Towards Application of User-Tailored Machine Translation in Localization

Andrejs Vasiļjevs
Tilde SIA
andrejs@tilde.com

Raivis Skadiņš
Tilde SIA
raivis.skadins@tilde.lv

Inguna Skadiņa
Tilde SIA
inguna.skadina@tilde.lv

Abstract

Although machine translation is very popular for various personal tasks, its use in business applications, including localization, is still quite limited. This paper describes the facilities of the LetsMT! platform for localization industry professionals and the results of an experiment which explored the use of the SMT system integrated into translation memory for an actual localization task. We present, LetsMT! platform from the perspective of localization, results of a user requirement analysis and our experiment of evaluating an English-Latvian SMT system integrated into SDL Trados tool. We show that such an integrated localization environment can increase the localization productivity by 32.9% without critical decrease in quality.

1 Introduction

Growing pressure to reduce translation costs and to increase translation volumes motivates the localization industry to embrace machine translation (MT) in addition to other widely used computer assisted translation tools (CAT).

For several decades, the most widely used CAT tools in the localization industry have been Translation Memory systems (TM). Since Translation Memories contain fragments of previously translated texts, they can significantly improve the efficiency of localization work in cases when new text is similar to previously translated texts. However, if a text is from a different domain than the TM or in the same domain from a different customer, using different terminology, benefit from such TM is minimal.

The localization industry has experienced increased pressure to provide more efficient services, particularly due to the fact that volumes of texts that need to be translated are growing at a greater rate than the availability of human translation, and translation results are expected in real-time. For this reason, the localization industry is increasingly interested in combining translation memories with machine translation solutions adapted for a particular domain or customer requirements.

Developers of TM systems recognize benefits from the application of machine translation in the localization industry. Some developers have already integrated machine translation in their products or they provide such solutions to MT developers. For instance, SDL Trados Studio 2009 supports 3 machine translation engines: SDL Enterprise Translation Server, Language Weaver, and Google Translate. ESTeam TRANSLATOR and Kilgrey's memoQ are other systems providing the integration of MT.

For the development of MT in the localization and translation industry, huge pools of parallel texts in a variety of industry formats have been accumulated. The most successful data collection effort is the online repository of TM data by the TAUS Data Association[1]. However, the use of this data alone does not fully utilize the benefits of modern MT technology.

Although the idea to use MT in localization process is not new, it has not been explored widely in the research community. Different aspects of post-editing and machine translatability have been researched since the 1990s (e.g.,

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

Ventsislav Zhechev (ed.): *Proceedings of the Third Joint EM+/CNGL Workshop "Bringing MT to the User: Research Meets Translators" (JEC '11)*, pp. 23–31. Luxembourg, 14 October 2011.

Berry 1997, Bruckner and Plitt 2001). A comprehensive overview of research on machine translatability and post-editing has been provided by O´Brien (2005). However, this work mainly focuses on the cognitive aspects, rather than on localization productivity.

Increasing the efficiency of the translation process without degradation of quality is the most important goal for a localization service provider.

In recent years, several productivity tests have been performed in the translation and localization industry settings at Microsoft, Adobe and Autodesk. The Microsoft Research trained SMT on MS tech domain was used for 3 languages for Office Online 2007 localization: Spanish, French and German. By applying MT to all new words, on average a 5-10% productivity growth was obtained (Schmidtke, 2008).

In experiments performed by Adobe, about 200,000 words of new text were localized using rule-based MT for translation into Russian (PROMT) and statistical machine translation (SMT) – for Spanish and French (Language Weaver). Authors reported an increase of translator's daily output by 22-51% (Flournoy and Duran, 2009).

At Autodesk, a Moses SMT system was evaluated for translation from English into French, Italian, German and Spanish by three translators in each language pair (Plitt and Masselot, 2010). For measuring translation time, a special workbench was created to capture keyboard and pause times for each sentence. Authors reported that although all translators worked faster when using MT, the proportion varied from 20% to 131%. They concluded that MT allowed translators to improve their throughput on average by 74%.

