



Master

2012

Open Access

This version of the publication is provided by the author(s) and made available in accordance with the copyright holder(s).

Post-editing English to Japanese statistical machine translation : effort and productivity A comparative analysis of three different Moses configurations

Di Rosa, Stéphanie

How to cite

DI ROSA, Stéphanie. Post-editing English to Japanese statistical machine translation : effort and productivity A comparative analysis of three different Moses configurations. Master, 2012.

This publication URL: <https://archive-ouverte.unige.ch/unige:22861>



**UNIVERSITÉ
DE GENÈVE**

**FACULTÉ DE TRADUCTION
ET D'INTERPRÉTATION**

Stéphanie Di Rosa

MA Thesis

**Post-editing English to Japanese statistical machine
translation: effort and productivity**

A comparative analysis of three different Moses configurations

Thesis Director:

Prof. Pierrette Bouillon

Thesis Jury:

Mirko Plitt

Multilingual Information Processing Department

Faculty of Translation and Interpretation

University of Geneva

June 2012

*Statistiquement tout s'explique,
Personnellement tout se complique*

D. Pennac

Acknowledgements

This thesis is the result of collaborative work and would not have been possible without the following people, to whom I would like to express all my gratitude.

My very first thanks go to Prof. **Pierrette Bouillon**, who introduced me to the world of machine translation. Not only did she transmit me all her passion and enthusiasm for this field, but she also created the opportunity for me to do an internship at Autodesk. When I came back from Neuchâtel with a thesis idea, she readily accepted to be the director and guided me throughout this work. I enjoyed these two years of learning and specially the last months of writing during which I could always count on advice and feedback - even handwritten comments on my drafts, which nowadays are so uncommon and thus even more invaluable.

My most sincere thanks also go to **Mirko Plitt**, who, as my internship supervisor in Autodesk first, and as my thesis' jury then, has been an inspiring mentor throughout this year. I am grateful for the possibility he gave me to work with Autodesk data and his support, and above all for his sharing his views and expertise with me, which I believe greatly contributed to the final outcome. It has been a great experience I will treasure.

Special thanks go to one of my former Autodesk colleagues, **Hidenori Yoshizumi**, who has read the drafts and given me much appreciated detailed feedback. His comments allowed me to look at my work from a different perspective, but above all gave me the motivation to go on and improve.

I would also like to thank other people who never denied me their help. These are, at Autodesk, Dr. **Ventsislav Zhechev**, for his technical support with data and for taking the time to answer my Moses and statistical machine translation related questions; and in the Multilingual Information Processing Department in Geneva University, Dr. **Violeta Seretan** and **Marianne Starlander**, for being always kind and ready to listen and give advice, in particular with XML technology and automatic scores respectively.

Once out of the office or the computer lab it was my **friends** in Neuchê and Geneva who were by my side: thank you for our "Wednesday nights", our week-ends and simply for being there for me despite it seemed like I had a new best friend named Moses who was

taking all my time. He will soon be replaced by someone named Lucy, I believe, but you my friends cannot be replaced by anyone!

Last - but not least! -, this thesis was possible thanks to my **parents'** sacrifices and unconditional support. Without them, I wouldn't even have *started* university.

Thanks to you all!

Table of Contents

List of figures, graphs and tables	8
1 - Introduction	11
2 - Machine translation, post-editing and localization.....	14
2.1 Machine Translation (MT)	14
2.1.1 History of MT	14
2.1.2 Types of MT systems.....	15
2.1.2.1 Rule-based approaches (RBMT)	16
2.1.2.2 Corpus-based approaches	18
2.1.3 MT Quality evaluation(s)	23
2.2 Post-editing (PE)	30
2.2.1 Types of PE.....	31
2.2.2 PE effort	32
2.2.2.1 Post-Editing Actions (PEAs)	34
2.3 Localization	36
2.3.1 The localization industry.....	37
2.3.2 A localization project: typical workflow and tools.....	37
2.4 MT today: current challenges.....	40
3 – Machine translation of Autodesk content: the work and research of the Localization Services department	43
3.1 Localization at Autodesk.....	43
3.2 Translation with Moses	44
3.2.1 Computational linguistics resources for SMT	44
3.2.2 Out of the box Moses	45
3.2.3 Customized Moses configurations for EN-JP SMT	47
3.2.3.1 Baseline configuration (NRO)	51
3.2.3.2 Stanford configuration (STANF).....	51

3.2.3.3 Open NLP configuration (NLP)	52
3.3 The Productivity Tests.....	54
3.3.1 The Machine Translation Post-Editing Workbench	55
3.3.2 Results (2009-2011)	58
4 – The analyses: methodology and metrics	61
4.1 Data set	62
4.2 MT Quality evaluation methodology and metric.....	65
4.3 PE productivity measurement	69
4.4 PE effort metric: PEAs adapted to Japanese language	69
4.5 Execution of the analyses	74
5 – Results	77
5.1 MT Quality	77
5.2 PE Productivity	82
5.3 PE effort	84
5.3.1 Average PEAs	84
5.3.2 Recurrent PEAs	85
5.3.3 PE mistakes and rewritten segments	92
5.4 Correlation of PE effort with MT quality and PE productivity	93
5.5 Automatic scores vs. PEAs	97
6 – Conclusion.....	100
Bibliography	102
Appendix A – Reordering rules.....	107
Appendix B – Translation examples	108
Appendix C – XML files.....	111

List of figures, graphs and tables

Figure 1 - Data set characteristics and the questions approached in this work.....	12
Figure 2 - Direct MT (Hutchins and Somers 1992, p.72)	17
Figure 3 - Indirect MT: interlingua method (Hutchins and Somers, 1992, p.74)	17
Figure 4 - Indirect MT: Transfer method (Arnold et al., 1994, p.68)	18
Figure 5 - General statistical machine translation process	20
Figure 6 - Distortion (illustrated with an alignment, from Koehn 2010, p.84).....	21
Figure 7 - Lexical translation probability distribution (Koehn 2010, p. 82).....	22
Figure 8 - Lexical translation probability distribution (Koehn 2010, p. 83).....	22
Figure 9 - The noisy channel model adapted from Manning and Schütze 1999, p.486.	22
Figure 10 - FEMTI: links between user-defined contexts and quality characteristics (Estrella et al., 2005)	28
Figure 11 – Linguistic evaluation of MT - Trigrams in HT vs. trigrams in MT (Aikawa and Rarrick, 2011, p.334)	30
Figure 12 - Classification of translation errors (Vilar et al.,2006)	35
Figure 13 - Parse tree example (Allen, J.,1995, Natural Language Understanding, 2nd edition, Benjamin Cummings)	45
Figure 14 Phrase-based Moses: translation model	46
Figure 15 - Phrase-based Moses: translation process	47
Figure 16 - Phrase-based Moses: translation example	47
Figure 17 - General statistical machine translation process with a pre-processing step (reordering rules).....	48
Figure 18 - EN-JP translation with Moses using reordering rules	51
Figure 19 - Example of a segment parsed with the Stanford Parser.....	52
Figure 20 – Example of a partial tree structure before reordering (STANF).....	52
Figure 21 - Example of a partial tree structure after reordering (STANF)	52
Figure 22 - Example of a segment parsed with the OpenNLP Parser	52
Figure 23 - Example of a tree structure before reordering (NLP).....	53
Figure 24 - Example of a segment after reordering (NLP)	53
Figure 25 - Autodesk's three different Moses configurations for EN-JP translation...	54

Figure 26 - The Machine Translation Post-Editing Workbench's main page.....	56
Figure 27 - PE workbench: a translation job overview	56
Figure 28 - PE workbench: a post-editing job overview	57
Figure 29 - PE workbench: segment to translate.....	57
Figure 30 - PE workbench: segment to post-edit.....	58
Figure 31 - Evaluated MT segments example	69
Figure 32 - Annotated segment example	75
Graph 1 - MT Quality results: all Moses configurations	78
Graph 2 - MT Quality: types of errors (overall)	79
Graph 3 - MT Quality: types of errors (configurations).....	79
Graph 4 - MT Quality: processed segments only	80
Graph 5 - MT Quality: processed segments (translators)	81
Graph 6 - MT Quality: error free segments in every configuration	81
Graph 7 - Productivity (configurations)	82
Graph 8 - Productivity (jobs)	83
Graph 9 - Productivity: post-editing vs. translation	83
Graph 10 - Post-editing actions: averages (configurations)	84
Graph 11 - Post-editing action categories: overall	86
Graph 12- Post-editing action categories: text type 1	90
Graph 13 - Post-editing action categories: text type 2	90
Graph 14 - Post-editing action categories: translators.....	92
Graph 15 - Completely re-edited segments.....	93
Graph 16 - 0 score MT: amount of edited and not edited segments	94
Graph 17 - PE effort related to MT quality	95
Graph 18 - PE effort (categories) to MT quality	95
Graph 19 – Productivity (wpd) and PEAs (number).....	96
Graph 20 - Average time to perform PEAs.....	97
Table 1 - Data: total PE jobs and segments	63
Table 2 - Data: processed segments	63
Table 3 - MT Quality evaluation metric.....	68
Table 4 - Post-editing actions adapted to Japanese.....	72
Table 5 - Post-editing actions: averages (translators)	84

Table 6 - Post-editing actions: averages (jobs)	85
Table 7- Post-editing action types: overall.....	86
Table 8 - Post-editing action categories: configurations	87
Table 9 - Post-editing action types: NRO configuration	88
Table 10 - Post-editing action types: STANF configuration	88
Table 11 - Post-editing action types: NLP configuration	89
Table 12 - Distribution of PEAs over configurations (percentage)	96
Table 13 - Correlation of edit distance with productivity (tentative).....	98

1 - Introduction

This master's thesis will analyse data on the linguistic quality of machine translation output, and the effort and productivity involved in post-editing this output in the context of English to Japanese statistical machine translation of Autodesk documentation.

I came across this topic during an internship at Autodesk Development Sàrl's localization department in Neuchâtel, during which one of my main tasks was to help the team that was working on Japanese machine translation implementation.

At the time of my internship, Autodesk had successfully been using machine translation post-edition in ten languages for about three years and wished to extend the number of language pairs that its localization process could handle, notably by adding English to Japanese. Since, however, implementation of a new language is not desirable until the raw output has attained a linguistic quality such that post-editing is faster than translating, the in-house system's performance – the open source statistical tool Moses – in Japanese translation had to be improved. To do this, three different configurations of Moses were set up and used to generate a data set that was post-edited by four professional translators in the context of a productivity test.

The aim of this work is twofold. First, we would like to gather information on the post-editing effort of the aforementioned data set of English to Japanese machine translation of Autodesk content. Secondly, we want to tentatively explore its relationship with machine translation quality and post-editing productivity figures. In all cases we will adopt a comparative approach, since the data set was translated by three different Moses configurations that we are going to juxtapose.

The main reason for carrying out this work is that Autodesk had productivity figures which it used to assess which of the three configurations was the best candidate for implementation. It had, however, no details on what actually happened during post-editing. It is useful to discover if there are any post-editing patterns because they can provide linguistically informed indications on potential areas of raw machine translation improvement.

Another reason is that the actual impact of machine translation output quality on post-editing effort is an unanswered research question in machine translation of the English to Japanese language pair in our context. Therefore, we also want to see if there is a correlation between MT quality and the type of edits performed (PE effort).

Finally, we want to explore if and how post-editing effort correlates with post-editing productivity.

To do this, we analyzed a data set that presented the following characteristics (fig.1 illustrates the situation):

- The data set comprises English to Japanese raw machine translation output from three different configurations and its four post-edited versions.
- The quality of the output was evaluated (following a metric described in section 4.2).
- The post-edited versions were accompanied by time information that allowed us to calculate productivity figures (following a method described in section 4.3).

The analysis measured post-editing effort by annotating the post-edited versions with an adapted version of “Post-editing Actions” (this metric is described in section 4.4.)

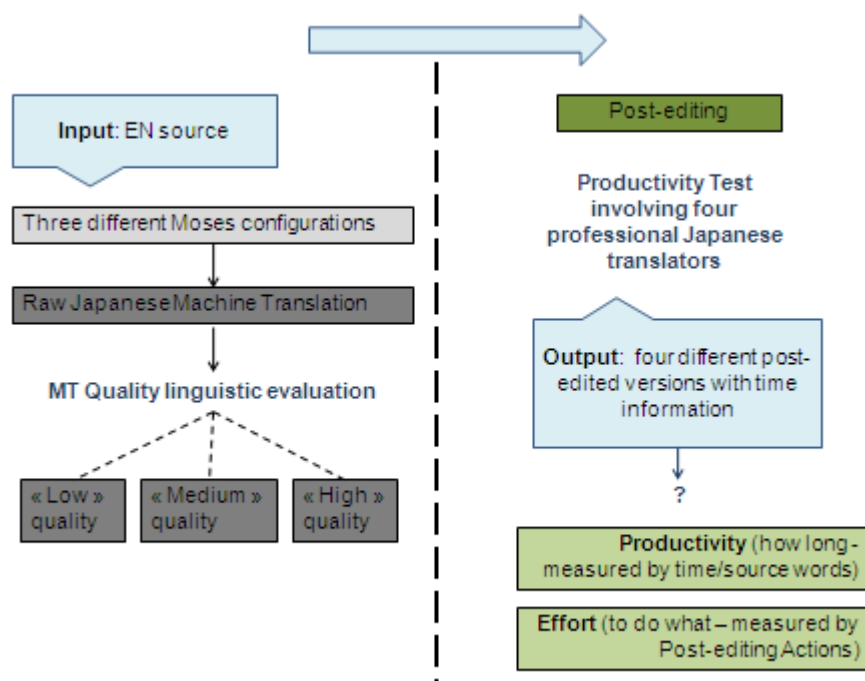


Figure 1 - Data set characteristics and the questions approached in this work.

The remainder of this work is organized as follows. A general introduction to machine translation, post-editing and localization is provided in part 2, which also includes a brief

summary of related work in the field of machine translation quality evaluation and post-editing effort measurement. Part 3 describes our context, or how these same topics are specifically dealt with at Autodesk, and how the Moses translation toolkit was customized for the English to Japanese language pair. Part 4 is the methodology: it contains the data set description and explains the metrics used for this work. In part 5, all of the results are reported. Conclusions and future work belong to part 6, followed by a bibliography and appendices.

2 - Machine translation, post-editing and localization

This chapter will introduce the topics of machine translation (MT), post-editing (PE) and localization in general. Section 2.1 briefly describes MT history and main system types with a focus on statistical machine translation, since it was the type of tool used for this work. Then, the complementary topic of MT Quality evaluation is presented as it is of relevance to our discussion. Section 2.2, which discusses post-editing, provides an introduction to the main PE types and surrounding areas of research – namely, the measurement of PE effort. At the end of the section, post-editing actions as a novel analysis method for PE effort will be presented, as this is the method we will use for our work. Section 2.3, on localization, provides theoretical background and one practical example of what localization is. It briefly describes the localization industry and illustrates a typical workflow and the tools involved. Localization in our context, on the other hand, is presented in chapter 3. The current chapter concludes with section 2.4 on challenges in the field of MT today.

2.1 Machine Translation (MT)

Machine translation is translation between natural languages performed by a computer application (Hutchins, 1986). “Automatic translation” and “machine translation” can be considered synonyms because they refer to the same type of fully automated application, whereas “computer-assisted translation”, abbreviated as “CAT”, refers to a different set of applications, those of translation memories and terminology (data)bases that automate only part of the translation process (L’Homme, 2008). The main object of this MA thesis is machine translation, or MT, while CAT will only be cited as a part of the localization workflow’s tools.

2.1.1 History of MT

The first attempts to translate text with a computer date back to the 1950s (Arnold et al., 1994). It is probably thanks to an American named Warren Weaver that the idea of translating automatically was born. In a memorandum to the Rockefeller Foundation (in 1949) he wrote about the resemblance of language and code, thus implicitly saying that

machine translation would simply be a matter of converting one code into another. Given the recent successes of cryptography during the Second World War, many research projects on MT had been initiated. However, funding authorities were soon disappointed by the results – probably because their expectations had been unrealistic. Their disappointment was made clear in a report commissioned by the US National Academy of Sciences and carried out by the Automatic Language Processing Advisory Committee (ALPAC). In its report, dated 1966, the Academy concluded that MT research was not a good investment, given the impossibility of achieving high standards of quality in a reasonable amount of time. The negative influence of this report, which resulted in the termination of much funding in the U.S. and the demotivation of many researchers in the field, lasted until the late 1970s. At that time, research activity expanded in Japan, and the European Commission acquired the English to French version of Systran and started the setup of the EUROTRA project. At the same time, some governmental and non-governmental organizations became involved in the development of MT systems, such as the Pan American Health Organization, the US Air Force and the TAUM group in Canada. This restored MT's popularity among both researchers and the general public. In addition, fields distinct from but close to machine translation, such as computational linguistics and artificial intelligence, were also explored. Advances in these areas (for example, improvements in parsing techniques) meant advances in MT, contributing to a virtuous cycle of success in research, funding and popularity.

2.1.2 Types of MT systems

An MT system is defined by four characteristics (Hutchins and Somers, 1992): the number of language pairs it handles, the degree of human intervention it requires, the lexical data and the type of rules it applies to translate.

