Term Translation Central: Up-to-date MT without Frequent Retraining

Ventsislav Zhechev ventsislav.zhechev@autodesk.com
Autodesk Development Sàrl, Neuchâtel, Switzerland

Abstract

In this paper, we present the Term Translation Central — a tool that allows Autodesk to keep the in-house machine translation (MT) engines up-to-date and ready to tackle new content on a per-product basis. The system handles a range of tasks including the extraction of term candidates from new software user interface (UI) and documentation content, a user-friendly web interface where the extracted terms can be translated and reviewed and the provision of lists of available terms on demand per language/product combination for use during MT processing. The introduction of the Term Translation Central in production allows us to provide high-quality product-specific MT without the need to develop and maintain a large number of domain-specific MT engines. The Term Translation Central is developed using Python with the Flask micro-framework and NLTK.

1. Introduction

Autodesk is a company with a very broad range of software products that are distributed worldwide. The high-quality localisation of these products is a major part of our commitment to a great user experience for all our clients. The translation of software documentation and user interface (UI) strings plays a central role in our localisation process and we need to provide a fast turnaround of very large volumes of data. To accomplish this, we use an array of tools — from document- and localisation-management systems to machine translation (MT).

In this paper, we present the Term Translation Central. This is a platform that facilitates the production of the best possible MT output at all times through a combination of a backend that extracts potential terms from new content, a web-based user interface (UI) for use by translators and an API that allows direct access to translated terms. We start in Section 2 with a quick look at our MT infrastructure. Section 3 focuses on our previous approach to handling product-specific terminology during MT. In Section 4, we describe in detail the Term Translation Central and give some examples of its use. We discuss some future plans and challenges in Section 5 and conclude in Section 6.

2. MT Infrastructure at Autodesk

In this section, we present in short the MT infrastructure that we have built to support the localisation effort at Autodesk. An in-depth discussion is available in (Zhechev, 2014).

We actively employ MT in a variety of setups — from translator productivity to raw MT — and we are constantly improving our toolkit to widen our language coverage and achieve higher perceived quality. At the core of this toolkit are the tools developed and distributed with the open-source Moses project (Koehn et al., 2007), which are used as a productivity tool for all our post-edited UI and user manual content. Currently, we use Moses-based MT for translating from US English into fourteen languages: Czech, German, Spanish, French, Hungarian, Italian, Japanese, Korean, Polish, Brazilian and European Portuguese, Russian, Simplified and Traditional Chinese (hereafter, we will use standard short language codes). We also employ MT from external providers in certain specific cases, but we will focus on our in-house Moses-based setup in this paper.

2.1. MT Info Service

We now turn to the MT Info Service that is the centrepiece of our MT infrastructure, handling all MT requests to our Moses-based engines from within Autodesk. This service and all its components handle service requests over internal and external network connections.

The first element of this infrastructure are the MT engines that are running on virtual servers in a data centre. These MT engines receive translation requests for individual segments of text (typically sentences) and output translations as soon as they are available. For each language that we use in production, we currently have up to ten MT engines running simultaneously on different servers to provide higher overall throughput. In addition to the actual translation, the MT engines also handle the tokenisation / detokenisation and lowercasing / recasing of the data, as well as any specialised language-specific processing that might need to be performed.

The MT Info Service itself acts as a central dispatcher and hides the details of the MT engines' setup, number and location from the clients. It is the single entry point for any MT-related queries, be it requests for translation, for information on the server setup or administrative functions. It has real-time data on the availability of MT servers for all supported languages and performs load balancing for all incoming translation requests to best utilise the available resources. In real-life production, we often see twenty or more concurrent requests for translation that need to be handled by the system — some of them for translation into the same language. These requests use a simple and easy-to-use API that we devised for client communication with the MT Info Service.

2.2. Integrating MT in the Localisation Workflow

Once we have our MT infrastructure in place and we have trained and deployed all MT engines, we need to make this service available within our localisation workflow so that raw data is machine translated and the output reaches the translators in due course. We use two main localisation tools, namely SDL Passolo for UI and SDL WorldServer for documentation localisation.

