MT use within the enterprise: Encouraging adoption via a unified MT API

Raymond Flournoy
Adobe Systems Inc.
345 Park Avenue
San Jose, CA 95110 USA
flournoy@adobe.com

Abstract

Adobe Systems has employed Machine Translation as part of the document localization process for over two years. In order to encourage the wider adoption of the technology within the company, we have recently created a unified API across our available MT technologies. This unified MT service enables simpler integration of MT within products and processes, allows sharing of license and server costs across the company, creates a platform for mixing technologies into a best-of-breed solution, and provides greater sharing of expertise and best practices.

1 Introduction

Adobe Systems has successfully employed Machine Translation (MT) with post-editing as part of the document localization process for over two years. While the use of MT for localization continues to grow, Adobe is also moving to expand the application of the technology to other products and processes throughout the company, both internally and externally facing.

In order to expand access to MT technology, the Globalization group has developed a unified API integrating all available technologies, including licensed, open-source, and free online engines. The API is being developed based on "personas" derived from use cases describing various product and project groups within the company.

In this paper we describe the motivations for the MT API, summarize its benefits, and conclude with some of the initial lessons and next steps for the unified API.

2 A Brief History of MT at Adobe

Beginning in 2009, Adobe began to experiment with incorporating MT into the localization process for product documentation. MT engines from two commercial vendors were licensed and customized for Adobe terminology. The first language pairs licensed were EN>FR and EN>ES from Language Weaver and EN>RU from PROMT. The MT output was post-edited by various translation vendors in order to bring the output to publishable quality, comparable to text translated by traditional means.

In order to measure the value of MT, the throughput for MT post-editing was measured and compared against baselines representing manual translation. Depending on the Adobe product line, the efficiency gains on the localization task – as defined as the reduction in time required to complete translation – ranged from roughly 20-40%, and were encouraging enough to support expanding the use of MT to more products and language pairs.

Since 2009, the number of language pairs integrated in the localization workflow has expanded to include EN>PT-BR (from PROMT), EN>ZH-CN (from CCID, via PROMT's framework), and EN>DE and EN>NL (from Systran). The approximate range of efficiency gains seen for each language pair is given in Table 1.

Language Pair	Vendor	Efficiency Gains
EN>FR	Language Weaver	25-50%
EN>ES	Language Weaver	25-40%
EN>RU	PROMT	15-35%
EN>PT-BR	PROMT	20-35%
EN>ZH-CN	CCID	7-20%
EN>DE	Systran	Still calculating
EN>NL	Systran	Still calculating

Table 1: MT language pairs used in document localization workflow

Since 2009, approximately 2.75 million new words of documentation for over two dozen product lines has been localized using MT with post-editing. Before the end of the year, we hope to add at least two more language pairs and expand the use of MT to the entire Adobe product line. Additionally, we have recently begun using MT for localization of user interface text, and have seen encouraging results for the first two test products.

3 Surveying possible MT use cases within the enterprise

Encouraged by the positive results from the application of MT to the localization task, at the end of 2010 we began to consider the best way to enable product teams and other groups within Adobe to leverage the MT technology as well. The goal was to encourage these other team to begin experimenting with and integrating the MT technology by creating an internal MT service.

In order to better understand the requirements for such an internal MT service, we conducted interviews with a wide range of groups within Adobe. The interview subjects included consumer products, enterprise products, internally-facing projects, and externally-facing services.

Through the interviews we endeavored to understand how the projects or products would benefit from MT, technically as well as linguistically. In addition to brainstorming ideas for MT integrations, we asked specific questions to flesh out the requirements for an internal MT service. Among the questions asked during the discussions were the following:

- What are the quality requirements for the MT output? Human-quality publishable text, or gist-level rough translations?
- In what format would the translation requests be sent? (E.g. plain text, XLIFF, PDF?)
- Would the translation requests contain confidential data, possibly requiring secure handling?
- Who is composing the text to translate? Someone within Adobe, or an external user?
- Will the text be related to a narrow subject domain, such as a specific product? Or is the possible topic unrestricted?