This paper describes the facilities of the LetsMT! platform (Vasiljevs et al., 2010) for localization industry professionals[2] and results of an experiment on using a translation SMT integrated into TM in a professional localization company. We present the results of a user requirement analysis, description of the LetsMT! platform from the perspective of localization and our experiment on the application of an English-Latvian SMT in localization by using LetsMT! plug-in for SDL Trados 2009 translation environment. In the localization experiment, we measured performance of a translator working with and without MT. In addition, quality assessment was performed according to standard internal quality assessment procedure.

2 Overview of the LetsMT! Project

The aim of LetsMT! project is to exploit the huge potential of existing open-source SMT technologies by developing an online collaborative platform for data sharing and MT building. This platform supports uploading of public and proprietary MT training data and building of multiple MT systems by combining and prioritizing data.

The number of open-source parallel resources is limited, which is a critical problem for SMT, since translation systems trained on data from a particular domain, e.g., parliamentary proceedings, will perform poorly when used to translate texts from a different domain, e.g., news articles. At the same time, a huge amount of parallel texts and translated documents are at the users' disposal and such texts can be used for SMT system training. Therefore, the LetsMT! online platform provides all categories of users (public organizations, private companies, individuals) with an opportunity to upload their proprietary resources to the repository and receive a tailored SMT system trained on these resources. The latter can be shared with other users who can exploit them further on.

The motivation for users to share their resources is based on the following factors:

- participate and make contribution in a reciprocal manner, in a community of professionals and for its goals;
- achieve better MT quality for user specific texts;
- provide tailored and domain specific translation services;
- enhance reputation of individuals and businesses;
- ensure compliance with the requirement set forth by the EU Directive to provide usability of public information in a convenient way for public institutions;

[2] LSPs – localization and translation service providers, organizations with multilingual translation needs, and freelance translators).

- deliver a ready platform for study and teaching purposes for academic institutions.

The goal of the LetsMT! project is to facilitate the use of open-source SMT toolkits and involve users in collecting training data. This will result in populating and enhancing the currently most progressive MT technology and making it available and accessible to users of all categories in the form of sharing MT training data and building tailored MT systems for different languages on the basis of the online LetsMT! platform. The LetsMT! project extends the use of existing state-of-the-art SMT methods, enabling the users to participate in data collection and MT customization to increase quality, scope and language coverage of MT. The LetsMT! platform supports uploading of public and proprietary MT training data and building of multiple MT systems, by combining and prioritizing data. To achieve it, the LetsMT! platform has the following key features:

- uploading of parallel texts for users that contribute their content;
- directory of web and offline resources gathered by LetsMT! users;
- automated training of SMT systems from specified collections of training data;
- custom building of MT engines from selected pool of training data;
- custom building of MT engines from proprietary non-public data;
- MT evaluation facilities.

The LetsMT! consortium includes the project coordinator Tilde, Universities of Edinburgh, Zagreb, Copenhagen and Uppsala, localization company Moravia and semantic technology company SemLab. The project started in March 2010 and should achieve its goals by September 2012.

3 Architecture of the LetsMT! Platform

Figure 1 shows the general architecture of the LetsMT! platform. Its components for SMT training, parallel data collection and data processing are described further in this section. The development was particularly facilitated by the open-source corpus alignment tool GIZA++ (Och et al., 2000) and the MT training and decoding tool Moses (Koehn et al., 2007).