On the Passolo side, the data first need to be exported into "Passolo bundles". These are then processed with in-house scripts that send any data that have not been matched against previous translations to the MT info service. The processed bundles are then passed on to the translators for post-editing.

As WorldServer is a Java-based tool, it allows us to build Java-based plugins that provide additional functionality. In particular, we have developed an MT adapter for WorldServer that communicates directly with the MT Info Service and sends all appropriate segments for machine translation. Currently, these are all segments for which no translation memory (TM) match with a fuzzy score higher than 75% exists.

3. A First Approach to Product-Specific Terminology Processing

To support the spectrum of domains represented by our broad product portfolio, we need an effective system that would select product-appropriate terminology during machine translation, as terminology lookup is one of the most time consuming and cognitively intense tasks translators have to deal with. This is particularly true for the data typically found in our software manuals — rich in very industry-specific terminology from the architecture, civil engineering, manufacturing and other domains.

One solution to this problem would be to create product and / or domain specific MT engines that should produce domain-specific output. Unfortunately, most of the localisation volume is concentrated in a few flagship products, while the rest of the products have fairly low amounts of data. Trying to train MT engines only on product-specific data is thus destined to fail, as out of the currently about 45 products that we localise, only about five have sufficient amounts of TM translations for training an operational statistical MT engine.

We could, of course, always train on the whole set of data for each language and only perform tuning and / or language model domain adaptation for each specific product / domain group. However, this would result in as much as 585 different product specific engines (13 languages times 45 products) that need to be maintained, with each further language we decide to localise into adding another 45 engines. The engine maintenance would include regular retraining and deployment, as well as the necessary processing power to have that number of engines (plus enough copies for load-balancing) available around the clock — the latter being particularly important as the software industry moves to agile continuous development of software products, rather than yearly (or similar) release cycles.

The solution we employ instead allows us to only train one MT engine per target language and use built-in Moses functionality to fix the product-specific terminology during a pre-processing step. Until recently, as part of our regular localisation process, product-specific glossaries were manually created and maintained for use by human translators. When new data is sent to the MT Info Service for processing, the MT request includes the corresponding product code, which allows the selection of the proper product-specific glossary and annotating any terms found in the source data with XML tags providing the proper translations. Moses is then instructed to only use these translations when processing the data, thus ensuring that the MT output has the proper target-language terminology for the specified product. Since the end of 2013, we have moved to an automatic approach to the creation of product-specific glossaries, which will be described in detail in the next section.

One drawback of this approach is that the product glossaries only contain one translation per language per term, that is one particular morphological form. This means that for morphologically rich languages — like e.g. Czech — the product-specific terminology will often carry the wrong morphological form. However, we estimate that the time needed to fix the morphology of a term is significantly less than the time needed to consult the glossaries in the appropriate tools to make sure the source terms are translated correctly.

Our approach also allows us to eschew the tuning step during MT training. Given our broad product portfolio, selecting a representative tuning set is particularly hard and necessarily biases the MT system towards some products at the cost of others. Considering these factors, as well as the level of performance of our non-tuned MT engines, we have decided to bypass the tuning step. We thus save computing time and resources, without losing too much in MT quality.

4. The Term Translation Central in Detail

In the previous section, we discussed at a high level our approach to enforcing product-specific translations during MT processing. The main tool supporting this process is the Term Translation Central, which was developed at the end of 2013.

The Term Translation Central[1] is a Python-based web application built using the Flask micro-framework[2] with a REST API. It performs three distinct functions that together provide the term handling functionality that we need: 1) terminology extraction functionality; 2) a web-based UI for use by translators and subject-matter experts (SME); 3) the ability to provide language- and product-specific terms and their translations on demand.

4.1. Term Extraction

This functionality is used whenever new content that needs to be localised becomes available. This happens when the product development teams at Autodesk implement a new product feature, which usually entails the addition of new UI elements or the change of old ones, as well as the creation of new documentation content by the technical writers, describing the functionality and use of the newly implemented feature.