Whether language pairs are many or just one – multilingual or bilingual – largely depends on the system type. By their nature, some systems are designed such that implementing additional language pairs is either impossible or very difficult, while others are specially designed to facilitate the process. Generally, the interlingual method and statistical approach (see below) are best suited to be multilingual, while the direct and transfer methods are typically bilingual.

The degree of human intervention depends on how the system was designed. Early stage systems are often non-interventionist because they can only operate in batch mode,

i.e. once a job has been prepared and submitted for translation, no intervention from the user is possible until the output of the given job is ready. When interaction is possible during translation, the system is interactive, meaning the user may intervene at the disambiguation stage. This kind of disambiguation differs from submitting a source text that has been controlled because in an interactive environment, it is the computer that prompts the user to provide answers. (Hutchins and Somers, 1992).

In MT, lexical data are the entries in the dictionaries, or lexicons. The organization of lexical information depends on the system type. For example, there can be bilingual or monolingual lexicons that contain different kinds of information useful for analysis or transfer, and specialized lexicons (high frequency vocabulary, specific domains). When computers had limited storage and computing capacities, mechanization of dictionaries was a problem.

Types of rules that are applied roughly correspond to the system type. There are two main families: rule-based systems and corpus-based systems. Their approach is different in that the former are based on some kind of linguistic knowledge of the language pairs at hand and follow translation rules that rely heavily on this linguistic knowledge, while the latter retrieve examples of translation in previously processed source and target material. The characteristics of the two main families are described in more detail below.

2.1.2.1 Rule-based approaches (RBMT)

Rule-based systems can have a “minimalist” or “maximalist” approach¹ (“direct” or “indirect” according to the terminology used by Hutchins and Somers). If minimalist, they do not attempt to “understand” the source, and translate word by word with the help of a dictionary. This kind of system works better for close language pairs and when the input has been controlled. If maximalist, they tackle source text comprehension and translation problems and attempt to understand the meaning of the source in order to produce a representation that will be translated.

Direct systems are minimalist. They were the first to appear, at the end of the 50s, and are thus sometimes referred to as “first generation” MT systems. The fact that their functioning is rather “primitive” is partly due to the technical limitations of computers at that time. In general, their approach is to carry out a morphological analysis of the words in the input to identify word stems and allow disambiguation. The second step is a look-up in the

¹ BOUILLON, Pierrette, Lectures in “Traduction Automatique 1”, Faculty of Translation and Interpretation, University of Geneva, winter term 2010.

bilingual dictionary to find equivalent target words, and the final step is a local reordering (mainly word-order reordering) to produce the output. These systems are therefore “naive” also from a linguistic point of view, since they lack syntactical analysis and application of grammar rules specific to the target language (Hutchins and Somers, 1992). An example of such a system is Reverso PROMT. General functioning of such a method is shown in figure 2.

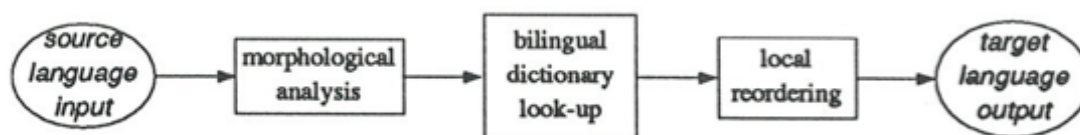


Figure 2 - Direct MT (Hutchins and Somers 1992, p.72)

Interlingual systems are maximalist. They make an abstract, language-independent representation of the meaning of the source text that serves as a basis for target text generation. These systems are best employed when translation between many pairs of languages in both directions is needed, since adding languages is facilitated by the presence of the “pivot language” (the interlingua). However, defining an abstract, language-independent representation of meaning is a very ambitious exercise that has not always proved successful even with closely related languages. (Hutchins and Somers, 1992). General functioning of such a method is shown in figure 3.

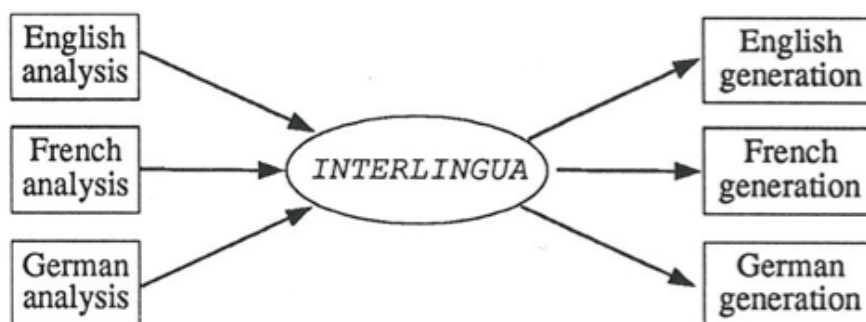


Figure 3 - Indirect MT: interlingua method (Hutchins and Somers, 1992, p.74)

Transfer systems are also maximalist, and represent a variant of the indirect approach. They can be an alternative to the difficulties of defining a language-independent representation for interlingual systems. They make abstract, language-dependent representations of the source first and then of the target. Generation of target text, contrarily to what happens in direct systems, is based on the target representation instead of the

source representation. (Hutchins and Somers, 1992). General functioning of such a method is shown in figure 4.

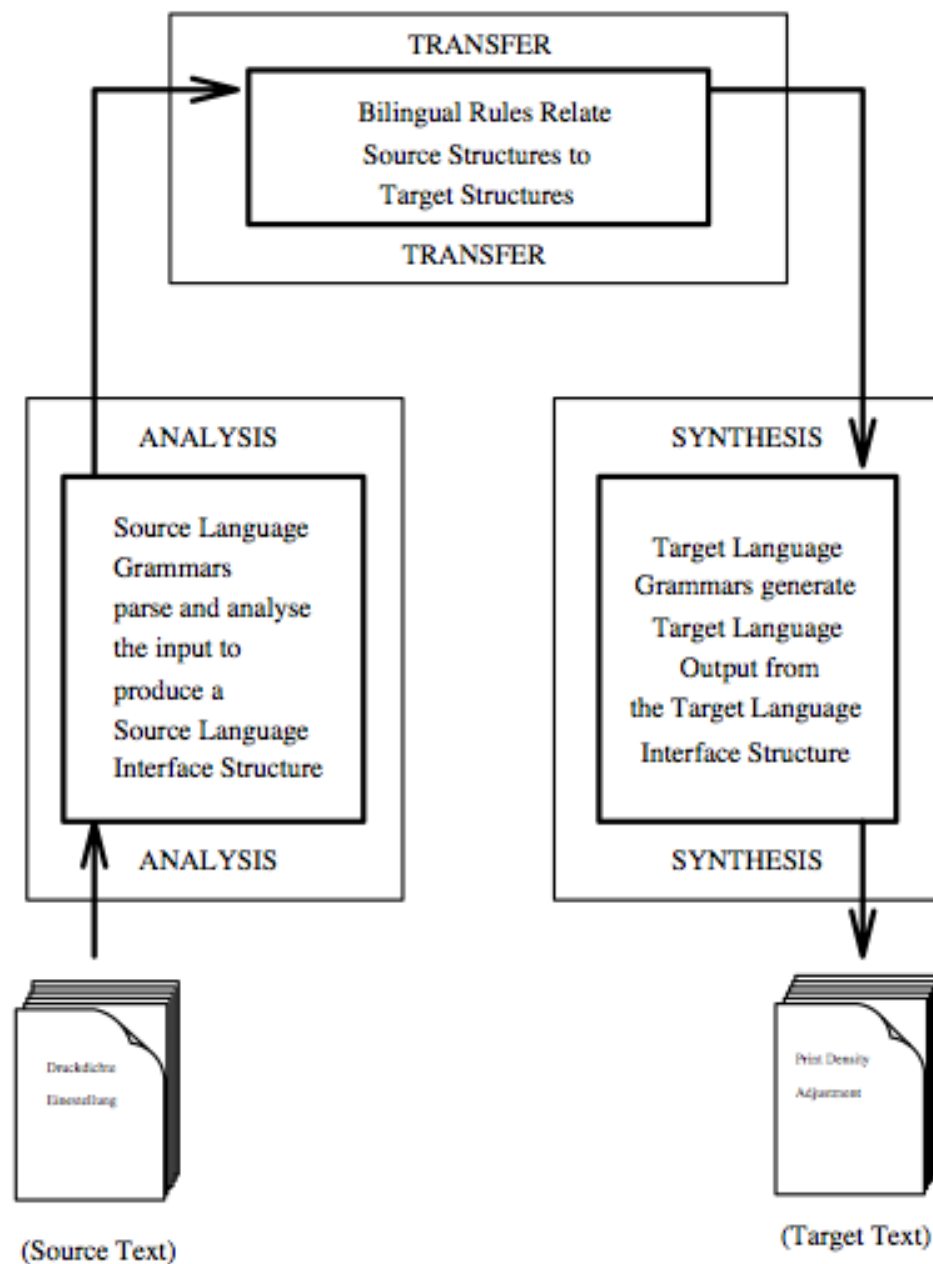


Figure 4 - Indirect MT: Transfer method (Arnold et al., 1994, p.68)

2.1.2.2 Corpus-based approaches

Corpus-based systems can be of two types: example-based and statistical. As their name suggests, their functioning relies on corpora. A corpus is a well-organized collection of linguistic evidence composed of attested language use in a machine-readable format and collected in compliance with a sampling frame and with the goal of being representative. Corpora can be monolingual (one language), comparable (many languages) or parallel (one language and its translation) and collect written or spoken language (McEnery, 2003).

The idea of the example-based approach is that existing translations can be reused: for any sentence to translate, the closest match in a bilingual aligned corpus is retrieved. It is unlike translation memories in that it is not interactive and that the result is the whole translation, not just matching chunks (Somers, 2003).

The statistical approach to MT was actually one of the first to be tested in the 1950s, before it was abandoned in favor of research on RBMT systems. (Brown et al., 1990). Nowadays, it has regained attention and success mainly thanks to the improved capabilities of computers and the extended availability of electronically usable data. Below we will present core concepts of statistical machine translation (SMT), but since advanced or complementary topics such as probability theory are beyond the scope of this work, we advise the interested reader to see Koehn (2010), which this section also refers to.

SMT systems compute the probability (calculated based on a corpus) that a source sentence is translated by a target sentence. This probability is the result of different types of probabilities, the so-called language model and translation model, and of the computing (searching) of the best solution at the decoding stage (see fig.5).

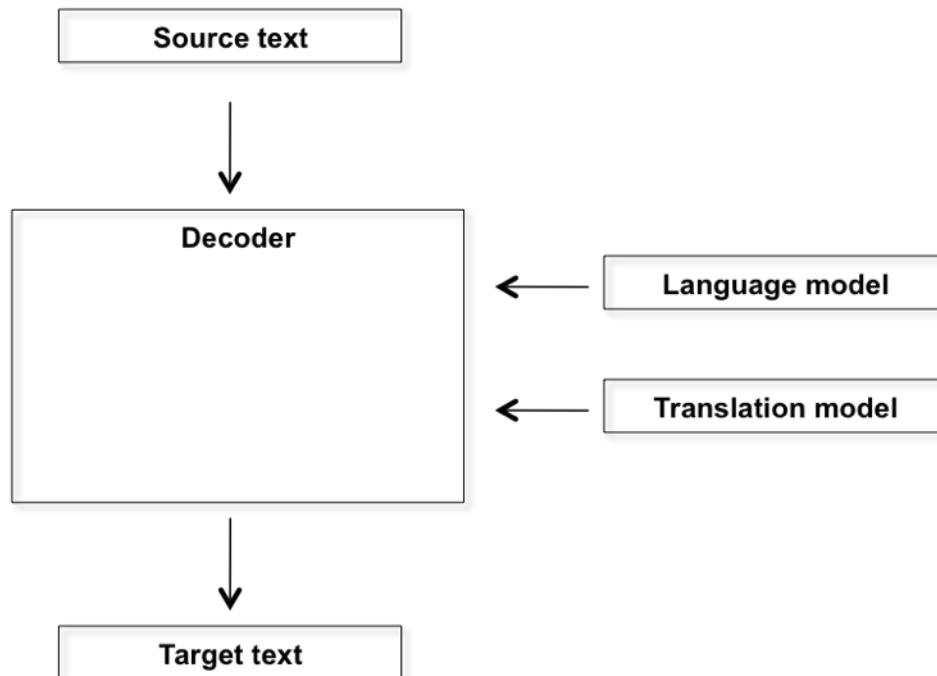


Figure 5 - General statistical machine translation process

A language model is a file that contains statistical information on the probability that the words of a sentence in a given language appear in the order they do. It is computed by looking at bigram and trigram occurrences in a monolingual corpus and it assesses the fluency of a given sentence. Explained in a slightly more scientific (and appropriate) way, it is the probability of the presence of one word in a sentence given the words that precede it, and is an n-gram model². A language model helps to produce fluent output as it also supports the statistical machine translation to make decisions on local word order: the English sentence “the house is small” will get a higher score from the language model than “small the is house” (Koehn, 2010). A good way to explain the role of the language model is the “English to English bag translation” (Brown et al., 1990). Let us imagine a sentence is cut into words that are thrown in a bag: the task of the model would be to reconstruct that same sentence. A language model is useful in the context of translation to make a choice between possible sentences³ among which there might be grammatically correct but nonetheless

² In natural language processing, an n-gram is a sequence of n items from a text (or speech), where n can range from phoneme to words depending on the application (Manning and Schütze, 1999).

³ Possible sentences are possible translations of a source sentence given by the translation model.

“wrong” (because native speakers would not utter them⁴) sentences: the English trigram “a heavy smoker” can be machine translated into French with “a fumeur lourd”, “un fort fumeur” or “un grand fumeur” but it is the language model that tells the machine which one of the above is statistically more likely to be the best choice.

The translation model of a target sentence t given a source sentence s is the probability that t is given by s . In other words, it assesses the faithfulness of translation and does so by computing three parameters on a parallel corpus⁵: fertility, distortion and translation probability.

Fertility is the number of target words that are produced given a source sentence in an alignment: for example, if we translate from English into French “John loves Mary”, fertility for each source word is 1 because “John” produces “Jean”, “loves” produces “aime” and “Mary” produces “Marie”, but in other cases fertility can be 0 or 2. In “John loves nobody”, the fertility of “nobody” is 2 because it produced “ne” and “personne”.

Distortion gives information about the position of a target word in the alignment with reference to its source. It can be visually represented as in figure 6 taken from Koehn:

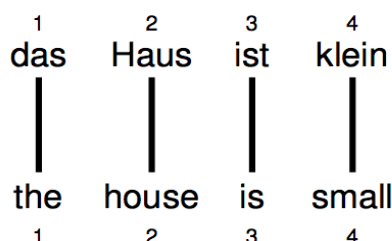


Figure 6 - Distortion (illustrated with an alignment, from Koehn 2010, p.84).

In the example above there is no distortion. If there was one, the first source word would be aligned with a target word different from the first.

The translation parameter is the probability that a given target word is the translation of a source word. For example, once a source word is provided, the first step is to map translations (often, there are several) and collect statistics about the likelihood of the source word being translated as each of its target equivalents. Fig.7 illustrates an example for the German source “Haus” translated into English and the statistics collected in a hypothetical parallel corpus to estimate translation probabilities.

⁴ Or, in SMT, “the corpus does not contain them”...hence the importance of the representativeness, and quality in general, of the corpus.

⁵ Parallel corpus means the corpus is bilingual – one language and its translation – and has been aligned.

Translation of <i>Haus</i>	Count
<i>house</i>	8,000
<i>building</i>	1,600
<i>home</i>	200
<i>household</i>	150
<i>shell</i>	50

Figure 7 - Lexical translation probability distribution (Koehn 2010, p. 82)

The second step would be to find a function that returns a probability for each choice of target word given a certain source (see fig. 8). The probability expresses the likelihood of the translation; it is the translation parameter.

$$p_f(e) = \begin{cases} 0.8 & \text{if } e = \textit{house}, \\ 0.16 & \text{if } e = \textit{building}, \\ 0.02 & \text{if } e = \textit{home}, \\ 0.015 & \text{if } e = \textit{household}, \\ 0.005 & \text{if } e = \textit{shell}. \end{cases}$$

Figure 8 - Lexical translation probability distribution (Koehn 2010, p. 83)

Fundamental underlying processes that need to be introduced are the noisy channel model and generative modeling.

The noisy channel model is what combines the language and the translation models. The SMT engine seeks at decoding stage the source sentence that will return the best value given the language and translation model parameters. To visualize this process, let us look at a schema (fig.9) adapted from Manning and Schütze where s =source sentence, t =target sentence and t' is the most likely target sentence of s given s and t .

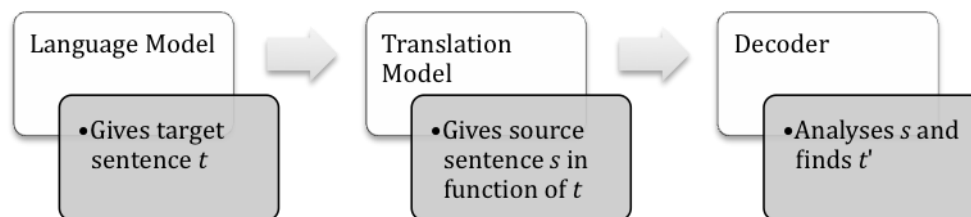


Figure 9 - The noisy channel model adapted from Manning and Schütze 1999, p.486.

