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. *Annual Meeting of the Association for Computational Linguistics (ACL).* Prague, Czech Republic.

Marcello, F., Bertoldi, N., & Cettolo, M. (2008). IRSTLM: an Open Source Toolkit for Handling Large Scale Language Models. *Interspeech 2008, ISCA*, (pp. 1618-1621). Brisbane, Australia.

Stolcke, A. (2002). SRILM—An extensible language modeling toolkit. *In Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002)*, (pp. 901-904).