Based on the answers to these and other questions, we were able to classify the various use cases within the company into five broad categories. Following a practice common in the product marketing world, we assigned personas to each of the five categories to help summarize the use cases and user requests.

The full set of personas cannot be discussed publicly at this time, but two in particular are addressed in the initial design of the unified API.

- **The Localizer**:[1] This user has a quantity of text that needs to be translated into one or more other languages. The output must be human quality translation, and the text is produced by people within Adobe. The text often contains confidential materials, is usually related to a narrow domain, and can come in a variety of formats, including structured XLIFF. This user's goal for using MT is to see efficiency gains which translate into time or cost savings, and enable her to localize into more languages.
- **The Reader**: This user is seeing and processing text in one or more languages which he does not understand. The text is usually produced by end-users outside Adobe, and can be within a narrow subject domain or on wide ranging topics. The format of the text is also unlimited, and the contents are possibly confidential. This user's goal for MT is getting a gist-level understanding of

[1] This use case includes product document localization which was our first application of MT.

the text, and high processing speed is crucial.

4 The Unified API

4.1 Description of the MT services APIs

In order to facilitate the integration of MT technology by product and project groups within Adobe, the Globalization group created an MT service with a unified API which sits across all of the available MT technologies, providing a single interface for all of the engines.

The unified API integrates the MT engines licensed from commercial providers, engines built using the Moses open-source package, and free online systems. The API acts as a router, selecting the best engine for the requested language pair and customization target, and it also provides load balancing and an additional layer of error handling for the engines.

The first method included as part of the API was a simple text translation call. The input and output to this command are both text strings, and the call is made synchronously. This basic synchronous text translation API was the most requested service during the user interviews.

Next, a method was added for translating XLIFF-formatted files. This method is essential for any translation call including placeholders. Asynchronous versions of both text and XLIFF calls were added next.[2]

4.2 Engine Customization and Secure Communication

Two of the major advantages that the unified API provides over a call to a free online service are the availability of the Adobe-customized engines and the ability to make a secure call to our servers when translating confidential materials.

In order to expose both of these features, a translation call includes three optional parameters:

- Product or project name
- Content type
- Confidentiality flag

The product/project name and the content type are used to route the translation request to an engine customized for that product and content. For example, a user might request an engine trained on the Photoshop product, and the user interface (UI) content specifically. If a customized engine does not exist for that customization target, the requirements are gradually relaxed until the most appropriate engine is found.

In the case that no engine has been customized for the requested language pair, an uncustomized, baseline engine will be called – possibly one provided by a free online service such as Google or Bing.

However, the confidentiality flag parameter ensures that requests are never sent outside of the Adobe servers when the text must be handled securely. If a user requests a language pair which Adobe has not licensed or built internally and the confidentiality flag is set to “true,” then the request will not find a valid engine and will return an error.

4.3 Benefits of the Unified API

The API provides four broad benefits as we roll out the use of MT within the company.

Shared costs: By centralizing the engine licenses and the server maintenance, the various costs and overhead of MT can be shared across groups, and more advantageous terms can be negotiated with vendors to benefit the entire company. Many of the groups within Adobe who are potential users of MT are still at an exploratory stage with the technology, and the availability of a free service is essential to encourage experimentation. Some groups turn to free online services to begin experimentation with MT, but this creates problems when confidential texts are being processed. Additionally, free online services do not allow for engine customization. The unified API serves as a secure alternative, which groups can use without worrying about costs at the initial stages.