The model is called “noisy channel” because we assume the translation process involves sending a message S to a receiver R through a noisy channel that corrupts the

message. To reconstruct the message, information about possible source messages and problems caused by the channel must be used. In the case of translation, these are respectively the language model and the translation model (Shannon, 1948 cited by Koehn, 2010).

Generative modeling is the act of breaking up sentences in smaller units (namely words or phrases) to compute translation probabilities on these units rather than on the sentence as a whole and then combining the result back into a sentence (Koehn, 2010). This is a necessary step because computing probabilities on sentences would not yield good results: sometimes, a given sentence occurs only a couple of times in a corpus. Generative modeling solves this problem, and finally provides the appropriate moment to mention an important difference between early SMT systems and the latest state-of-the-art tools: pioneering work at IBM focused on the word-by-word surface level, meaning that the systems were word-based, i.e. generative modeling resulted in words, and probabilities were computed on words in isolation without further considering the context. However, later work (namely that by Philipp Koehn) introduced phrase-based systems, where generative modeling gives “phrases” that are “multiword units” or “sequences of words” e.g. “to go”, rather than individual words. The main motivation for this was to overcome the word fertility problem whereby it was not possible to have one-to-many translations: if the unit is a phrase, we can translate phrases one-to-one⁶. The phrase-based system introduced by Koehn is Moses. It will be described in some more detail in chapter 3, since it is the engine used at Autodesk for English to Japanese translation.

Research in statistical machine translation has also acknowledged that SMT systems would benefit from the inclusion of some kind of linguistic information during the decoding stage. One such application exists in the form of factored translation models (Koehn, 2010), but given the complexity of the subject and the fact that the Moses implementation at Autodesk does not use any factors, we will leave this topic here and move on to the next: MT Quality evaluation.

2.1.3 MT Quality evaluation(s)

⁶ For the motivation behind phrase-based models see Koehn (2010, chapter 5).

MT quality evaluation is a topic complementary to MT addressed in every introductory work on machine translation. It has also drawn much attention as a study field in its own right, and has been addressed by a vast literature.

It must be understood that the topic is complex, and even confusing at first, since there are in fact many different types of evaluations depending on their object and goal. A distinction must be made between the following evaluation methodologies or types: at the macro level, MT system vs. MT output only; black-box vs. glass-box; context-based vs. functionality based; and at the micro level of MT output, quality assessment vs. error counting and rating; and human metrics vs. automated metrics (to cite only the principal distinctions which will be introduced hereafter).

The diversity of evaluations has been accounted for on two grounds: difficulty and variety of contexts. MT evaluation is admittedly problematic because of its subjectivity: how can we evaluate MT when even human translation quality cannot be assessed with objective measures (Van Slype, 1979)? But while subjectivity might be a major hindrance and make it a difficult practice, the necessity of evaluating MT remains, for two main reasons: on one hand, *fully automatic high quality machine translation* is not possible, therefore one can expect – to some degree – a revision of MT output (Hutchins, 1986). On the other hand, from a business perspective, a customer needs to check whether the service he has paid for has met her/his needs, while the service provider may want to quantify the amount of work (Schiaffino and Zearo, 2009). Therefore, why and how MT evaluations are carried out depends on the specific purposes of the recipients, and this is one of the factors leading to the current diversity of evaluation types.

The first macro distinction that has to be made is the system vs. output evaluation. MT evaluation can be an activity carried out to assess which system should be considered for acquisition and use in a localization or translation department, but it can also be a purely linguistic assessment of the actual output. Often the latter is an important part of the former, but depending on the evaluation scenario and goal, there can be linguistic assessments of raw output outside of the context of a system evaluation. As Flanagan (1994) puts it, “translation quality is only one consideration in the decision to purchase MT software, but for most MT consumers it is both the most important and the most difficult to assess”.

System evaluations can be ‘comparative’ when they contrast two or more alternatives or ‘absolute’ when they aim, for example, to assess improvability over time (Lavie, 2010). Hutchins (1986) calls these evaluations operational or recipient because they are carried out

by potential purchasers to assess MT performance in an operational environment. When the evaluator has access only to input and output to make his decisions, the evaluation is called “black-box”, as opposed to “glass-box”. Glass-box evaluations are likely to yield interesting results as problems will be linked to their origin, while black-box ones have the merit of allowing comparison of otherwise incomparable systems that have different architectures (ibid.).

Linguistic evaluations, on the other hand, take a closer look at the output (usually raw, not post-edited, output) of the software, be it at the development stage, before acquisition, or at runtime, depending on the situation. In fact, outside of the system evaluation context by potential purchasers, researchers or system developers might want to test the prototype or see the effect of any given change to the system during development. Sometimes test-suites are used for this purpose (Arnold et al., 1994).

The second macro distinction that has to be made was introduced by King’s work on evaluation and metrics. In a 2005 article looking back on the history of MT evaluation and ahead at its future, King explains that up to the mid 1990s, two paradigms of evaluation had coexisted: the context-based and functionality focused evaluation. The former, which is also the oldest paradigm, would consist of individual tailor-made evaluations of specific systems on behalf of their users based on the assumption that one context of use calls for one set of requirements and, thus, evaluation criteria. The latter, on the other hand, would focus on the functionality of systems alone, with particular attention on how to assess it with valid metrics. Today, the two paradigms have been combined into a single ISO model (see end of section), but for forty years they were in conflict before it was acknowledged that they should both be considered and applied in the domain of MT evaluation.

The other distinctions are at the micro level of linguistic evaluation of the output.

In this context one can find quality assessment vs. error counting (or analysis) and rating. When we read “quality” in the context of MT evaluation, it is often if not always a reference to the linguistic characteristics of the output. Trying to define quality, Hutchins and Somers came to the conclusion that it depends on the readership and the use that will be made of the translation (Hutchins and Somers, 1992), but that since it is an unclear, subjective criterion, it should be measured using error analysis, which is one of the most objective ways of measuring something subjective.

Error analysis is a linguistic evaluation method that consists of counting the mistakes made by the system following a previously established error classification. Although it has

the merit of allowing for quantification of the amount of work needed to “fix” raw output, in practice, it still represents a thorny problem because not every evaluator will agree on what constitutes an error, and boundaries between mistakes are often blurred. In other words, error analysis requires much preliminary work on error classification and metrics. Nowadays and to the best of our knowledge, there are not many publicly available data or concrete examples of such error classifications in practice. One such example is given in a case study of a rule-based system by CompuServe (Flanagan, 1994) and it provides a comprehensive explanation of the difficulties that one encounters when designing and running an error analysis-based evaluation: multiple correct reference translations, unclear error boundaries, and unclear error origin. It also stresses the importance of defining new metrics for each new language pair and using a ranking to facilitate the task of checking whether user needs have been met.

The other linguistic evaluation method is quality assessment. It involves the measurement of fidelity, intelligibility and, sometimes, style. As Hutchins and Somers explain, fidelity assessments check whether the target contains the same information as the source, and are sometimes called accuracy.⁷ Intelligibility assessments consider the grammatical and syntactic correctness of a sentence and whether it is clear for the reader. Style, which can be appropriate or not to the content and intention, is now often left aside.

Another topic that cannot be ignored in the field of MT evaluations is human metrics and automated metrics.

One of the first works on MT evaluation was the Van Slype Report, commissioned by the European Union and published in 1979. At that time, evaluations were carried out manually.

Nowadays there is a wealth of academic papers that present automatic evaluation metrics. Human evaluation being necessarily subjective, researchers in MT have tried to address this problem by designing automatic evaluations, which also have the advantage of being less costly and time-consuming. Although sometimes unreliable, even almost unusable – in the case of Japanese, for example: see Isozaki et al. (2010) – or difficult to interpret, they serve their intended purpose and are useful in more than one context (e.g. benchmark tests, tuning of statistical systems, contrast of two versions of one system) (Lavie, 2010).

⁷ I am aware that “fidelity” is a quite problematic concept in translation studies and that many researchers have worked on it for almost half a century. It shall be considered in this work with the meaning given in this sentence.

Important automatic evaluation systems are BLEU⁸, NIST⁹, and METEOR¹⁰, to cite only a few. Their functioning cannot be explained in detail here, but in general, they will give each output segment a score based on a benchmark test set made of reference translations.

The diversity of machine translation usage contexts and the lack of publicly shared research results led to the early conclusion that MT evaluation was best done when designed by the recipient for the recipient's needs: Arnold stated that a good evaluation was done in-house (in a large organization that had translators and had purchased a MT system). But this in turn led to the development of a confusing situation where existing methodologies, as explained above, were too numerous and too complicated to use or re-use. In order to address this problem, a project called ISLE (International Standards for Language Engineering) tried to gather all the information on MT evaluation methods and metrics into one interactive, user-friendly website called FEMTI (The Framework for Machine Translation Evaluation in ISLE)(King et al., 2003). FEMTI provides two taxonomies, one for MT context of use and user needs, and one for possible or desirable MT system quality characteristics any evaluator can use to design an MT evaluation task with the benefit of reusing what has been done before in the field. The two taxonomies are very detailed, each comprising a set of values plus the metrics to measure them. Moreover, the taxonomies are linked: choosing one context of use will result in the activation of a certain set of MT system characteristics, as shown in figure 10 below taken from Estrella et al. (2005). The basis for ISLE's work were, among other things, the findings of the EAGLES project on language processing software evaluation, which was inspired by ISO work on standardizing software quality measurement in general¹¹.

⁸ On BLEU See Papineni K., Roukos S., Ward T., Zhu W.-J. (2001) BLEU: A Method for Automatic Evaluation of MT. Research Report, Computer Science RC22176 (W0109-022), IBM Research Division, T.J. Watson Research Center.

⁹ On NIST see Doddington, G. (2002) Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In Proceedings of the second international conference on human language technology research—HLT 2002, March 24–27, San Diego, CA, pp. 138–145.

¹⁰ On METEOR see Banerjee, S., Lavie, A. (2005) METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In ACL-2005, workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, University of Michigan, Ann Arbor, 29 June, pp. 65–72.

¹¹ For a complete and detailed discussion of FEMTI and its background, see (King et al. 2003; Estrella et al., 2005; King, 2005) and for its implementation see the website:

<http://www.issco.unige.ch:8080/cocoon/femti/st-home.html>.

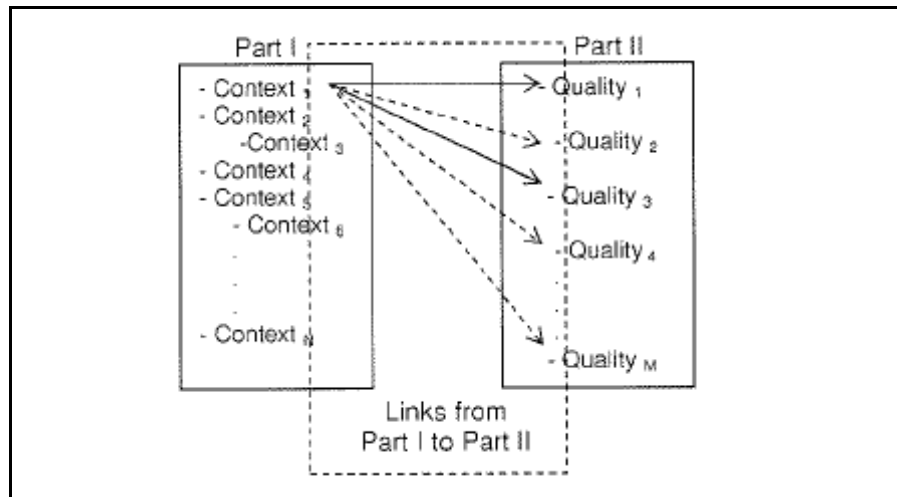


Figure 10 - FEMTI: links between user-defined contexts and quality characteristics (Estrella et al., 2005)

Although it may seem less “up to date” than FEMTI because it predates it, it is worth mentioning one of the results of the EAGLES work, the “7-steps recipe” (EAGLES, 1999). This document presents the group’s findings, that is to say the seven steps that have to be carried out for a successful evaluation of a language processing software: 1) Determine why the evaluation is being done, 2) Elaborate a task model that details relevant users and context of use, 3 and 4) Define what features of the software need to be evaluated, 5) Define metrics for the features to be evaluated, 6) Prepare the actual evaluation, 7) Execute it. As can be seen, the linguistic assessment is not mentioned, since it is up to the evaluator to define what will be evaluated: depending on the user needs, MT output might or might not be rated. As FEMTI was created especially for MT, it handles this aspect in a more complete way.

On the topic of human and automated metrics, allow us to mention that in some contexts it is human assessments of MT output that yield the most meaningful analyses (Lavie, 2010 and Roturier, 2009). In recent years, the weaknesses of automated metrics for MT Quality Evaluation have been underlined and alternative or complementary approaches introduced. While automated metrics can provide fast and free rankings of systems in comparison, they fail to provide qualitative information. Such qualitative information can be supplied by means of linguistic, largely human (or manual) evaluations of MT output. Although the EAGLES and FEMTI works already stated this fact, their goal was not to describe linguistic taxonomies for output analysis and as a matter of fact, such taxonomies were scarce until 2011, when Naskar et al. and Aikawa and Rarrick presented examples of linguistic evaluations.

In their paper Naskar et al. (2011) present a tool and framework for evaluation of MT output¹² based on a taxonomy containing “linguistic checkpoints”; that is, linguistically-motivated units that can range from POS n-grams to ambiguous words, and which represent linguistic phenomena of the source language. These linguistic checkpoints are not detailed further, but the authors claim they can be defined on a case-by-case basis (depending on the evaluation goal and so forth) and allow evaluation of the MT system with regards to specific linguistic phenomena. This entails knowing beforehand which phenomena are problematic for translation. The analysis consists of tagging, aligning and parsing source and target and then comparing the desired checkpoints of source and target with a n-gram based evaluation method. For example, as they put it in their paper, if we want to test the translation quality of noun-noun compounds, all source sentences in the test set containing noun-noun checkpoints are selected, then target-side references are identified (automatically, thanks to alignment information previously gathered) and finally the machine translations are matched against the references with an n-gram evaluation. The more n-gram matches, the better the score is. In the end, this provides information on the strengths and weaknesses of the examined system in the handling of those linguistic phenomena.

The work of Aikawa and Rarrick (2011) is similar, as they also assumed that a linguistic evaluation of the MT output could provide information on the strengths and weaknesses of a given MT system and compared phenomena using an n-gram based metric. The objects of their work were two online statistical MT systems on the English to Japanese language pair. The idea was that counting discrepancies in the number of n-grams of a well-formed human translation with the n-grams of machine translation on the corpus level should provide a valid indication of a) phenomena that are difficult for the system to produce and b) phenomena that are produced by the system but that are ungrammatical or unnatural. Their proposed analysis gives a list of n-grams that appear frequently in the HT and a list of n-grams that appear frequently in the two outputs of the systems being tested. Thanks to the analysis of trigrams, for instance, they were able to identify structures written by translators that machine translation systems realized differently and linguistic mistakes in the MT output. Figure 11 below is taken from their paper and illustrates trigrams that appeared frequently in human translation but not machine translation.

¹² The objects of the comparison were the following MT systems: CoSyne M12, Systran (online version), Freetranslation, Bing Translator and Google Translate.

SMT1	1. できるよう <UNK>	'in order to <UNK>'
	2. いる <u>こと</u> も	'be nominalizer also'
	3. れる <u>の</u> は	'passive nominalizer Top'
	4. ことも ある	'nominalizer also exist'
	5. する <u>の</u> を	'do nominalizer Acc'
	6. ある <u>の</u> は	'exist nominalizer Top'
	7. <UNK> が つい	'Nom unintentionally'
SMT2	1. いる <u>こと</u> も	'be nominalizer also'
	2. <u>こと</u> も できる	'nominalizer also can'
	3. ない <u>場合</u> も	'Neg case also'
	4. ない <UNK> も	'Neg <UNK> also'
	5. サービス により	'service according-to'
	6. の に <PUNC>	'in order to <PUNC>'

Figure 11 – Linguistic evaluation of MT - Trigrams in HT vs. trigrams in MT (Aikawa and Rarrick, 2011, p.334)

However, we should recall that these linguistic evaluations are nonetheless semi-automatic: their object is linguistic, but they avoid relying exclusively on manual work.

In the end, this latest work provides examples on how to take advantage of automatic scoring methods to obtain not an absolute judgment or result of MT quality as a whole, but linguistically informed indications on improvable features of MT.

In conclusion, whilst there are standards and attempts to normalize the practice of MT evaluation, there is no *de facto* universally accepted method. It has been emphasized that in every case, the central question that should be kept in mind and answered when evaluating is “Is this system suitable to the user’s needs?” (Arnold et al., 1994). Designing tailor-made evaluations might be one of the only feasible approaches.

This section concluded the introduction to the topic of machine translation and section 2.2 will introduce one of its natural corollaries, post-editing.