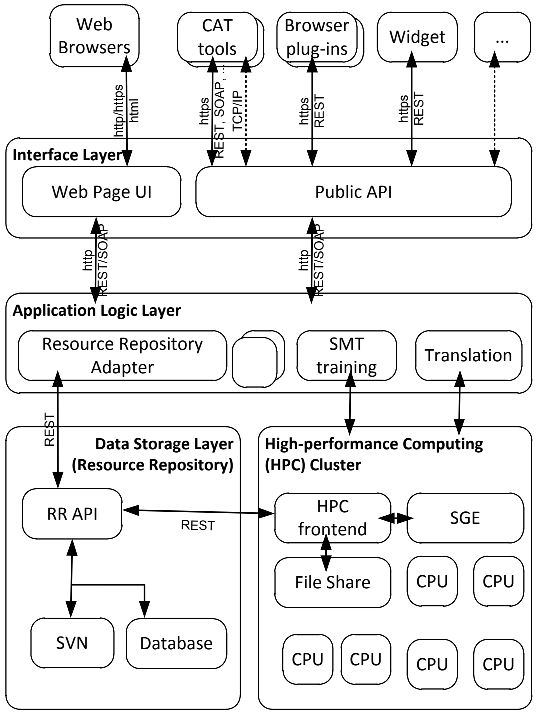

Figure 1. The LetsMT! system architecture

LetsMT! services for translating texts can be used in several ways: through the web portal, through a widget on a user's webpage, through browser plug-ins, or through integration in computer-assisted translation (CAT) tools and different online and offline applications. Localization and translation industry business and translation professionals can access LetsMT! services in their production environments (typically, various CAT tools).

The LetsMT! system has a multitier architecture. It has (i) an interface layer for user interface and APIs with external systems; (ii) an application logic layer for the system logic, and (iii) a data storage layer for file and database storage. The LetsMT! system performs various time and resource consuming tasks; these tasks are defined by the application logic and the data storage and are sent to HPC[3] cluster for execution.

The interface layer provides interface between the LetsMT! system and external users. The system can be used by both human and ma-

[3] HPC – High Performance Computing.

chine users. Human users can access the system through web browsers by using the LetsMT! webpage interface. External systems such as CAT tools can access the LetsMT! system through a public API.

An application logic layer contains a set of modules responsible for the main functionality or logic of the systems. It receives queries and commands from the interface layer and prepares answers or performs tasks using data storage and the HPC cluster. This layer contains several modules such as the Resource Repository Manager, the User Manager, the SMT Training Manager, etc.

The LetsMT! system as a data sharing and MT platform stores a huge amount of SMT training data (parallel and monolingual corpora), as well as trained models of SMT systems. The data is stored in one central resource repository. The LetsMT! resource repository consists mainly of a revision control system (Subversion), a database (TokyoCabinet) and a batch-queuing system (SGE, Oracle Grid Engine). The purpose of the web API is to enable interaction with the repository system for uploading and downloading data, requesting and searching information and triggering batch processes. The LetsMT! resource repository is implemented in Perl and uses the Apache server and mod_perl to handle the requests and responses to and from a client system.

A HPC cluster is used to execute many different data processing tasks, training and running of SMT systems. Modules from the application logic and data storage layers create jobs and send them to the HPC cluster for execution. The HPC cluster is responsible for accepting, scheduling, dispatching and managing the remote and distributed execution of large numbers of standalone, parallel or interactive jobs. The LetsMT! HPC cluster is based on Oracle Grid Engine.

The hardware infrastructure of the LetsMT! platform is heterogeneous. The majority of services run on the Linux platform (Giza++, Moses, Resource Repository, data processing tools). The web server and application logic services run on the Microsoft Windows platform.

The system hardware architecture is designed to be highly sizable. The LetsMT! platform contains several machines with both continuous and on-demand availability:

- Continuous availability – the core frontend and backend services that ensure availability of the LetsMT! webpage and external API;
- On-demand availability – training, translation and data import services (HPC cluster nodes), additional frontend and backend server instances to increase availability.

To ensure scalability of the entire system, the LetsMT! system, including the HPC cluster, is hosted in the Amazon Web Services infrastructure, which provides an easy access to on demand computing resources.

One of important advancements of the LetsMT! project will be the adaptation of the Moses toolkit to fit into the rapid training, updating, and interactive access environment of the LetsMT! platform. The SMT training pipeline implemented in Moses currently involves a number of steps that each require a separate program to run. Within the framework of LetsMT!, this process will be streamlined and made automatically configurable with a set of user-specified variables (training corpora, language model data, dictionaries, tuning sets).

Additional important improvements of Moses that are being implemented by the University of Edinburgh for LetsMT! are the incremental training of MT models, randomised language models (Levenberg et al., 2009), and separate language and translation model servers. We expect some users to add relatively small amount of additional training data at frequent intervals. The incremental training will benefit from the addition of such data without re-running the entire training pipeline from scratch.

4 User Requirement Analysis

In order to develop a well-designed, high quality and easy-to-use system for the localization industry, we started with an overview of user types with respect to job profile, tasks and technical competencies, as well as aspects related to general working conditions, availability of different CAT tools and overall specifications of translation tasks.