Simplified integration: One major goal of the unified MT API is to abstract technical details away from the end user. This is especially important when multiple MT engines are in use. Different MT engines have differing call syntax, handle customization and dictionaries differently, and

[2] In a *synchronous* call, the calling agent waits for a result from the service. When a result is received, or after a maximum wait time is reached, the call ends. In an *asynchronous* call, the calling agent issues the request, but does not wait for a response. A separate call is used to retrieve the results at a later time.

support a different subset of input formats. With the unified API, the user does not need to learn the specifics for the particular engines being called, and in fact often does not even know what underlying engine is being called. The downside of this simplified integration is that the full power of each engine is not surfaced – for example, if only one engine supports PowerPoint translation but the others do not support this input format, the unified API cannot support it. However, the benefits of simplified integration currently outweigh this disadvantage.

Best-of-breed: The technical abstraction that the unified API provides allows us to integrate engines from various vendors, selecting the best technology for each language pair or customization target. Additionally, by separating the user from the actual engines, we can swap out vendors if better alternatives are found, as well as perform upgrades and retrainings without interrupting the service.

Best practices and MT consulting: A final benefit of the unified API is that it provides a framework for sharing of information and best practices as related to MT. During our user interviews we found that many products and projects had very little familiarity with MT technology, and were uncertain about the possible points of integration. The Globalization group now serves the role of MT consultant, educating teams about the basics of the technology, and sharing with them example integrations from within Adobe as well as from other corporations.

4.4 Initial Users of the Unified Service

Two project groups have integrated the unified MT API and have begun using the service.

Customer Support: The Customer Support organization has integrated the MT API with the tool that customer support agents use to generate outgoing communication. All communication with customers is maintained both in English and in the language of the customer, so the goal of incorporating MT is to speed up the process of generating documents in two different languages. The customer support agents act as post-editors on the MT output to guarantee that all translations are corrected before being sent outside of the Adobe network.

The integration of MT with the customer support tool is in the testing stage, but a second point of integration is already being discussed. The customer support agents receive large amounts of communication in various languages, and currently, bilingual employees are used to triage and route the communication to the proper office or product group. In some cases, the customer support agents cannot even identify the language of the incoming communication. Machine Translation would allow agents to perform gisting translations of the incoming email to speed up the routing of these incoming messages.

Community Translation: Adobe has begun experimenting with community translation projects, enabling interested users to contribute to the localization of products into new languages. The MT API has been integrated with the community translation tools to provide an MT pre-translation to the community translators. Here, the users act as post-editors, polishing up the MT output, or substituting a new translation of their own.

Initial feedback from community translators has been positive. The next step is to close the feedback loop so that the edits provided by the community translators can be used to improve the MT engine training.

5 Lessons and Next Steps

As we continue to talk with groups internally about adopting MT technology and integrating the unified API, a number of interesting issues and lessons have surfaced.

- Because the MT engines differ in their supported features, it will become increasingly difficult to maintain uniform support across the board. If we decide to support certain features for some engines and not others, it will require users to know which engine they are using at the time of calling, which violates the abstraction that the API was designed for.
- Similarly, the engines have very different response times and failure rates. It may be technically impossible to guarantee the same response times and service levels for all language pairs because of these differences.

- There is a high cost to the best-of-breed approach. We feel that this cost was justified because of the large number of language markets that Adobe serves as well as the large number of customization targets that Adobe's product lines require. However, other enterprises might be better served by a cheaper, single-provider model.
- One advantage of our internal service is the access to Adobe customized engines, however many of the potential MT users have very little data with which we can train a statistical MT engine. Additionally, MT can be leveraged to enter a new language market more cheaply, however in that situation as well there will be little existing training data.
- Many product and project teams are excited about the prospect of using MT, but the familiarity with the technology is still very low. Thus, the amount of time we spend on education and gathering example use cases will need to grow.

6 Conclusion

Adobe has realized many benefits from integrating Machine Translation with its document localization process, and now is working to leverage the investment in MT technology to benefit product and project teams throughout the company.

The Globalization group at Adobe has assumed the role of MT consultants, helping to educate and guide groups as they explore this new technology. In order to facilitate this exploration, we have developed the unified API based on input from possible users of the MT service. Currently very simple, the API will grow in complexity as users become more familiar with MT technology and the integrations become more sophisticated.