2.2 Post-editing (PE)

In the context of translation and localization, “post-editing” is the (professional) activity of revising machine translation systems’ output to fix it and adapt it to the desired quality standards. It is therefore directly linked to the implementation of MT (Allen, 2003).

It is often emphasized that post-editing is very different from the type of revision done by a senior translator on a junior translator’s work, mainly because of the types of errors involved (Krings, 2001). Although its existence as a professional activity is said to be very recent, PE has existed for at least three decades: it is mentioned in the 1979 Van Slype

report and since then, it has raised many research and practical questions. Unfortunately, publicly available research results are scarce (Allen, 2003).

The literature and research on PE focused first on finding methods for carrying out PE in a cost-efficient and timely manner, and more recently on how to minimize the tediousness of the task.

2.2.1 Types of PE

Post-editing means reading and, if necessary, editing segments that have been machine translated. While, given the quality of MT, post-editing *per se* is considered to be almost always necessary (O'Brien, 2004), according to Allen (2003) to *what extent* one should post-edit MT output can vary and depends on user, volume to translate, quality expectations, turnaround time, perishability of the information being translated and, most importantly, the use of the translated text. Whether the text will eventually be only read to understand the gist of it or published determines whether small or thorough corrections should be made to the target (Hutchins and Somers, 1992); this has elsewhere been called “light” or “full” PE. There are five possible scenarios described by Somers:

1. Only MT, for rough understanding involving no human intervention.
This is referred to as “gisting” and is the case with online free MT: it is not post-edited.
2. MT with rapid PE for rough understanding in cases where the goal is just to have a general understanding of a text, sometimes to see if it is worth having it properly translated. As specified by Hutchins and Somers, this is usually the case when the readership is expert or at least familiar with the subject field and also has linguistic knowledge that allows it to guess where mistranslations are¹³.
3. MT for publication with no PE: the METEO system.
4. MT for publication with minimal PE
5. MT for publication with full PE

One of the biggest concerns is to define “minimal” as opposed to “full”. In scenarios four and five, PE is performed because MT has been implemented. For PE to be regarded as

¹³ This is not the case any more with statistical systems producing perfect output where only a negation is missing. Even a native speaker cannot always tell there is a mistranslation if the source is not given for comparison.

a viable economic alternative to translation from scratch, it should never be slower than translating, while the quality obtained must meet the desired standards. The main risks are either losing time by editing too much (over-correcting) or losing quality by editing too little (under-correcting), or possibly both. What it means in practice to do “minimal” or “full” PE is therefore very difficult to define and usually varies according to internal company guidelines. Even training usually happens at the company department level.

According to Hutchins and Somers (1992), PE should be carried out by professional translators who have access to both target and source segments. In the editor’s introduction to Krings (2001), Koby also states that it requires specific skills and experience in translation. Koehn thinks it could even be envisaged when having access only to the target (“monolingual post-editing”¹⁴). However, Allen reports that in 2003 PE was mainly taken care of by “experienced translators”.

Translators’ point of view or attitude towards PE seems to be negative. According to Hutchins (1986), attitude to PE could be one of the explanations to why PE productivity varies so greatly. There certainly are other reasons, which are being investigated. PE’s unpopularity amongst translators is not unjustified: to this day, the general environment and setting in which it is taking place has not been optimized.

2.2.2 PE effort

As with MT, the problematic aspects with PE are how to quantify it in order to justify its use in a commercial setting and improve it as a professional activity. PE types such as “minimal” or “full” are only prescriptive guidelines on how to proceed. However, to understand the process and evaluate it, a descriptive approach is needed as well.

It is difficult to describe PE in a meaningful way. One of the most complete analyses of PE activity as a process to this day is Krings’, which qualifies PE as being a “temporal, technical and cognitive effort” (2001). Temporal refers obviously to time, technical to the type of keyboard operations done while post-editing, for example deletions, additions and reordering, while cognitive is the intellectual effort needed. The first two are relatively

¹⁴ Koehn, P. (2010) Enabling Monolingual Translators: Post-Editing vs. Options In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*, June, Los Angeles, California, pp. 537–545.

straightforward to measure and a variety of methods have been adopted to try to account for temporal and/or technical PE. Krings' extensive study gathered many results. He used the thinking aloud protocol, a method where post-editors are required to verbalize what they think during their task. As a result, Krings determined post-editing effort by analyzing text similarity of MT and its post-edited version (technical effort), processing time and processing speed of post-editing with or without source text (temporal effort) and the type and extent of cognitive processes used (cognitive post-editing effort). However, cognitive effort has proved difficult to measure in a satisfactory way.

To account for temporal and technical PE effort, some have tried to use automated metrics, such as edit distance (Aikawa et al., 2007). Other ways to go about it included keyboard monitoring and eye-tracking (O'Brien, 2004). Finally, both O'Brien and Plitt assume that pause times are indicators of cognitive effort.

Often in the literature, PE has been compared to or put into relation with other elements, for example: PE effort and MT quality (Krings, 2001) and (Guerberof, 2009), PE time to translation time (see section 3.3), text translatability to PE effort (O'Brien, 2004), controlled language and PE effort (Aikawa et al., 2007).

Krings' findings, that would be confirmed by later studies, namely Plitt and Masselot, were that whereas one can observe that the better the MT quality, the less technical effort has to be made to post-edit, the same does not hold true for time effort and cognitive effort. In other words, MT output quality does not always have the expected positive impact on post-editing in terms of temporal and mental workload.

On this aspect, Tatsumi and Roturier (2010) suggested that in some cases post-editor variance was responsible: edits are the same (technical effort), but time to make them varied across individuals (temporal effort).

Yet another descriptive approach to PE is that of PE typologies. A PE typology is a classification of types of edits performed during actual post-editing. It describes the process and is mainly linked to MT errors, as edits are caused by mistakes in the MT output. A recently developed PE typology is "post-editing actions" by Blain et al. (see sub-section 2.2.2.1).

Either way, PE is an effort. Two questions arise from this fact. One is: how can it be reduced? The intention behind this area of research is to make life easier for post-editors, namely by automatically reducing the number of repetitive edits by pre or post-processing the

MT output. Research went in the direction of implementation of controlled language (a pre-processing step) to reduce PE time (Aikawa et al., 2007 and O'Brien, 2004) and use of regular expressions (a post-processing step) to search and replace (erroneous) linguistic patterns in the output of RBMT systems (Guzmán, 2007). Dugast et al. (2007) used a statistical post-editing system, while Grove and Schmidtke (2009) extracted post-editing patterns automatically to identify the most common edit types.

2.2.2.1 Post-Editing Actions (PEAs)

The intended approach of this work to measure post-editing effort is to analyse post-edited versions manually using a classification scheme developed by Blain et al. called “Post-editing actions” (PEA). This could be considered as a PE typology, as defined above (section 2.2.2).

The work of Blain et al. is based on previous contributions to the field of PE evaluation from (in chronological order) Font Llitjòs et al. (2005), Vilar et al. (2006), and Dugast et al. (2007). The idea of Font Llitjòs was to automatically identify translation rules that need refinement to avoid producing incorrect output and correct these rules: going to the root of the problem. The identification of such rules is entirely based on the performance of post-editing, assuming that frequent corrections to the raw output are a valid indicator of MT errors. The MT system they used is rule-based, so that it is possible to assume a direct cause-effect relationship between translation rules and MT output. This is not really the case with statistical machine translation, but in Font Llitjòs’ approach MT errors and PE actions are put into direct relation, for practical reasons: as stated, “errors nicely correspond to correction actions that can be performed”. They give a scheme of the MT error typology that was adopted and extended by Vilar et al. for their error analysis of statistical machine translation output. The system Vilar et al. used is the RWTH Statistical Machine Translation System¹⁵ and the analysis to evaluate MT output was aimed at better understanding “prominent source[s] of errors” that could not be identified thanks to automated metrics alone. Thus, they carried out a human analysis of SMT output errors using an error classification scheme and a human translation as reference. Although no description of the

¹⁵ For more details see Popovič et al. “The RWTH Machine Translation System for WMT 2009” in Proceedings of the Fourth Workshop on Statistical Machine Translation, ACL, Athens, 2009, although the version used for the paper (2006) might differ.

actual analysis is given, very detailed error statistics are provided and their conclusion is that mistakes are for the most part language-pair dependent.

Their translation (MT) errors classification is given below in fig.12. MT errors correspond to correcting (PE) actions.

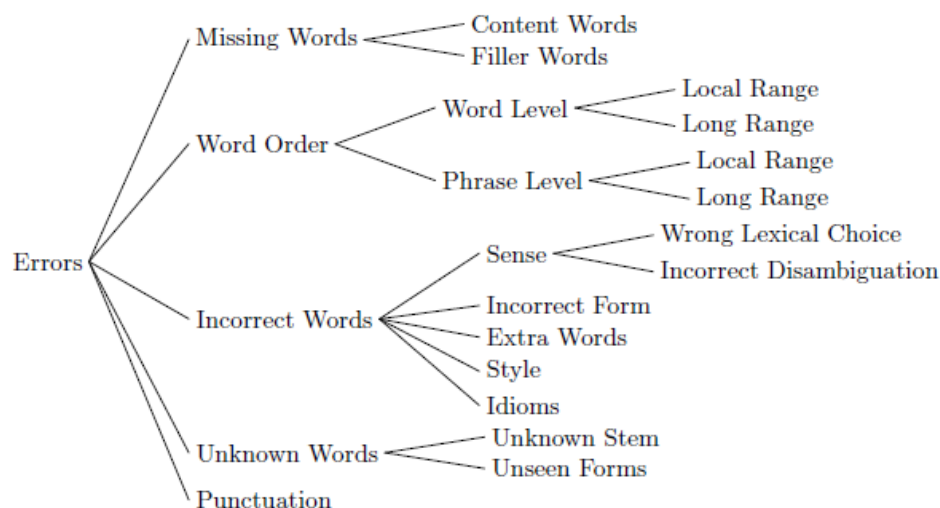


Figure 12 - Classification of translation errors (Vilar et al.,2006)

Last, in their paper on statistical post-editing of SYSTRAN output, Dugast et al. make a linguistic categorization of post-editing changes for their own research. In this case, post-editing was performed by a statistics-based system.

Back to post-editing actions, the motivation for introducing this new PE effort measurement unit is to understand what has been done at the post-editing stage to fix mistakes rather than understand why mistakes occurred in the translation. Consequently, obtaining that information allows for implementation of a module whereby the machine automatically learns about PE effort. To do this, PE activity is analyzed and modeled following PEAs, that is to say “logical edits” as opposed to “mechanical edits” (such as keystrokes or eye movement detection). To explain, let us present the example made by Blain et al.: if the raw output is “le bord est affiché” and was post-edited to “la bordure est affichée” it could be seen as three mechanical edits, but with PEAs it is considered one logical edit whereby if the noun is changed, number and gender change accordingly. As they add, “in that PE, the intent of the post-editor was to correct only one single word, and the

introduction of PEA is to reflect that intent”. Consequently, PEAs have the merit of being more intuitive and of addressing the problem of measuring the PE effort involved when there are multiple word changes. There are two necessary conditions for applying PEAs: that there be no radical changes at the PE stage and that the identification of minimal logical changes be possible.

To link MT quality to PE in some way is a recent trend confirmed by Specia’s work on “Quality Estimation” where quality can be defined in terms of PE efforts and needs. If the quality is perfect, no PE is needed; if it is good, some PE is needed but not to the point that HT would be faster, and if it is bad much PE is needed and HT would be faster.

What all of the aforementioned studies have in common is the assumption (which has been proved right) that PE is faster than human translation (HT) and can result in productivity gains. Return on investment and time-to-market remain in the background and motivate the search for the ideal combination of human and automated PE on controlled or domain-specific MT.

2.3 Localization

Localization is the translation and adaptation of a software or web product to a target locale where it will be used and sold (Esselink, 2000). One can intuitively understand the meaning of locale, which comes from Latin and means “small area”, and replace it in the above definition with “language and country”. However, this would make the definition of localization a redundant one: translating already includes adapting to a different language and culture. In fact, to be precise we shall add that locale refers to a set of standards and rules specific to a language and geographical region that go beyond cultural and linguistic aspects and encompass technical characteristics of hardware and software, and linguistic policies. Therefore, localizing is more than just translating and adapting, as it involves technical activities related to software and the web like engineering and testing. Localization is intertwined with globalization and the expansion of the demand for products, documentation and web sites for an international audience, different from the public they were originally designed for. Therefore, to localize means taking care of the translation, engineering, testing and quality assurance of software, as well as providing online help or documentation for a specific target audience that is identified as a group of speakers of a

certain language in a certain area. Being able to sell on local markets and meeting legal requirements are the key reasons why localization is taking place. Localization is often abbreviated by “L10n”, that is to say “10 letters between L and N”.

2.3.1 The localization industry

The localization industry can be considered the meeting point between the information technology (IT) industry and localization service providers, or language service providers (LSP). This is not a surprising fact, because localizing is a demanding task, and IT industry companies that used to have an in-house translation or localization department in the 1980s have had to turn to outsourcing. Most of the largest localization service providers today were founded in the mid-80s, at the time when businesses realized they had to localize but the task required skills they did not have. As a result, in-house departments were reduced or closed, and the key figure of vendor manager was introduced, to serve as a link between the company and the LSP.

In the mid-90s the consolidation that occurred in the industry resulted in a reduction in the number of localization service providers globally from about 30 to a dozen. Although the industry developed worldwide, for historical reasons one of its centers of excellence and development is Ireland. Not only had the Irish government provided foreign companies with attractive conditions to develop there, but labor costs and supply in the workforce were competitive, and the geographical position of this English-speaking country was also strategic.

Historically and naturally, localization has taken place for language pairs where the source was English and the target was one of the FIGS: French, Italian, German and Spanish, plus Japanese. The reasons for this are that many IT companies are US-based, and their primary markets are France, Italy, Germany, Spanish speaking countries and Japan. Now localization takes place for other languages as well.

2.3.2 A localization project: typical workflow and tools

As explained above, the localization industry is the meeting point of IT companies and localization service providers. LSPs are vendors that combine linguistic and technological expertise and can be divided into MLVs, multi- language vendors, or SLVs,

single language vendors. Localization means "project management", as each activity (translation, engineering, testing...) depends on the others and requires a workflow where every task is organized, not only on the product developer side but also on the vendor side.

Typically, a localization project comprises:

- Project management
- Translation and engineering of software
- Translation, engineering, and testing of online help or web content
- Translation and desktop publishing of documentation¹⁶
- Translation and assembly of multimedia or computer-based training components
- Functionality testing of localized software or web applications.

According to Esselink (2000), a typical workflow will follow the steps explained below, sometimes with more than one task running simultaneously:

1. Pre-sales phase. Competitive bid to be awarded a project by a company: the vendor makes a quotation or project proposal based on the source material provided by the company.
2. Kick-off meeting. After the project has been assigned, a meeting is organized between all staff involved to have an overview of the project.
3. Analysis of source material; scheduling and budgeting. Files for localization are analyzed by specialists or by a "project evaluation team". At this stage, potential problems, an approach, tools and a tentative schedule, a budget and a resource plan are defined. Schedule is one of the most critical points, since a late delivery on the part of the publisher or vendor can compromise release date deadlines.
4. Terminology setup. Product-specific glossaries are created, if they already do not exist, and are sent to the publisher for review and approval before the beginning of the localization project. Such terminology bases should include how to translate and how not to translate.
5. Preparation of source material. This step involves preparing a translation kit, and has

¹⁶ Desktop Publishing definition in the glossary of terms (Esselink, 2000:468): "Formatting and layout of text and images on a computer prior to output on paper, CD-ROM, or any online format". Nowadays printed manuals and the like are less and less common, and are being replaced by online material.

to be multiplied by the number of target languages. Each kit should contain everything the translators need, from reference material to technical information and “translatables”.

6. Translation of software, online help and documentation. Translation is carried out in a precise order. The material that gets translated first is software. Software translation should start with dialog boxes and menus and end with strings because the former usually contain most of the terminology that will be encountered in a particular software. Glossaries that can be created at this stage will be useful at the translation stage of online help and documentation stage.
7. Engineering and testing of software. As soon as translation is done, the localized resource files are compiled into a running application, which will be tested from a linguistic and technical point of view (linguistic testing and functionality testing). Engineering also involves resizing the user interface and assigning hot keys.
8. Screen captures; help engineering and DTP of documentation. Each target language manual and documentation needs to be illustrated with localized screen captures of the software, therefore the stage of preparing the screen captures cannot begin before translation has been carried out. When the graphic aspect of the project is completed the desktop publishing of documentation and online help files testing can start.
9. Processing updates. Updates are necessary because often, translation starts before the English original product has been finalized. Obviously, processing updates after DTP are harder to perform, and technology such as translation memories should be used throughout the process to manage content modifications.
10. Product QA and delivery. Before the localized product is delivered, a sample quality assurance check is performed, including proofreading, software testing and bug or problem reporting.
11. Project closure. Finally, a wrap-up meeting is organized so that the publisher and localization vendor can discuss the quality of deliverables, improvable areas and evaluate the project with future improvements in mind.

To sum up, the tools used in this process are translation memories, content management systems, testing scripts, proofreaders and workflow managers. Machine translation and post-editing increasingly belong to the localization workflow at the pre-translation and translation stage, not to mention the budget- and schedule-planning stage.