Before going through the regular translation process, this new content is first submitted to the Term Translation Central for terminology extraction as a POST HTTP request using a JSON-based format. The content is submitted once for each language it needs to be localised into and individual term extraction jobs are created. The reason behind this language-specific processing (even though only the English sources are available at this point) is that not all products have been localised into the same set of target languages — that is, the localisation history is different for each language — so the list of extracted terms could differ across target languages. The term extraction is then performed in the background and the requesting client does not have to wait for the term extraction jobs to finish before it can continue with the content processing as necessary.

We have developed this on top of Python NLTK[3], due to its ready availability and ease of use. The submitted content is first POS-tagged (Part-of-Speech), using the Brill POS tagger. The tagger is trained on the corpora generally available with NLTK, augmented with

[1] `http://langtech.autodesk.com/ttc`

[2] `http://flask.pocoo.org`

[3] `http://www.nltk.org`

manually POS-tagged Autodesk-specific data. A small number of chunking rules are then used to identify nouns and noun phrases based on the POS annotation — the presumption being that the majority of meaningful terms will be of this type — thus creating a first draft list of potential new product-specific terms.

Next, we need to filter the draft list of terms to exclude the ones that have been translated before for the requested language/product combination. Previously translated terms should by the time of the current processing be handled properly by our MT infrastructure and their translations are discoverable by our translators. For this, the Term Translation Central queries one of our other systems — NeXLT[4] — that contains an index of all software and documentation segments that have been translated at Autodesk as part of the localisation process. If a potential term is discoverable on NeXLT (regardless if on its own or as part of a larger segment), it is filtered out. Even if this particular term's translation isn't being enforced as a consequence of an earlier term processing effort, the translators can always look it up on NeXLT should they be unsure about the proper translation. The terms that remain after the filtering process are labelled as either 'new to corpus' — when they were never translated before in any Autodesk content — and 'new to product' — when they are discoverable in NeXLT but only for products different to the one being processed.

After finishing the filtering process, the final list of terms is stored in a database and the term extraction job's status is set to 'processed', allowing translators to start work on the translation of the extracted terms.

4.2. Term Translation Process

When a term extraction job has been processed, the list of extracted terms is accessible via the Term Translation Central web-UI and translators use this as the platform for terminology translation. In addition to the extracted English terms, the UI presents on demand the context for each term (the segment(s) from which the term was extracted), any comments the translators or reviewers may have entered and a history of the translations of this term.

The translations of the terms 'new to corpus' are particularly important, as they will probably end up being used for other products, too. Here, translators will often need additional knowledge to come up with the proper translations and the Term Translation Centre supports this by providing the option for an SME to login to the website and modify the term translations as necessary.

Specifically to support communication between the SMEs and translators, each term has its's own dedicated comment thread where questions/answers and explanations of the translations can be listed, each labelled by the commenter's name.

For the terms 'new to product', translators are instructed to consult NeXLT before translating and to only introduce new translations for the terms, if there is a clear necessity to have a product-specific translation different from what is discoverable in NeXLT. The translators also have the option to mark extracted terms as 'ignore', in which case they do not need to translate these terms. This is used both for cases where the extracted term is too generic,

[4] `http://langtech.autodesk.com/nexlt`

and for cases where a term candidate was obviously a result of extraction error (e.g. only a partial phrase, or not a noun phrase, etc.).

All existing terms and their translations are publicly available, but a user login is required to enable editing and commenting functionality.

4.3. Term List Availability

Once the translation for a particular term is finalised, it needs to be labelled as 'approved' in the Term Translation Central UI. Immediately after this change is saved by the user, the term becomes available for use by other systems.

As a first step, the newly approved term translation is pushed to NeXLT for indexing, so that translators can only consult one source for terminology queries during content translation.[5] The translators also have the option to manually export a TBX file containing approved term translations for the language/product combination(s) they are interested in. This TBX file can subsequently be, for example, imported in to Xbench[6] or similar tools for translator use. Finally, the the Term Translation Central supports a REST call through which the MT Info Service can request the list of term translations for a specific language/product combination to be enforced during machine translation.