A series of questions has been created to describe the interviewee organization. These questions concern a specification of CAT tools (and other tools) applied in the organization together with an outline of the organization's experience with the CAT tools. It is followed by a description of the organization's translation tasks. The description gives information about domains, language pairs and translation volumes, as well as some information about the organization's stored text resources.

Closing questions of this group focus on the localization/translation workflow of the particular organization and specify intellectual property rights of text resources stored in the organization.

21 interviews have been conducted with localization/translation agencies. The replies regarding MT-based translation reflect that MT systems are not frequently used by the respondents.

Only 7 of the respondents replied that they use fully automatic MT systems in their translation practice and only one LSP organization employs MT as the primary translation method. The MT systems that are used vary from SMT systems (Language Weaver and Asian Online) to more traditional rule-based systems, such as Systran and PROMT.

Since 20 out of 21 organizations in the LSP group are business agencies, efficiency in terms of low labor cost is an important parameter. By reducing labor costs, these agencies therefore highly value CAT tools.

Based on the replies about the respondents' use of CAT tools, it is surprising that (full) manual translation is done as much as TM based translation. It should be added that some confusion exists about how human involvement in the translation process should be understood. It seems that some organizations count the human post-editing efforts as human translation. This confusion could explain the larger emphasis on human translation.

5 Application of LetsMT! in Localization

LetsMT! services focus on two application scenarios: (1) MT use in localization and translation, and (2) online MT translation of financial news.

For the localization and translation industry, LetsMT! provides facilities for training of SMT systems on their data and generating customized higher quality MT services based on specific terminology and style required by their customers. It will take into account the workflow, technical requirements and legal ramifications characteristic to the localization industry.

Although the LetsMT! facilities are inclusive and universal, they focus specifically on a number of European languages that currently have no MT services of professional quality: Latvian, Lithuanian, Estonian, Czech, Slovak, Polish, Croatian, and Danish. Thus the initial collection of corpora is focused on parallel texts in the above-mentioned languages and in English in the IT and Telecommunication domain. With release of the first version of the service, the range of domains and languages supported will be largely user-driven, i.e., determined by the requirements and opportunities in the localization and translation market.

For application in the localization scenario, LetsMT! provides a plug-in for the SDL Trados 2009 environment for using generated MT systems. The MT systems run on the LetsMT! platform and are accessible using a web service interface based on the SOAP protocol. Connectivity to additional localization environments will be ensured by providing web services for further integration efforts either by partners or by the user community of the LetsMT! service.

The plug-in has been developed using standard MT integration approach described in SDL Trados SDK. It has been written in .NET (C#), using .NET framework 3.5. The setup is compiled by using Nullsoft Install System (NSIS).

To use the plug-in, a user needs to download a setup file from the LetsMT! website (https://www.letsmt.eu/Integration.aspx) and run it. When the user starts SDL Trados Studio, the plug-in is loaded. Machine translation suggestions from the selected LetsMT! system appear on screen during translation of a document or can be used to pre-translate documents in the batch process. SMT system must be specified manually for each language direction.

6 Evaluation

6.1 Evaluated SMT System

The Giza++ and Moses SMT toolkits (Koehn et al., 2007) are used for data alignment, training of SMT models and translation (decoding).

Total size of the English-Latvian parallel data used to train the translation model is 5.37 M sentence pairs (Table 1). The parallel corpus includes publicly available DGT-TM4 (1.06 M sentences) and OPUS EMEA (0.97 M sentences) corpora (Tiedemann, 2009), as well as a proprietary localization corpus (1.29 M sentences) obtained from translation memories that were created during localization of interface and user assistance materials for software and user manuals of IT&T appliances. To increase word coverage, word and phrase translations were included from bilingual dictionaries (0.51 M units) from high quality reliable sources. A larger selection of parallel data was used which was automatically extracted from a comparable web corpus (0.9 M sentences) and from 104 works of fiction (0.66 M sentences).