2.4 MT today: current challenges

Machine translation has evolved from being an application of natural language processing known only to specialists to being an everyday tool all of us can access thanks to the development of online free translation services such as Google Translate (www.translate.google.com) and Bing Translator (www.microsoft.translate.com). APIs also appeared, simplifying access to automatically translated content. The wealth of data available in electronic and exploitable format (such as corpora) favored the development of statistical machine translation against rule-based systems. If we look at MT from a technological perspective and realize how much technology is shaping its development, it might seem that humans are less and less involved in the process of translation, but in fact they are never really so far away from the process. Despite this undeniable fact, MT is often wrongly perceived as daunting by human translators. Indeed, it has been so for some time, if we consider when this statement in “Machine Translation: past, present, future” was made:

Machine translation should be seen as a useful tool which can relieve translators of the monotony of much technical translation and spare them the wasteful expenditure of much tedious effort on documents of ephemeral or marginal interest. Translators can then be employed where their skills are most wanted, in the translation of sensitive diplomatic and legal documents, and in the translation of cultural and literary texts (Hutchins, 1986:18)

Therefore, one of the main challenges MT has to face today is that of acceptance for what it is and it can offer, be it *as is* or post-edited. As Allen (2003) says, the general public still needs to be educated about the low quality of MT.

But MT perception is not the only area that needs improvement. The abundance of work on MT proves that it is still a very active research field. To have an idea of current research topics, it is sufficient to look at calls for papers whenever one of the many MT summits takes place: MT evaluation is still on the list. In the past ten years, after exploring automated and human metrics, researchers have been attempting to establish the possible correlation between automated and human metrics¹⁷. More recently, the question of what it means to evaluate MT has been seen from a different perspective with the presentation of

¹⁷ On this topic, see for example Coughlin, D. (2003) Correlating Automated and Human Assessments of Machine Translation Quality. In *Proceedings of MT Summit IX*, New Orleans, USA. pp. 63-70.

the concept of Quality Estimation by Specia (2011), which aims to make MT quality estimation more reliable for users who are not fluent in both the source and target.

Maybe worst of all is the fact that, outside the academic setting where research is carried out to provide businesses with this technology, MT is perceived as being difficult to actually implement. According to a contribution by Bourland on *Multilingual* citing a Common Sense Advisory research result, only a few localization departments buy or install a MT system due to the difficulty and cost of choosing one (Bourland, 2011). Many studies and reports tell us that MT is difficult to handle even when implemented (Morland, 2002) so there is room for improvement.

However, we would like to conclude this introductory chapter with a (incomplete) list of examples of current successful implementations in production environments:

- Adobe (Flournoy and Duran, 2009)
- Autodesk's Localization Department – customized open source MOSES system plus TM for post-editing by external vendors allowed for productivity gains (Thicke, 2011)¹⁸
- European Commission – customized SYSTRAN system (Van der Meer, 2003)
- Microsoft – customized MSR-MT system plus TM (Aikawa et al., 2007 and Groves and Schmidtke, 2009)
- Traslan – (Groves, 2008)
- Symantec's Localization Department – customized SYSTRAN system plus TM for post-editing by external vendors allowed for better speed, turnaround time and consistency (Roturier, 2009)

In this chapter we have provided a general introduction to Machine Translation, Post-Editing and Localization to present these topics in themselves and with regard to their complementarities, showing that post-editing of raw MT is now increasingly used in the context of document and software localization processes. Despite the fact that post-editing and machine translation are now a reality in the business world, they are still subject to research. Indeed, the very processes of translating automatically and post-editing have to be improved. On one side, to achieve better system performance on MT we need to evaluate the output quality. This is the task of MT quality evaluation, and we have tried to present both

¹⁸ For more details, see the dedicated chapter 3 in this work.

human and automated metrics, highlighting their differences, advantages and disadvantages. On the other hand, if the localization industry wants to improve the translators' post-editing experience, it needs to better understand that process. This is taken care of by PE effort measurement, and we have briefly summarized research and the latest findings in this area. The next chapter will describe how these topics are specifically dealt with at Autodesk.

3 – Machine translation of Autodesk content: the work and research of the Localization Services department

This chapter provides insight into Autodesk's implementation of machine translation and post-editing in its localization process. First, section 3.1 introduces the overall workflow of the department; then, section 3.2 presents the implementation of the Moses machine translation tool at Autodesk and the three configurations compared in this work; finally, section 3.3 concludes with a description of the PE productivity tests carried out to date and the Post-Editing Workbench on which our data set was post-edited.

3.1 Localization at Autodesk

Autodesk is a U.S. software company that develops computer assisted design programs. Its best known product is "AutoCAD", a computer-assisted drawing and planning tool, but there are many others, such as "Inventor" or "Maya". Its software packages are for modeling, prototyping, simulating and creating. The industries covered include architecture, engineering, construction, manufacturing, automotive, and entertainment, to name a few. As it is based in California, Autodesk develops software and writes technical documentation in English. Its market abroad is one of its key assets and, like many other companies, it localizes to maintain that market.

Currently, localization takes place in up to twenty languages and is being managed by three teams, one of which is located in Neuchâtel, Switzerland. Marketing policy entails a yearly publishing cycle for new software and new versions of existing packages, beginning around March in "sim-ship" mode (i.e. publishing in all the localized languages simultaneously). This accounts for the concentration of workload for the localization department in the quarter preceding launch. Translation, in particular, usually begins in autumn.

The content management system (CMS) tool involved is SDL's WorldServer, which accepts XML format as input. It is the internal localization team that creates translation projects and manages translation memories and terminology bases, as well as MT. There is an in-house quality assurance team that takes care, but only to a certain extent, of linguistically reviewing translated material, with testing being done by the vendors. Feedback on localized products is given by individual countries' sales teams.

3.2 Translation with Moses

The MT toolkit in use at Autodesk and therefore for this work is the open-source statistical Moses system (Koehn et al, 2007). This section describes the “out of the box” Moses (i.e. the default installation) and how it was implemented and further customized at Autodesk. The section starts with a brief summary of computational linguistics resources necessary for understanding statistical machine translation’s pre and post-processing steps, then explains how Moses works in general, recalling the section 2.1.2.2. on SMT functioning, and introduces the three different configurations of Moses that are the object of our analyses.

3.2.1 Computational linguistics resources for SMT

When working with MT, and especially SMT, there are a number of methods and resources involved that are complementary and necessary and come from the field of computational linguistics, which deals with the processing of natural languages by computers. Computational linguistics is a vast area of research and study and it can by no means be covered even partially in this work. However, to provide the necessary minimum background to resources that are frequently mentioned because they are an important part of (S)MT, this section briefly presents tokenization, segmentation, part of speech tagging, and parsing¹⁹.

Tokenization is the process of obtaining tokens from an input text. A token is a string of characters recognizable as a linguistic unit and useful for language processing. It can be a punctuation mark, a word, a number and so on. Tokenization is challenging when the input contains hyphenated words, apostrophes or an ambiguous punctuation, or when the language being tokenized has no clear word boundaries: usually white spaces serve as word delimiters, but many oriental languages, like Japanese, are written continuously.

Segmentation is the process of segmenting an input text into sentences. It is referred to as sentence splitting, and can also be challenging in presence of punctuation marks that usually serve as sentence boundaries (take the dot, for instance), but which can serve

¹⁹ References for this section are the corresponding chapters on these topics by Mikheev, Voutilainen and Carroll respectively found in the Oxford Handbook of Computational Linguistics edited by Mitkov (2003).

another purpose (to indicate a decimal), depending on typographic rules. Segmentation rules need to address the typographic specificities of languages.

Assigning part-of-speech (POS) tags to input tokens means analyzing them from a grammatical point of view and flanking them with an indication of which POS they belong to: noun, verb, article, pronoun, adjective, preposition, adverb, conjunction, or participle.

POS tagging is a complex task for a computer because of the inherent ambiguity of language.

To parse means to carry out the syntactic analysis of an input text given under the form of tokens to extract information about its structure. The information thus obtained is a parse tree, a representation of the structure of the input, shown in figure 13.

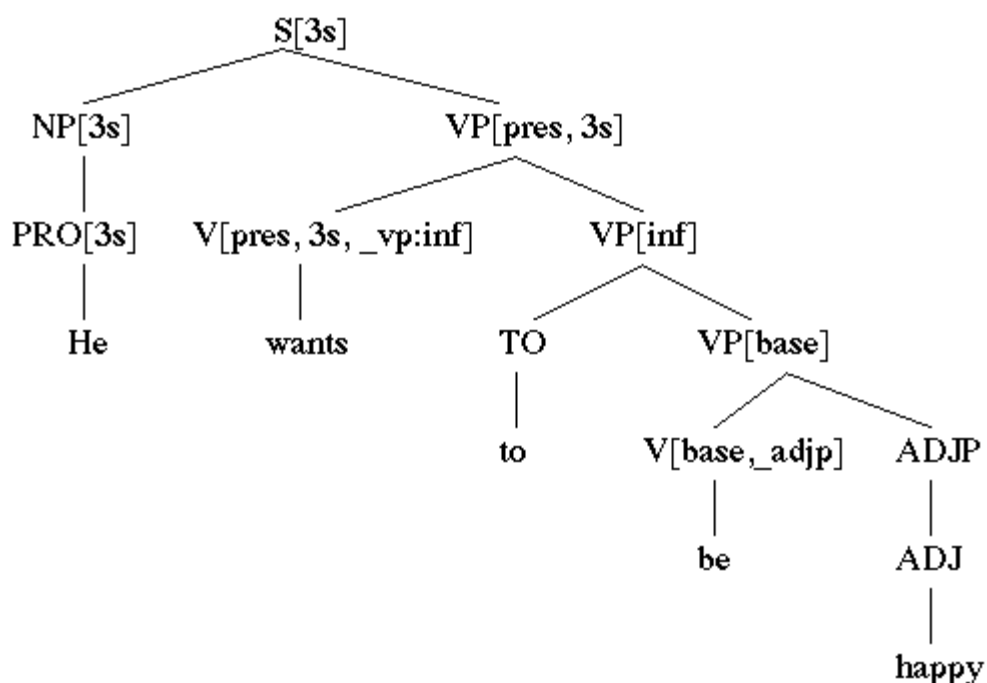


Figure 13 - Parse tree example (Allen, J.,1995, Natural Language Understanding, 2nd edition, Benjamin Cummings)

Section 3.2.3 will describe in some more detail the two parsers used in combination with Moses, namely the Stanford Parser and the OpenNLP toolkit.

3.2.2 Out of the box Moses

The general functioning of SMT was introduced in section 2.1.2.2. Below, we present the “out of the box” Moses, that is to say the default installation. The standard phrase-based translation model of Moses consists of three elements, shown in fig.14.

These are:

1. The phrase translation table, i.e. statistical information on the parallel corpus of source and target. There is at least one of these.
2. The reordering table, i.e. statistical information on the frequency of changes in word order. In other words, this model describes distortion and allows us to handle it, if we wish: it is not compulsory to use the reordering model. If we modify the value of distortion, the input gets reordered.
3. The language model, i.e. a statistical description of the monolingual target language corpus. It can influence a weak reordering model.

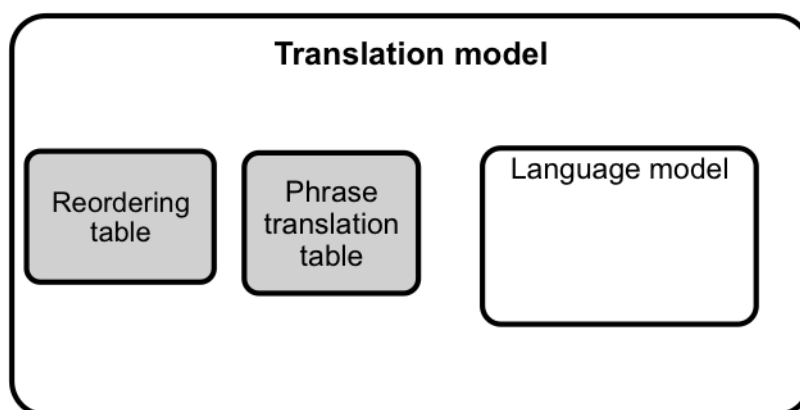


Figure 14 Phrase-based Moses: translation model

At training stage, the engine aligns data and creates the translation model files with it, namely the phrase translation table and reordering table.

At translation (decoding) stage, the engine completes the following operations:

- Source text input is segmented into sequences of phrases.
- The phrases are mapped against the translation phrase table to find possible translations (translation options) and the language model is taken into account for output fluency.
- The phrases are possibly reordered.
- The target text is produced.

This process is presented in a simplified way in fig.15 and one example from the Moses website²⁰ is provided in fig.16.

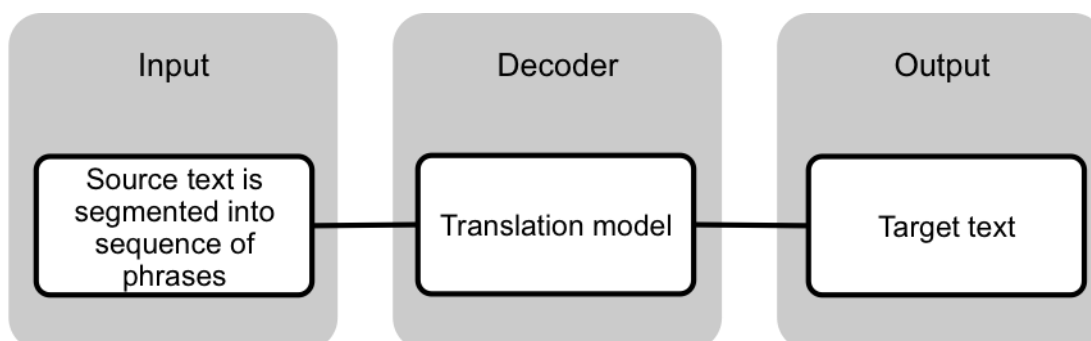


Figure 15 - Phrase-based Moses: translation process

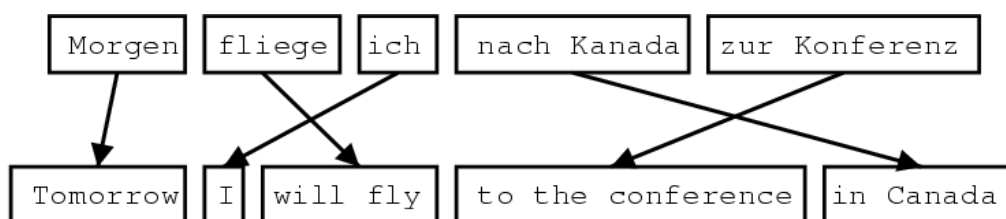


Figure 16 - Phrase-based Moses: translation example

3.2.3 Customized Moses configurations for EN-JP SMT

When translation involves distant languages that require more extensive reordering than what the reordering table can handle, a further customization might be necessary. One of the ways to do this is to introduce a pre-processing step to reorder source input, as shown in fig. 17 and into more detail in fig.18 further below. However, any “customization” of the core Moses system should not be interpreted as a kind of RBMT dictionary entry creation and specialization with immediate effect on the output since, in that respect, SMT engines are less predictable (i.e. the effect of one modification might not be immediately clear).

²⁰ Moses – Main/HomePage, <http://www.statmt.org/moses/?n=Moses.Background>, last visited in April 2012.

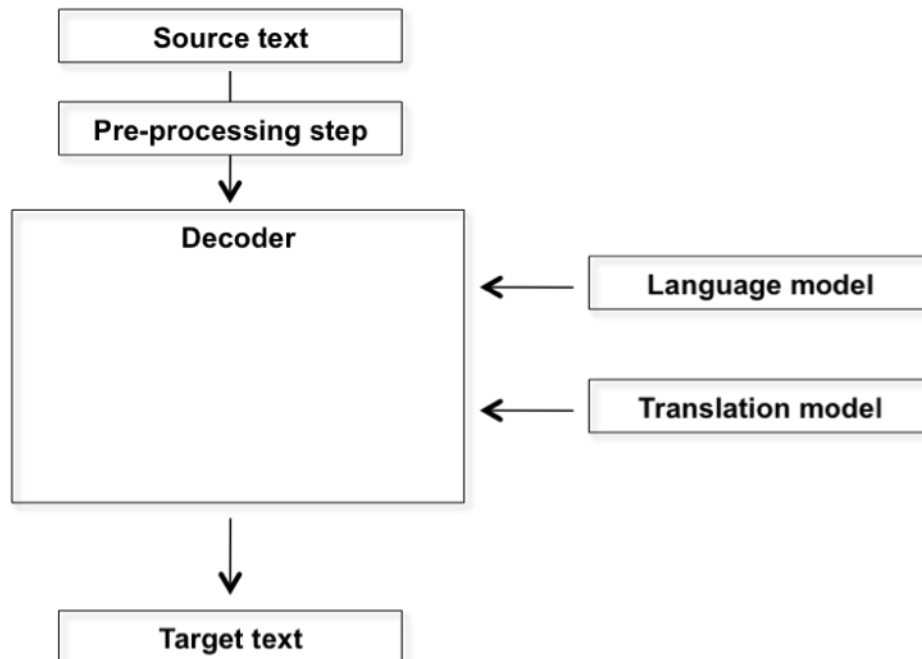


Figure 17 - General statistical machine translation process with a pre-processing step (reordering rules)

The pre-processing step involves first using third-party tools to tokenize and parse the input for POS annotation and then writing a script that reorders input based on POS information.