5. Future Work

The Term Translation Central is still a relatively new tool in our localisation toolchain and is only coming to full production use for the 2014/15 localisation cycle. As such, there are still a number of development avenues that we plan to explore in the short/mid term timeframe.

One of the first upcoming improvements we plan on will be to present to the translators a raw MT version of the term contexts for the extracted terms on the Term Translation Central. This will allow them to see whether our current MT engines can already produce the correct translation for the terms in question or not. If the MT output is already correct, then the translator does not need to enter a translation for the term and we do not need to enforce it during MT.

Currently, the terminology process described in this paper is only used in production for handling software UI content. One of the main reasons why we have not yet introduced it for documentation content is that we do not yet have enough data on how it would affect the continuous translation model used for documentation localisation to effectively process products developed using agile practices. For the 2014/15 localisation cycle we plan to introduce the necessary connectors that would allow the submission of new documentation content for terminology extraction, but without requesting the translation of the extracted terms. This approach will allow us to gather statistics on the number and type of terms extracted from documentation content that would allow us to later decide on the best approach for integrating our terminology processes with the continuous documentation translation model. We also see the capturing of terms from UI content as the most crucial step, as this content would usually

[5] This functionality was disabled after a recent system redesign, but will be reintroduced shortly.

[6] `http://www.xbench.net/`

contain most of the specific terms discussing the newly implemented product features. UI content is also usually localised well before documentation, so the documentation localisation will benefit from the application of the terminology process to UI content.

Another important opportunity for improvement we see relates to the actual term extraction process. In particular, currently the POS tagger we use is trained mostly on news data and on a limited amount of Autodesk-specific data. These latter data were manually annotated by a single Autodesk employee within a limited timeframe. The improve the quality of the POS tagging, we plan to have 10–20 thousand Autodesk-specific segments manually annotated by a language service provider. These high-quality annotated data will then be used for the POS tagger training and we expect this to result in significantly fewer false positives during term extraction.

A research question that we plan to address in the near future is to what extent does the enforcing of product-specific terminology in MT output help translators work faster and/or be more consistent. We especially need to evaluate the impact of translation memory matches here, as we do not currently have a way to enforce product-specific terminology in that content and it represents about 40% of the volume of post-edited documentation content for us (the rest being post-edited MT output).

We also plan to develop a term lifecycle management guidelines that would clearly state the conditions, under which a particular term processed in the Term Translation Central should not further be enforced during MT — say if we can reliably expect the MT engines to produce the correct translation without explicitly using the terminology data.

6. Conclusions

In this paper, we gave a short overview of the MT infrastructure at Autodesk and presented the Term Translation Central — a multi-faceted tool for processing product-specific terminology extracted from new content. This is our solution to ensuring our in-house MT engines are always up-to-date and ready to handle new content that needs to be localised, as an alternative to other approaches like incremental/on-line retraining and/or domain adaptation.

The Term Translation Central was built at the end of 2013 and after a successful pilot deployment is now an integral part of the localisation process at Autodesk. The tool has an intuitive and modern web UI that enables translators to easily work on terminology translation, collaborating effectively with subject-matter experts to settle upon the most appropriate translation in each case.

As a relatively new development at Autodesk, this tool has a lot of growth opportunities and also presents us with a number of interesting research and analysis questions we hope to answer in the near future.

The source code of the tool is available at `http://github.com/venyz/Terminology`

References

Koehn, Philipp, 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. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In Proceedings of the Demo and Poster Sessions of the 45th Annual Meeting of the Association for Computational Linguistics (ACL '07), pp. 177–180. Prague, Czech Republic.

Zhechev, Ventsislav. 2014. Analysing the Post-Editing of Machine Translation at Autodesk. In Post-editing of Machine Translation: Processes and Applications, eds. Sharon O'Brien, Laura Winther Balling, Michael Carl, Michel Simard and Lucia Specia, chap. I, pp. 2–23. Newcastle upon Tyne, UK: Cambridge Scholars Pubishing.