Bilingual corpus	Parallel units
Localization TM	~1.29 M
DGT-TM	~1.06 M
OPUS EMEA	~0.97 M
Fiction	~0.66 M
Dictionary data	~0.51 M
Web corpus	~0.9 M
Total	**5.37 M**

Table 1. Bilingual corpora for the English-Latvian system

The monolingual corpus was prepared from news articles from the web and the monolingual part of the parallel corpora. Total size of the Latvian monolingual corpus was 391 M words (Table 2).

Monolingual corpus	Words
Latvian side of parallel corpus	60 M
News (web)	250 M
Fiction	9 M
Total, Latvian	**319 M**

Table 2. Latvian monolingual corpora

[4]http://langtech.jrc.it/DGT-TM.html

Since Latvian belongs to the class of highly inflected languages with a complex morphology, the SMT system was extended within the Moses framework by integrating morphologic knowledge (Skadiņš et al., 2010). The high inflectional variation of target language increases data sparseness at the boundaries of translated phrases, where a language model over surface forms might be inadequate to estimate the probability of target sentence reliably. Following the approach of English-Czech factored SMT (Bojar et al., 2009), we introduced an additional language model over disambiguated morphologic tags in the English-Latvian system. The tags contain morphologic properties generated by a statistical morphology tagger. The order of the tag LM was increased to 7, as the tag data has significantly smaller vocabulary.

We used the BLEU (Papineni et al., 2002) metric for automatic evaluation. The BLEU score of the SMT system is 35.0. The detailed description of test and development sets and system comparison to other English-Latvian systems are given by Skadiņš et al. (2010).

6.2 Evaluation Scenarios

Evaluation in the localization scenario was based on translation performance measurements. Performance was calculated as the number of words translated per hour (Skadiņš et al., 2011).

For the evaluation, two test scenarios were employed: (1) a baseline scenario with TM only and (2) a MT scenario with a combination of TM and MT. The baseline scenario established the productivity baseline of the current translation process using SDL Trados Studio 2009 where texts are translated unit by unit (sentence by sentence). The MT scenario measured the impact of MT in the translation process when translators are provided with not only matches from a translation memory (as in the baseline scenario), but also with MT suggestions for every translation unit that does not have a 100% match in translation memory. Suggestions coming from the MT were clearly marked (see: Figure 2) for translators to treat them carefully.

Typically translators trust suggestions coming from a TM and they make only small changes, if a TM suggestion is not a 100% match. Translators usually are not double-checking terminology, spelling and the grammar of TM suggestions relying that TM should contain good quality data. However, translator must pay particularly careful attention to suggestion coming from MT as it may be inaccurate, ungrammatical, with terminological errors, etc.

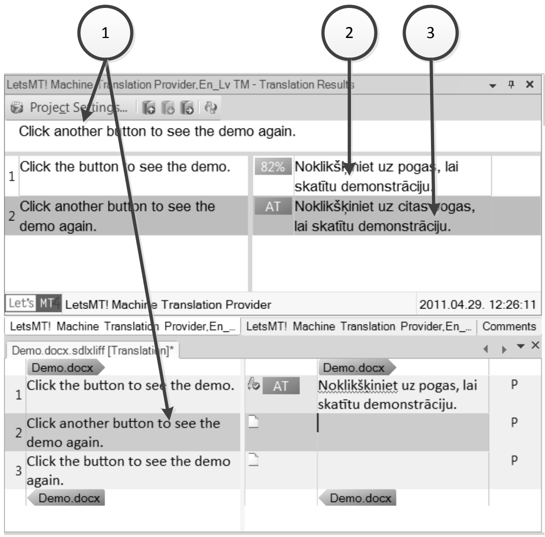

Figure 2. Translation suggestions in SDL Trados Studio 2009; 1 – source text, 2 – a suggestion from the TM, 3 – a suggestion from the MT.

In both scenarios, translators were allowed to use whatever external resources needed (dictionaries, online reference tools, etc.), same as during regular work.

Five (5) translators with different levels of experience and average productivity expectation were involved in the evaluation.

In the MT scenario the first translated document of each translator was removed from the results analysis to avoid "start-up" impact.