At Autodesk, early work on English to Japanese SMT has resulted in what is hereafter called the “Moses engine configurations”, where by “configurations” we mean the particular set of pre-processing steps put into practice to reorder source English to make it more suitable to Japanese word order. These configurations are described into more detail in the sub-sections below. Indeed, one of the major problems in statistical MT from English to Japanese is word order: due to the fact that the two languages are very different in this respect, namely what often comes at the beginning of the sentence in English is at the end in Japanese, local and long-distance reordering need to be performed. Long distance reordering means using the adequate distortion limit, but that entails having too great a search space and it is one of the reasons SMT fails (Isozaki et al., 2010). According to research done by Google, “word order is a strong predictor of translation quality” (Talbot et al., 2011). The challenge is even harder if we consider that it is difficult to quickly evaluate reordering performance, and that even though it is desirable to have a means to evaluate reordering independently from lexical choices, the two are in fact linked.

An explanation of how Japanese differs from English and constitutes a challenge for translation can be found in an introduction for language technology professionals and researchers by Kay and Fine.

- Japanese is a subject-object-verb (SOV) language, whereas English has a subject-verb-object (SVO) structure. The verb is at the end, and the verb itself unfolds as a kanji carrying semantic information and suffixes with information about tense, aspect and statement type (positive or negative).
- Modifying clauses, such as relative clauses, precede the principal clause and there are no pronouns or prepositions to make a connection between them. For example “The store he went to” is “彼が行った店”, literally *he-went-store*.
- In Japanese the subject of a sentence and pronouns are often omitted.

Autodesk’s Moses English to Japanese translation model is a standard phrase-based model built with default training configuration and no factors. The training corpus size is about 5.8 million segments from the documentation and software strings of the company alone and the language model was trained solely on target side of this bilingual corpus.

Reordering (to solve the aforementioned problems) is performed both as a pre-processing step (a) based on linguistic information and thanks to the statistical information gathered by the reordering model (b). The pre-processing step (a) is one that has to be programmed by the MT team so that the desired algorithm is applied. These will be referred to as “reordering rules”. The reordering model (b) was built during engine training and is used with its default value during decoding to influence distortion at the local level.

Three configurations were tested, of which one only served as a baseline, with no pre-processing step. What distinguishes the remaining two, the so-called STANF and NLP configurations, is that their reordering rules were completely different. These configurations work as sets of tools and rules: STANF uses the Stanford Parser²¹ to parse the segments and one script (script 1) to reorder; NLP uses the OpenNLP Toolkit²² to parse and another script (script 2) to reorder.

²¹ Stanford Parser: a statistical parser <http://nlp.stanford.edu/software/lex-parser.shtml>, last visited in April 2012.

²² Open NLP <http://opennlp.apache.org/>, last visited in April 2012.

With reference to fig. 18, let us describe the processes in general. The pre-processing step involves tokenization, parsing and reordering, in this order. However, before being tokenized and parsed, the input data is prepared as follows:

1. Placeholders in input are masked.

Placeholders are XML entities present in the text because all Autodesk translation and documentation data are XML objects or software strings. A segment with a placeholder looks like this:

“The {646} Density {647} and {648} Mass {649} parameters for a physical material are tied together through the volume of the object” (PhysX 3, seg14)

2. Lines with internal dots are split.
3. Segments that contain more than 50 words or 10 commas are commented to prevent the parser from processing them since they are too complex.

This pre-processed data is then sent to whichever of the two parsers is called.

After parsing, it is post-processed:

1. Placeholders are unmasked.
2. Split lines are put back together.
3. Skipped segments are commented in order not to be reordered.

After the input has been parsed and put back together, it is ready to be reordered by the applicable script. Translation (or decoding) is carried out as described in section 3.2.2. The following three sub-sections explain the configurations and the reordering rules implemented by the scripts.

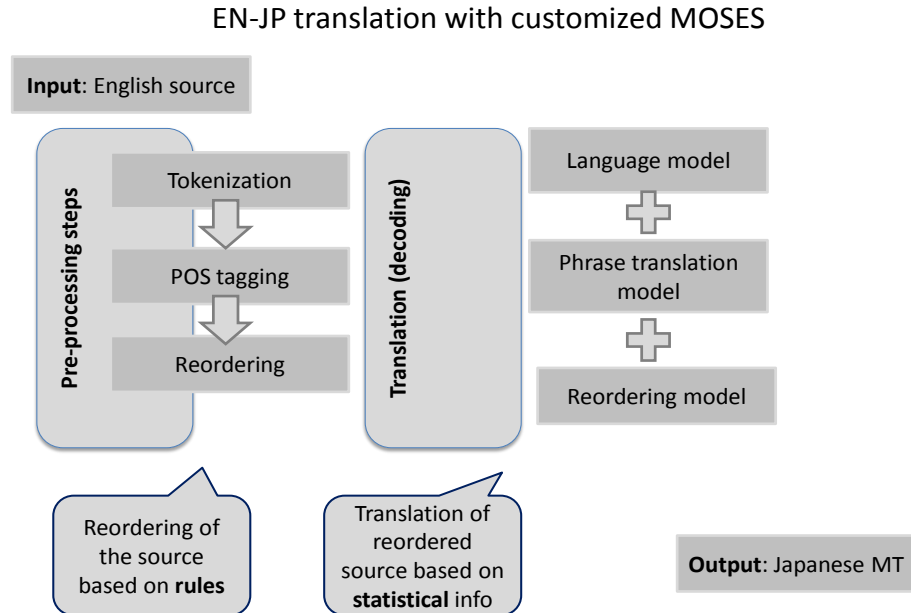


Figure 18 - EN-JP translation with Moses using reordering rules

3.2.3.1 Baseline configuration (NRO)

As already pointed out, there is no pre-processing of the English source before translation with this configuration. The input is only divided into phrases for translation with the phrase-based translation model. As no reordering is performed, the baseline configuration is called “NRO”. NRO has yielded the “worst” results as far as translation quality is concerned. Details of the evaluation that was carried out to determine MT quality are presented in chapter 5, section 5.1.

3.2.3.2 Stanford configuration (STANF)

There is a pre-processing step applied to English source before translation. The input, for example “use the command line” (a segment that comes from our data), is tokenized and then parsed by the Stanford Parser to give head-phrase information. The result is an annotated tree with a root, a sentence, a verbal phrase that contains a verb and a noun phrase made up of a determiner and nouns (fig.19). Then, the reordering rules (i.e. script 1) manipulate the tree structure of the parsed segment by modifying the children’s position in the nodes. For example, if the partial tree structure is verbal phrase followed by

noun phrase, the script 1 will reverse the order of the verb phrase and noun phrase (figures 20 and 21):

```
(ROOT (S=H (VP=H (VB=H use) (NP (DT the) (NN command) (NN=H line))))))
```

Figure 19 - Example of a segment parsed with the Stanford Parser

```
(ROOT=H (S=H (VP=H ... ) (NP ... )))
```

Figure 20 – Example of a partial tree structure before reordering (STANF)

```
(ROOT=H (S=H (NP ... ) (VP=H ... )))
```

Figure 21 - Example of a partial tree structure after reordering (STANF)

In general, script 1 moves phrasal heads to the right (although the necessary exceptions were made), based on the assumption that Japanese word order is generally the opposite of English, with verbs at the end and prepositions and modifiers after the element they modify. This configuration has yielded “medium” translation quality. Detailed results of the MT evaluation are presented in section 5.1.

3.2.3.3 Open NLP configuration (NLP)

There is a pre-processing step applied to English source before translation. The input (the same segment as above) is tokenized and then parsed for POS tagging by the Open NLP Toolkit. The result is an annotated tree with a top: a noun phrase that contains noun phrases with determiner and nouns (fig. 22).

```
(TOP (NP (NP (NN use)) (NP (DT the) (NN command) (NN line))))
```

Figure 22 - Example of a segment parsed with the OpenNLP Parser

Then, the reordering rules (i.e. script 2) manipulate the tree structure of the parsed segment by modifying the position of the POS-tagged elements. For example, if the tree of the segment “In addition, you will learn how to change view mode” has a main clause and a subordinate which contain prepositional phrases, noun phrases, verb phrases and so on (see fig. 23), following the steps described below, script 2 will change the segment to “addition in, you view mode change to how learn will” (fig. 24):

```
(TOP (S (PP (IN in) (NP (NN addition))) (, ,) (NP (PRP you)) (VP (MD
will) (VP (VB learn) (SBAR (WHADVP (WRB how)) (S (VP (TO to) (VP (VB
change) (NP (NN view) (NN mode))))))))) (, .)))
```

Figure 23 - Example of a tree structure before reordering (NLP)

```
addition in , you view mode change to how learn will .
```

Figure 24 - Example of a segment after reordering (NLP)

We describe the general functioning and steps of script 2: assuming there are only three “positions” for each sentence, 1 = beginning, 0 = middle and -1 = end, the script assigns values -1, 0 and 1 to linguistic elements (identified by the POS tags and tree information) in order to move them around. The flowchart of this process is presented below as well as in Appendix A.

STEP 1: Elements to be moved to the end and the beginning are identified. Elements to be skipped and processed separately because of their importance and/or complexity are identified as well.

STEP 2: The three skipped elements are processed: one type is moved to the end, another is reordered according to precise instructions, and the last is divided into two possible specific cases, each calling for a different treatment.

STEP 3: The specific cases are treated; some are assigned a -1 value, and some a 1 value.

STEP 4: At this point, there is a loop with an end condition: if up to now a leaf has been reached, then the components are reordered following step 6. If not, they go through step 5.

STEP 5: Sentences (that are not a leaf) are checked to see if they contain conjunctions such as “and” or “or”. If not, they are sent back to step 1. If so, they are split before being sent back to step 1 for treatment. At the end, they are reconstructed (the conjunction is left out during processing).

STEP 6: Output is produced: -1, 0, 1.

The underlying assumptions about the Japanese language are, of course, the same, but what differentiates script 2 from script 1 is that a linguistic analysis was carried out to identify specific linguistic phenomena problematic to the English to Japanese MT of Autodesk documentation. These linguistic phenomena include: (a) elements to move to the

beginning (subordinate sentences, verbal phrases with an infinitive to, prepositional phrases introduced by “in” or “to”, wh-determiners), (b) elements to move to the end (verbs, modal verbs, conjunctions, the adverb not) and (c) elements that have to be treated separately (past participle in verbal phrases, present participles in verbal phrases, past tenses in verbal phrases). This configuration has yielded the highest translation quality of the three candidates. Detailed results of the MT evaluation are presented in section 5.1.

To sum up, the following figure (25) illustrates the translation process with the three configurations:

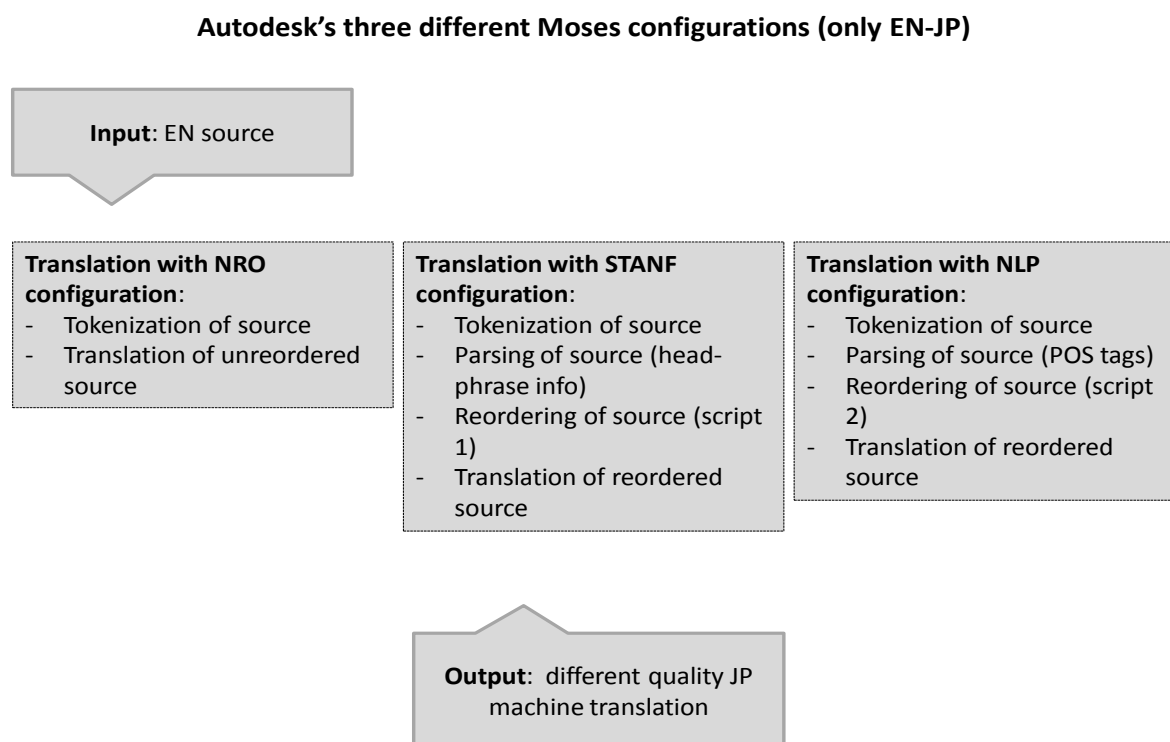


Figure 25 - Autodesk's three different Moses configurations for EN-JP translation

To see translation examples of all configurations and text types, refer to Appendix B.

3.3 The Productivity Tests

With regard to the integration of computer assisted translation (CAT) tools, Autodesk is a pioneer and has been using translation memories and content management systems for a number of years. It was also one of the first medium-to-large companies to integrate MT in

its localization workflow, in 2001 (Schenker, 2001), and to manage it internally. Back then, the MT system was Systran, the target languages were French and Spanish, and there were plans to add German, Italian and Asian languages. At present, the MT system is Moses and all segments that have a TM match lower than 75% are sent to Moses for pre-translation and then presented as such to the external LSPs for post-editing. This process has been in place for about three years for ten languages: the FIGS plus Brazilian Portuguese, Chinese (Simplified and Traditional), Czech, Polish and Russian. Productivity gains are the main reasons for this implementation: studies conducted by Autodesk proved that in its case, post-editing raw MT output is faster than translating from scratch, thus helping the company meet its tight budget and, above all, time constraints imposed by the publishing cycle (mentioned in section 3.1) (Thicke, 2011).

However, MT is not the type of solution that can be once adopted and then left to itself. PE productivity, in particular, has to be constantly measured, and translation quality monitored. This is particularly true with statistical MT systems that are trained on a given set of data and then perform sometimes worse on new (although similar) content. The localization MT team needs information to orient its work and respond to the questions and expectations of LSPs and their translators who deal with MT output. Given the lack of publicly available and comparable data on productivity, it gathered its own, with productivity tests in 2009, 2010 and 2011 (see 3.3.2 for results). Thanks to these productivity tests, Autodesk was able to compare the data gathered over the years for the same languages and monitor productivity. For new target languages, it could determine the existence and extent of productivity gains and make informed decisions about implementation in production.

The productivity tests were carried out under controlled conditions, in a real setting, on a dedicated post-editing workbench described in the next sub-section.

3.3.1 The Machine Translation Post-Editing Workbench

Translators were asked to perform bilingual post-editing on a dedicated online platform called the Machine Translation Post-Editing Workbench, for short: PE workbench²³ (see fig.26). The workbench had been the same since 2009, but we will describe it with reference to the 2011 test.

²³ The description is based on the 2011 version this work is dealing with. Previous years' workbenches were overall the same but for differences refer to Plitt and Masselot (2010).

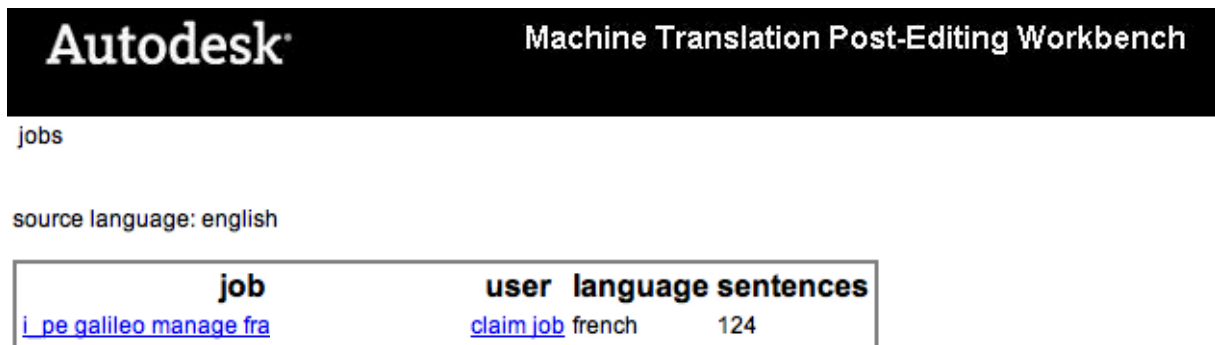


Figure 26 - The Machine Translation Post-Editing Workbench's main page

The PE workbench is where both translation and post-editing jobs were performed. For translation jobs, the “MT” field was empty, whereas in the post-editing jobs it was pre-populated with a MT proposal. Translators could click on a job name to have a look at the whole job (all segments), as shown in the below figures (27 and 28).

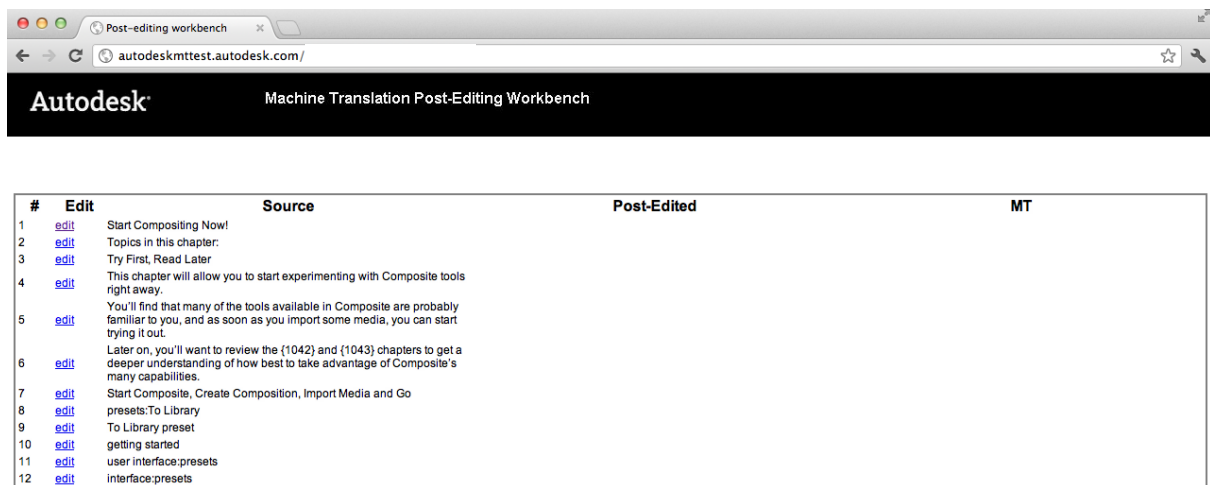
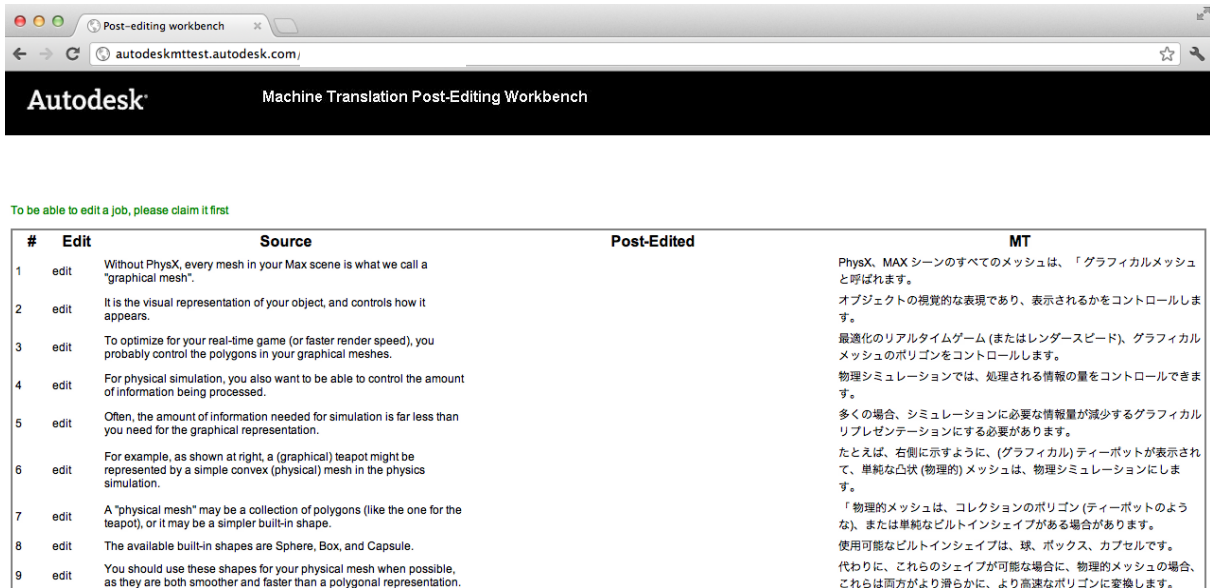


Figure 27 - PE workbench: a translation job overview

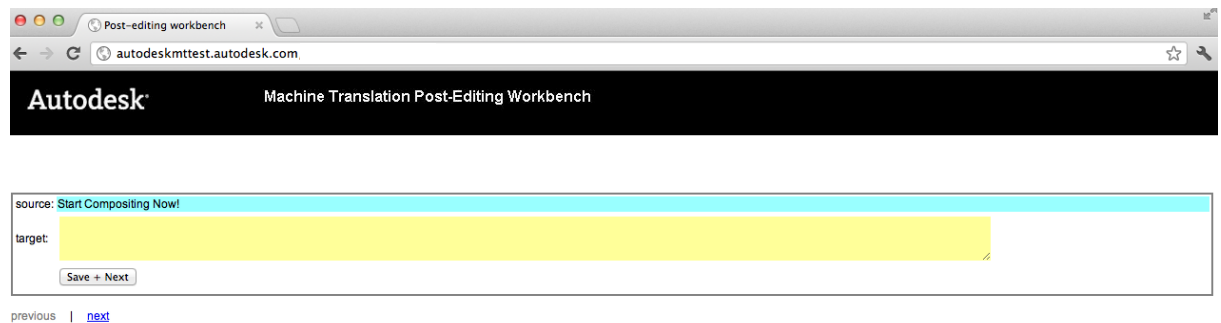


To be able to edit a job, please claim it first

#	Edit	Source	Post-Edited	MT
1	edit	Without PhysX, every mesh in your Max scene is what we call a "graphical mesh".		PhysX, MAX シーンのすべてのメッシュは、「グラフィカルメッシュ」と呼ばれます。
2	edit	It is the visual representation of your object, and controls how it appears.		オブジェクトの視覚的な表現であり、表示されるかをコントロールします。
3	edit	To optimize for your real-time game (or faster render speed), you probably control the polygons in your graphical meshes.		最適化のリアルタイムゲーム (またはレンダースピード)、グラフィカルメッシュのポリゴンコントロールします。
4	edit	For physical simulation, you also want to be able to control the amount of information being processed.		物理シミュレーションでは、処理される情報の量をコントロールできます。
5	edit	Often, the amount of information needed for simulation is far less than you need for the graphical representation.		多くの場合、シミュレーションに必要な情報量が減少するグラフィカルリプレゼンテーションに必要があります。
6	edit	For example, as shown at right, a (graphical) teapot might be represented by a simple convex (physical) mesh in the physics simulation.		たとえば、右側に示すように、(グラフィカル)ティーポットが表示されて、単純な凸状 (物理的) メッシュは、物理シミュレーションにします。
7	edit	A "physical mesh" may be a collection of polygons (like the one for the teapot), or it may be a simpler built-in shape.		「物理的メッシュ」は、コレクションのポリゴン (ティーポットのような)、または単純なビルトインシェイプがある場合があります。
8	edit	The available built-in shapes are Sphere, Box, and Capsule.		使用可能なビルトインシェイプは、球、ボックス、カプセルです。
9	edit	You should use these shapes for your physical mesh when possible, as they are both smoother and faster than a polygonal representation.		代わりに、これらのシェイプが可能な場合に、物理的メッシュの場合、これらは両方がより滑らかに、より高速なポリゴンに変換します。

Figure 28 - PE workbench: a post-editing job overview

To start working and thus recording the time, the translators had to claim the job and click on the “Edit” button at the left of the source segment. Only once they clicked on “Edit” were they presented with one segment at a time, their working space being made of a browser window split in two with a “Source” field and pre-populated/empty “MT” field, as shown below (fig.29 and 30).



Autodesk Machine Translation Post-Editing Workbench

source: [Start Compositing Now!](#)

target:

[Save + Next](#)

[previous](#) | [next](#)

Figure 29 - PE workbench: segment to translate

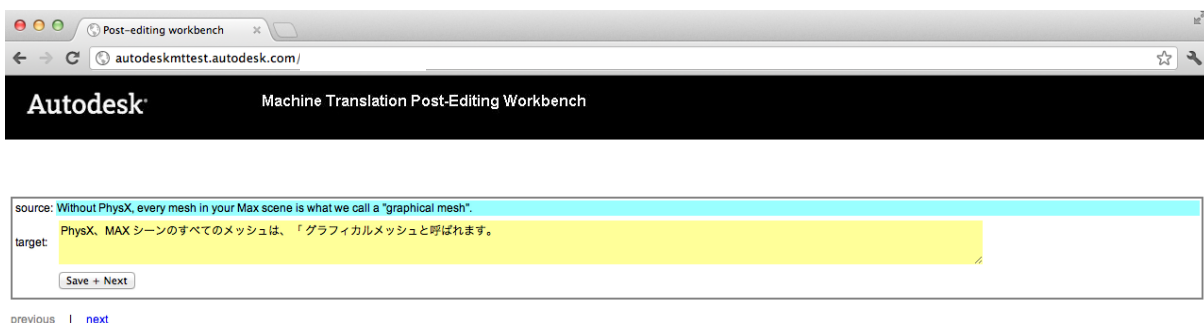


Figure 30 - PE workbench: segment to post-edit

The instructions given to translators were: *“To make this exercise successful, don't try to be particularly fast or slow - just work as usual. The recommendation is to handle this work as if it was normal production work. The end result is supposed to be entirely free of translations mistakes, typos, badly positioned [...] placeholders etc.”*²⁴

Moreover, translators could take breaks and come back to their work to make changes. If they spent more than 5 minutes on one segment, a message would appear asking them to either discard the time recording in case they had not been working on the segment (e.g. they had to answer a phone call, etc.) or to save the recorded time because they were actually working on that segment. This was to prevent recording unannounced breaks. Also, since it was important to gather enough data on both translation and post-editing, translators were instructed to follow the given job order – but there was no way to force them to do so. As to the content of the jobs, the actual text order was respected, to preserve context.

The test in which our data set was post-edited took place over two days in August 2011 and translators were asked to complete sixteen hours of translating/post-editing.

3.3.2 Results (2009-2011)

In the 2009 test, twelve translators participated and translated into French, Italian, German and Spanish. Moses was trained on translation data up to 2008 and the pre-translated segments came from 2009 documents. This is a typical situation with a statistical machine translation system in use in a company with yearly cycles. Translation jobs gave a benchmark time for translation (individual and across the groups), and post-editing jobs did

²⁴ Autodesk, The Machine Translation Post-Editing Workbench, online website, access restricted, last visited in April 2012.

the same for post-editing. The translation benchmark provided a reference as to an individual translator's working pace.

The findings were that:

- Productivity gains were achieved in every language, for all translators, but individual variance was high.
- It was slower translators who benefited most of MT (they achieved greater productivity increases).
- Sentences of about 25 and 22 words were optimal for translation and post-editing respectively.
- No particular language or content domain was more suitable than others for MT and PE.
- MT evened out the work pace across translators.
- Automated metrics were used to track changes that occurred in post-editing and were found to be relatively consistent.
- Sample QA of processed segments resulted in translation jobs having more mistakes than post-editing jobs.
- Sample QA of processed segments indicated that all jobs were of publishable quality.
- Translators were not always able to self-assess their performance.

The same approach was followed for the 2010 and 2011 tests. This approach has several advantages, namely its empirical methodology can be reused, there are benchmark times for translation and post-editing allowing for valid comparisons, and translators' differences in work pace are taken into account instead of assuming everyone has a daily throughput of 2500 words. As stated, the 2009 productivity test covered only FIGS and yielded positive results as productivity gains for all languages and translators were observed, but this was not the case in the 2010 test that covered new target languages. Among those was Japanese, for which a loss of productivity was observed²⁵.

²⁵ At the time of my internship, the company was aiming to eventually add languages to the set of those machine translated and, if possible, to improve the overall post-editing experience. Japanese was on the list of the target languages to add.

The findings of the 2011 test described on the website also showed positive results and confirmed the trend that post-editing is more productive than translating from scratch for (sorted according to ascending productivity gains) Chinese, Japanese, Polish, Portuguese, German, Italian, Korean, Spanish and French²⁶.

Therefore, overall, these results are in line with the 2009 results. Namely, for the most important aspects:

- Productivity gains were achieved in every language, for all translators, but variance among languages and individuals was high.
- Sentences of about 21 and 25 words were optimal for translation and post-editing respectively.
- Sample QA of processed segments highlighted that there was no loss of quality in final output.

This chapter has provided insights into Autodesk's localization workflow, with particular focus on the implementation of machine translation and post-editing. The statistical MT tool Moses for the English to Japanese language pair was described, including the different reordering rules that characterize the configurations we are comparing in this work. Then, the productivity tests carried out to date were summarized, including the online platform used to gather data (the PE workbench) and the results. Our data set comes from the above mentioned Moses configurations and was post-edited in the PE workbench, described in the present chapter. The following chapter gives details about the data set. It also presents the metrics used for the analyses of this work and explains how they were carried out in order to obtain the desired results.

²⁶ Machine Translation at Autodesk. <http://translate.autodesk.com/productivity.html> last visited in May 2012.

4 – The analyses: methodology and metrics

This chapter focuses on the analyses that were carried out. The aim of our work is to (a) gather information on PE effort and (b) put it into relationship with (1) MT quality and (2) productivity figures. We present successively:

- The data set we worked on: the two text types submitted for translation and some statistics on the number of segments (section 4.1)
- The metrics designed to measure (a) PE effort, (1) MT quality and (2) PE productivity (sections 4.4, 4.2 and 4.3 respectively)
- The execution of the analyses (section 4.5)

Once we have carried out these analyses, we hope that the results will allow us to draw the following conclusions:

- Post-editing patterns, revealed by PEAs, shed light on strengths and weaknesses of the three configurations and, at the same time, on translators' behavior in PE; therefore we can determine which configuration needs improvements, for example in reordering, or lexical choice accuracy; secondly, we can determine whether translators need more precise PE guidelines; thirdly we could for example suggest avoiding using machine translation for certain text types.
- Correlation of PE effort and MT quality shows whether main errors in raw MT identified within the MT quality evaluation were actually predominantly object of post-editing; it also highlights the "impact" of MT quality on the type of edits translators performed. At the same time, conversely, recurrent PE edits shed light on the MT quality evaluation from a different perspective. This makes it possible to understand the limits of both the MT quality evaluation and the productivity measurement, which only gives us numbers. In the future, based on this information, we can better tailor evaluations and know what to expect from their results.
- Correlation of PE effort and productivity figures gives us a global perspective on the two sides of the same activity, which is post-editing. It adds a little knowledge about the time it takes to perform lexical, structural or stylistic PEAs. This, in turn, could be useful to decide whether to refine PE guidelines or quality of the raw MT output.

We summarized our expectations regarding the results and the conclusions we could draw. The actual conclusions are presented in part 6, but section 4.1 describes the data set.

4.1 Data set

The data that was analyzed for this work comes from Autodesk's 2011 productivity test described in 3.3.

There were 8 jobs²⁷ : 6 post editing and 2 translation, but we will focus on the post-editing ones only. They are tut2a, tut2b, tut3, physx1, physx2 and physx3. These "job names" correspond to two different text types, the characteristics of which are presented below:

Text type 1 (tutorials): tut2a, tut2b and tut3 are all tutorials of the web-based version of AutoCAD, one of Autodesk's best-known software packages for computer-assisted design. The tutorials explain to new users of AutoCAD WS (a free web-based/mobile client for AutoCAD) its basic functionalities and the main differences between it and the "traditional" AutoCAD. Their content is not new, especially for the first two, since they repeat instructions that previous Autodesk tutorials already contained. "Matches" in the corpus were thus expected to be relatively high. All of them have been checked with Acrocheck to ensure consistency and compliance with basic Autodesk technical writing guidelines. It should however be stressed that syntax and phraseology of the third tutorial are substantially different from the first two. Therefore, from a MT quality perspective, we expected slightly better results for the first two, and from the point of view of post-editing difficulty, we thought translators who have experience with Autodesk material should be highly efficient, at least in comparison to working on the "PhysX" documents.

Text type 2 (physx): The PhysX jobs (physx1, physx2 and physx3) belong to a plug-in that had not been localized yet and that has highly technical terminology combined with an unusually casual writing style. The fact that it is a plug-in to existing Autodesk software called Maya, which by contrast had already been localized, could mean some of the terminology related to that software will not pose any problem to MT; but its syntax is so different from the other "main" Autodesk products that it was expected to be challenging in that respect. For

27 The number of jobs of the actual productivity test was higher, but only completed and comparable jobs have been taken into account.

this very same reason, it could be considered an interesting indicator of Moses' performance (namely, reaction to new words and new chunks of sentences).