6.3 Evaluation of Translation Quality

Quality of each translation was evaluated by a professional editor in the standard quality assurance process of the service provider. The editor was not made aware whether the text was translated using the baseline scenario or the MT scenario. An error score was calculated for every translation task. The error score is a metric calculated by counting errors identified by an editor and applying a weighted multiplier based on the severity of the error type. The error score is calculated per 1,000 words and it is calculated as:

$$ErrorScore = \frac{1000}{n}\sum_{i} w_i e_i$$

where

- n is the number of words in a translated text,
- e_i is the number of errors of type i,
- w_i is a coefficient (weight) indicating severity of type i errors.

There are 15 different error types grouped in 4 error classes: accuracy, language quality, style, and terminology. Different error types influence the error score differently because errors have a different weight depending on the severity of an error type. For example, errors of type *comprehensibility* (an error that obstructs the user from understanding the information; very clumsy expressions) have weight 3, while errors of type *omissions/unnecessary additions* have weight 2.

Depending on the error score the translation is assigned a translation quality grade: Superior, Good, Mediocre, Poor, or Very poor (Table 3).

Error Score	**Quality Grade**
0…9	Superior
10…29	Good
30…49	Mediocre
50…69	Poor
>70	Very poor

Table 3. Quality grades based on the score of weighted errors

6.4 Test Set

The test set for the evaluation was created by selecting documents in the IT domain from the tasks that have not been translated by translators in the organization before the SMT engine was built. This ensures that translation memories do not contain all segments of texts used for testing.

Documents for translation were selected from the incoming work pipeline if they contained 950-1,050 adjusted words each. Each document was split in half and the first part of it was translated as described in the baseline scenario but the second half of the document – as in the MT scenario. The project manager ensured that each part of a single document was translated by a different translator so the results are not affected by familiarity to a translated document.

Altogether 54 documents were translated. Every document was entered in the translation project tracking system as a separate translation task. An adjusted word is a metric used for quantifying work to be done by translators. Larger documents were split into several fragments.

Although a general purpose SMT system was used, it was trained using specific vendor translation memories as a significant source of parallel corpora. Therefore, the SMT system may be considered slightly biased to a specific IT vendor, or a vendor specific narrow IT domain. The test set contained texts from this vendor and another vendor whose translation memories were not included in the training of the SMT system. We will call these texts as *in narrow IT domain* and *in broad IT domain* for easier reference to them in the following sections. Approximately one third of the texts translated in each scenario were *in broad IT domain*.

6.5 Evaluation Results

The results were analyzed for 46 translation tasks (23 tasks in each scenario), by analyzing average values for translation performance (translated words per hour) and an error score for the translated texts.

Usage of MT suggestions in addition to the translation memories increased productivity of the translators on average from 550 to 731 words per hour (32.9% improvement). There were significant performance differences in the various translation tasks; the standard deviation of productivity in the baseline and MT scenarios was 213.8 and 315.5, respectively.

At the same time, the error score increased for all translators. Although total increase in the error score was from 20.2 to 28.6 points, it still remained at the quality evaluation grade "Good". We have not made a detailed analysis of reasons causing an error score increase, but possible explanation could be higher rate of errors in translated segments originating from MT than in translations made from scratch.

Grouping the translation results by narrow/broad domain attribute reveals that MT-assisted translation gives a higher increase in translation performance for a narrow IT domain (37%) rather than for broad IT domain texts (24%). Error scores for both text types are very similar 29.1 and 27.6, respectively.

Grouping the errors identified by error classes reveal the increase in the number of errors, as shown in Table 4.

Error class	**Baseline scenario**	**MT scenario**
Accuracy	6	9
Language quality	6	10
Style	3	4
Terminology	5	7

Table 4. Comparison by error class (error score)

There were significant differences in the results of different translators from performance increase by 64% to decrease by 5% for one of the translators.

Analysis of these differences requires further studies, but most likely they are caused by working patterns and the skills of individual translators.

7 Conclusions and Future Work

Current development of SMT tools and techniques has reached the level where they can be implemented in practical applications addressing the needs of large user groups in a variety of application scenarios.

The work described in this paper promises important advances in the application of SMT in localization by integrating available tools and technologies into an easy-to-use cloud-based platform for data sharing and generation of customized MT.

The results of our experiment clearly demonstrate that it is feasible to integrate the current state-of-the-art SMT systems for highly inflected languages into the localization process.

The use of the English-Latvian SMT in addition to translation memories in the SDL Trados tool lead to an increase of translation performance by 32.9% while maintaining an acceptable quality of translation. Even higher performance results are achieved when using a customized SMT system that is trained on a specific domain and/or same customer parallel data.

Error rate analysis shows that overall usage of MT suggestions decrease the quality of the translation in all error categories, particularly in language quality. At the same time, this degradation is not critical and the result is acceptable for production purposes.

In the future, we plan to make this experiment on a larger scale. We will repeat similar experiments by (i) involving more translators, (ii) translating texts in different domains and (iii) in other language pairs. More detailed analysis of reasons causing an error score increase in MT scenario will also be made.

Acknowledgements

The research within the LetsMT! project leading to these results has received funding from the ICT Policy Support Programme (ICT PSP), Theme 5 – Multilingual web, grant agreement no250456.

References

Berry Mark. 1997. Integrating Trados translator's workbench with Machine Translation. *Proceedings of Machine Translation Summit VI*.

Bojar, Ondrej, Mareček D., Novák V., Popel M., Jan Ptáček J., Rouš J., Žabokrtský Z. 2009. English-Czech MT in 2008. *Proceedings of the Fourth Workshop on Statistical Machine Translation. Association for Computational Linguistics*, Athens, Greece.

Bruckner, Christine and Mirko Plitt. 2001. Evaluating the operational benefit of using machine translation output as translation memory input. *MT Summit VIII, MT evaluation: who did what to whom (Fourth ISLE workshop)*, 61–65.

Flournoy, Raymond and Christine Duran. 2009. Machine translation and document localization at Adobe: From pilot to production. *MT Summit XII: Proceedings of the Twelfth Machine Translation Summit*, Ottawa, Canada.

Hudík Tomaš, Achim Ruopp. 2011. The integration of Moses into localization industry. *Proceedings of the 15th Conference of the European Association for Machine Translation EAMT2011*, 47-53.

Koehn, Philipp, Federico M., Cowan B., Zens R., Duer C., Bojar O., Constantin A., Herbst E. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. *Proceedings of the ACL 2007 Demo and Poster Sessions*, Prague, 177-180.

Levenberg, Abby, Miles Osborne. 2009. Stream-based Randomised Language Models for SMT. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*.

O´Brien, Sharon. 2005. Methodologies for measuring the correlations between post-editing effort and machine translatability. *Machine Translation*, 19(1):37–58, March 2005.

Och,Franz Josef, Hermann Ney. 2003.Improved Statistical Alignment Models. *Proc. of the 38th Annual Meeting of the Association for Computational Linguistics*, pp. 440-447, Hongkong, China, October 2000.

Papineni K., Roukos S., Ward T., Zhu W. 2002. BLEU: a method for automatic evaluation of machine translation, *Proceedings of the 40th Annual Meeting of the Association of Computational Linguistics (ACL)*.

Plitt Mirko, François Masselot. 2010. A Productivity Test of Statistical Machine Translation Post-Editing in a Typical Localisation Context. *The Prague Bulletin of Mathematical Linguistics*, 93(January 2010): 7–16

Schmidtke, Dag. 2008. Microsoft office localization: use of language and translation technology.

Skadiņš, Raivis, Kārlis Goba and Valters Šics. 2010. Improving SMT for Baltic Languages with Factored Models. *Proceedings of the Fourth International Conference Baltic HLT 2010*, Riga.

Skadiņš, R., Puriņš, M., Skadiņa, I., & Vasiļjevs, A. 2011. Evaluation of SMT in localization to under-resourced inflected language. *Proceedings of the 15th International Conference of the European Association for Machine Translation EAMT 2011* (pp. 35-40). Leuven, Belgium

Tiedemann, Jörg. 2009. News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces. *N. Nicolov, K. Bontcheva, G. Angelova, R. Mitkov (eds.) Recent Advances in Natural Language Processing* (vol V), John Benjamins, Amsterdam/Philadelphia, 237-248.

Vasiljevs, Andrejs, Tatiana Gornostay and Raivis Skadins. 2010. LetsMT! – Online Platform for Sharing Training Data and Building User Tailored Machine Translation. *Proceedings of the Fourth International Conference Baltic HLT 2010*, Riga.