To build the data set, each text type was equally divided into comparable size tasks (a similar number of segments, about 150²⁸) that were in turn machine translated by the NRO, NLP and STANF configurations. The aim of this division was to gather enough and comparable data on every configuration, as shown in the table 1 below: "configuration total" shows that each configuration translated about 150 segments, about half of which were from one text type, and the other half from the other.

Table 1 - Data: total PE jobs and segments

Configuration	Job	Segments	Configuration total
NRO	Tut3	87	155
NRO	<i>Physx3</i>	68	
STANF	Tut2b	68	145
STANF	<i>Physx2</i>	77	
NLP	Tut2a	66	148
NLP	<i>Physx1</i>	82	
	Text type 1	221	448 grand total
	<i>Text type 2</i>	227	

448 segments post-edited by 4 translators would mean 1792 segments. However, not all translators finished working on every job: there were only 1654 segments. Moreover, for statistical reasons, we filtered out outliers²⁹ to even out the data. This lowers the number of segments to 1301 segments, which constitute the basis of the analyses. We will refer to these 1301 segments as "processed segments" and illustrate them in table 2 below. In the table headline, "Processed Segments (translator number)" indicates the processed segments for translator 1, 2, 3, divided into job (text type 1 or 2) and configurations (NLP, NRO or STANF). Inside the "Processed Segments (translator number)" column, "0/total number" means the job was skipped as a whole by a particular translator.

Table 2 - Data: processed segments

Configuration	Job	Processed Segments (1)	Configuration total
NRO	Tut3	73/87	73
NRO	<i>Physx3</i>	0/68	

²⁸The average segment length (in words) was: 9,9 in tut2a, 10,6 in tut2b, 11,5 in tut3, 11,3 in physx1, 13,7 in physx2 and 14,8 in physx3.

²⁹Outliers are minimum and maximum values of editing time (automatically recorded in milliseconds by the PE workbench).

STANF	Tut2b	60/68	60
STANF	<i>Physx2</i>	0/77	
NLP	Tut2a	60/66	124
NLP	<i>Physx1</i>	64/82	
	Text type 1	193	257 grand total
	<i>Text type 2</i>	64	
Configuration	Job	Processed Segments (2)	Configuration total
NRO	Tut3	79/87	89
NRO	<i>Physx3</i>	10/68	
STANF	Tut2b	61/68	132
STANF	<i>Physx2</i>	71/77	
NLP	Tut2a	63/66	63
NLP	<i>Physx1</i>	0/82	
	Text type 1	203	284 grand total
	<i>Text type 2</i>	81	
Configuration	Job	Processed Segments (3)	Configuration total
NRO	Tut3	84/87	148
NRO	<i>Physx3</i>	64/68	
STANF	Tut2b	64/68	136
STANF	<i>Physx2</i>	72/77	
NLP	Tut2a	64/66	136
NLP	<i>Physx1</i>	72/82	
	Text type 1	212	420 grand total
	<i>Text type 2</i>	208	
Configuration	Job	Processed Segments (4)	Configuration total
NRO	Tut3	78/87	78
NRO	<i>Physx3</i>	0/68	
STANF	Tut2b	63/68	131
STANF	<i>Physx2</i>	68/77	
NLP	Tut2a	61/66	131
NLP	<i>Physx1</i>	70/82	
	Text type 1	202	340 grand total
	<i>Text type 2</i>	138	
Processed segments →			1301

In conclusion, text type 1 yields a comparable basis of 708 segments i.e. 177 *same* segments were processed by all 4 translators: 55 segments of tut2a, 55 of tut2b and 67 of tut3.

Regarding text type 2, since physx1 and physx2 were processed only by 2 translators (3 and 4) and there are only 10 segments of physx 3 that were processed by 2 translators (one of whom did only physx 2 but not physx1!), comparisons must be made with due care.

To compare translators over the same jobs, it looks advisable to treat translator 1 and translator 2 as one subgroup and translator 3 and translator 4 as a different one. Finally, concerning Moses' configuration comparisons, there is considerably less data for the baseline (only 1 translator worked on both NRO jobs *and* STANF and NLP jobs) while data for the customized configurations is based on three translators' work.

4.2 MT Quality evaluation methodology and metric

In this section, we clarify the metric that was applied to measure MT quality in order to obtain a ranking of the three configurations on English to Japanese translation presented in section 3.2.3. of the previous chapter. The raw MT output of the data set was evaluated by calculating penalties on mistakes according to an error classification created *ad hoc*. The evaluation methodology followed the EAGLES 7 steps recipe and is presented below.

1. Why is the evaluation being done?

What is being evaluated is not a system but only its raw MT output, from a linguistic quality perspective. The aim of the evaluations is to highlight the number and type of raw MT errors in general, and to be able to obtain a ranking of the three configurations of the system that were involved in the translation process. The data set evaluated is the one described in 4.1; all 448 segments were evaluated, regardless of specific parts which translators did not post-edit.

2. Elaborate a task model

There were few agents: one computational linguist was providing the source and translated texts; the evaluation was performed by the author after been adapted from a previous one approved by the manager and a Japanese mother tongue colleague.

3. Define top-level quality characteristics

There are six top-level characteristics in software quality evaluations: functionality, reliability, usability, efficiency, maintainability and portability. These were designed to evaluate systems; since we limit our evaluation to the output of the system, we will rely on the relevant characteristic, namely **functionality**. Its sub-characteristics are **accuracy**,

suitability and **interoperability**. Accuracy is the ability of the system to produce the desired output, whereas suitability evaluates the pertinence of the output to the context of use. Interoperability concerns availability of the system on platforms and other technical aspects. For translation quality evaluation in our context, we considered only accuracy to be relevant. Suitability is also relevant, but in our context it would mean determining whether the output is suitable for post-editing. We deliberately left out suitability to better center the aim of the present evaluation, which is to rank three systems configurations.

Therefore, we tested **accuracy** of the raw MT translation: are the lexical choices good? Are there too many missing words? Is company terminology enforced? The detailed criteria that we tested in order to obtain a system ranking are listed in point 4.

4. Produce detailed requirements for the system under evaluation, on the basis of 2 and 3

Given that we could carry out a human/manual evaluation and that we wanted to test accuracy, we defined the following requirement or **criteria**:

- 1) **Lexical choice** criteria: are (a) company terms and (b) general words correctly retrieved from the training corpus? To be more precise: are the translations correct and how many unknown words were left in English or mistranslated?
- 2) **Word order** criteria: (c) is the order of words between them correct? (d) Is the order of multi-words correct?
- 3) **Sentence structure** criteria: (e) are the sentence structures possible structures in Japanese and is the order of principal and subordinate sentences correct?
- 4) **Added or omitted words** criteria: were there any (f) arbitrarily added or deleted words; (g) added or deleted negations?

The motivation for the above-mentioned criteria is explained below.

When assessing translation accuracy, one of the key elements is the *lexical choice*. We separated *company terminology* from *general language words* and also introduced a separated criterion for *word order*. The presence of criteria for both a lexical choice and word order for terms is motivated by practical observations: for example, a multiword might have been translated well, but in the wrong order. To be able to account for these phenomena, we needed two distinct metrics.

Impossible constructions (pertaining to the *sentence structure* criteria) are to be monitored because they show the system's behavior towards totally new content. Moreover, the structure of sentences must be grammatically correct and this means checking the correct positioning of principal and subordinate sentences in a clause, as the order is often inverted in English to Japanese translation.

The number of *added or omitted words* being more or less directly connected to statistical machine translation in general, it has to be monitored to observe system functionality. Last but not least, previous studies on SMT of Autodesk content revealed that one of the system's vulnerabilities lies in the risk of making close-to-perfect translations in which only the modality of the sentence is wrong, e.g. affirmative statements turn into negative statements. This is why particular care will be taken to check statement modality, and any mistake will count as 2 penalties.

5. Devise the metrics to be applied to the system for the requirements produced under 4

The metrics used are the following (there was also a color code for people not familiar with Japanese to be able to "visually understand" the evaluation at a glance) shown in table 3:

Criteria	Pertains to	MT error and description	Penalty point given
Lexical choice (a)	Terminology	Autodesk term -- An Autodesk term (i.e. product name, button name, etc.) is lexically wrong. Synonyms are also considered as errors.	-1
Lexical choice (b)	Terminology	General language word – A general language word is lexically wrong but synonyms are accepted if they convey the same	-1

		meaning.	
Word order (c)&(d)	Structure	Word order -- Words are lexically correct but misplaced. Autodesk terms order -- Autodesk terms are correct lexically but misplaced	-1
Sentence structure (e)	Structure	Sentence structure – The sentence is impossible in Japanese or the order of subordinates and main clause is wrong.	-1
Added/omitted (f)&(g)	Added/omitted	Words added or omitted/ Negation added or omitted -- Words in source were not translated or words not in source were added/ A negation was added or omitted where not appropriate	-1/-2

Table 3 - MT Quality evaluation metric

We calculated subtotals for terminology, structure and added/omitted words.

Fig. 31 below is a screenshot of some evaluated segments: the first column is the job name, the second is the English source, and the last is the MT proposal that was evaluated.

tut3	Plot Drawings	図面の印刷
tut3	Share Drawings	図面を共有
tut3	Use Timeline	を使用してタイムライン
tut3	Lesson 1: Plot Drawings	レッスン 1: 図面の印刷
tut3	In this lesson, you will learn to specify	このレッスンでは、尺度を指定するには図面を出力する。
tut3	When you plot a drawing, you either sp	図面を印刷すると、尺度を正確に指定するか、または、イメージを現在の用紙サイズを選択します。
tut3	To plot a drawing to your desktop	図面を印刷するには、デスクトップに
tut3	1 Click Open ➤ My Drawings and select	1 の [開く] \$submenu: [図面] を選択し、サンプル図面に [ファイルを選択] ダイアロ
tut3	2 Click Output ➤ Plot.	2 つの [出力] \$submenu: [印刷] をクリックします。
tut3	4 Verify that the layout in the Layout &	4 をレイアウトのレイアウトと範囲がモデルになります。
tut3	5 Under Scale select 1:50 from the Scale	5 尺度を [1:50] を選択し [スケール] ドロップダウンリストを表示します。
tut3	6 Click OK to plot the drawing.	6 [OK] をクリックして、図面を印刷します。

Figure 31 - Evaluated MT segments example

6. Design the execution of the evaluation

The evaluation was to be carried out in a short timespan (about one week), directly in Excel to allow for plotting of results.

7. Execute the evaluation

During the internship, the evaluation was carried out, results were analyzed and briefly made the object of a report; for this thesis, the evaluation was double-checked, corrected where necessary, and digitized (i.e. annotated in XML format).

4.3 PE productivity measurement

The approach to measuring PE productivity in our work is the one adopted at Autodesk, described in its paper by Plitt and Masselot (2010) summarized in section 3.3. of the previous chapter.

Given that we have no reason to change the way of calculating productivity for our work, we will proceed in the same way that Autodesk did. This means that editing time is divided by source words to determine “throughput” (or productivity). What we will do is to provide statistics regarding Japanese language that are not published on their website (as of May 2012), namely the calculation of PE productivity over the MT configurations (NLP, NRO and STANF described in chapter 3, section 3.2.3).

4.4 PE effort metric: PEAs adapted to Japanese language

We summarize Blain et al.'s post-editing actions – already presented in 2.2.2.1. -in four key points:

1. PEAs give qualitative information.
2. PEAs must be logical edits which means that they can include more than one mechanical edit (see example below table 4).
3. MT quality should be high.
4. The textual difference between raw MT and post-edited version must not be too big, to allow a smooth annotation of PEAs.

We adapted the PEAs to English – Japanese SMT in such a way that these four key points would be respected, with one exception (point 3):

1. Japanese PEAs give “qualitative” information, i.e. linguistic information about the edits. We expect from Japanese PEAs to give us information on what translators did at the post-editing stage and it is the reason why we used them, rather than automatic scores.
2. Japanese PEAs are logical edits, and at the same time they follow the “logic” of the Japanese language (see example below table 4).
3. MT quality is **not** “high”.
4. As a consequence of MT quality (point 3) we expected a significant amount of textual difference between raw MT and post-edited versions. Therefore, we designed one PEA to tag segments that had been completely re-edited and that were therefore not suitable for PEA annotation³⁰.

To be more precise, the Japanese PEAs were designed in the following way (see also table 4):

1. Qualitative information is gathered on two levels, PEA categories and their sub-group, types, which all refer to linguistic features: we introduced four categories, partly inspired on Blain et al.'s work, partly motivated from facts such as recurrent MT errors: lexical PEAs, grammatical PEAs, structural PEAs and stylistic PEAs. Lexical PEAs are changes at the level of words, for example nouns, adjectives and verbs. Here, by “adjectives”, we meant

³⁰Our work, contrarily to Blain et al.'s, was manual, but even so annotation is feasible only if textual difference is not too high.

“modifiers” and included adverbs; for nouns, we introduced a distinction between Autodesk terms and general words following the MT quality evaluation (section 4.2), but the present work lacks a scientific approach for terminology. Grammatical PEAs are category changes (from a noun to a verb or vice versa), verb tense changes (from accomplished to non-accomplished or vice versa), verb mode changes (from indicative to imperative or the like) and preposition changes. Verb tense and verb mode are particularly important as they have an influence on the structure of the sentence as well (certain verb modes can serve as a connector between sentences, and certain verb tenses can greatly affect the relationship between a subordinate and a principal sentence because an accomplished tense in front of a noun “creates” a relative subordinate whereas an unaccomplished tense does not). There were almost no preposition change annotations in our data since this PEA turned out to be too specific for the MT quality at hand. Structural PEAs are local-level and clause-level reordering by the translators. Local reordering means the translator swapped the order of words or chunks while clause reordering means words or chunks were moved from one position of the sentence to its very end or opposite, more than one unit (word or chunk) away. Stylistic PEAs are stylistic changes in lexical choices or phrasing. Specially, the former were used when a synonym was chosen for a correct MT, but as we privileged obtaining detailed information on PEAs, we preferred to avoid annotating data with stylistic PEAs, whenever possible³¹. Last but not least, there was a “miscellaneous” PEA for unclassifiable PEAs and a “local change” PEA that was used to annotate added or deleted words/chunks. This last PEA did not belong to any category as, depending on the context, it can be lexical, structural or grammatical. We chose to introduce it as a stand-alone PEA to reflect one of statistical machine translation’s characteristics (added or missing words), which was measured in our MT quality evaluation.

2. During analysis, structural PEAs had the priority over other PEAs. This meant that if there was a lexical choice change in a reordered chunk, only reordering

³¹Another reason for this is that a translation quality assessment had already been performed on sample post-edited segments by Autodesk’s relevant team, and indicating “stylistic changes” could be interpreted misleadingly as a critic to the way translators/post-editors worked, which is of course not the case.

was annotated.

3. A PEA named “clause change” was introduced to annotate post-edited segments that were too different from MT proposal; these were then filtered out as appropriate for results analysis. This PEA does not belong to any category.
4. As explained in 2.2.2.1., PEAs are a qualitative evaluation method that applies to “high-quality MT”. We are aware that our level of quality was not always “high” (even though there may be different ways of defining “high”) but we decided to use PEAs for the other advantages this analysis method had to offer. To follow Blain et al.’s intended use of PEAs, on one hand we applied filters to exclude “bad quality” MT segments (leveraging the available MT quality evaluation with its scores). On the other hand, we created a PEA (“clause change”, above) that signalled textual differences that are too great and also filtered them out from the final results. Moreover, a “PE error” PEA was introduced. This PEA also does not belong to any category and was used to annotate mistakes by translators.

The PEA categories and types are summarized in table 4 below.

Table 4 - Post-editing actions adapted to Japanese

PEA category	PEA type	Raw MT output	Action performed by translator
<i>Belonging to one of the 4 categories</i>			
Lexical	<i>Noun change (Autodesk term)</i>	An Autodesk term (i.e. product name, button name, etc.) is lexically wrong	An Autodesk term is changed.
Lexical	<i>Noun change (general) Adjective change Verb change</i>	General language word is lexically wrong	A noun, adjective or verb is changed.
Grammatical	<i>Verb tense change Verb mode change</i>	A verb tense or mode is wrong.	Change in tense (accomplished/non accomplished) or mode

			(imperative negative/positive, etc.)
Grammatical	<i>Preposition change</i>	A preposition or conjunction or the like is wrong.	The relevant preposition is changed.
Grammatical	<i>Category change</i>	The category of the source was not recognized or it sounds strange in the target language.	Category is changed, for instance a verb is nominalized or vice versa.
Structural	<i>Local- level reordering</i>	A multiword term/word order is wrong (but lexically correct). Words or chunks are in the wrong position in the sentence.	Word order reordering (between 2 words, e.g. noun-adjective) or chunks.
Structural	<i>Clause-level reordering</i>		Words or chunks are moved around in the sentence.
Stylistic	<i>Lexical or clause-level stylistic change</i>	No particular mistake.	Lexical change to a specific lexical item or to the wording/phrasing of the sentence.
<i>Not belonging to any of the categories</i>			
--	<i>PE error</i>	--	Mistake in MT is not corrected or a mistake is introduced during PE.
--	<i>Clause change</i>	--	MT proposal is largely rewritten.
--	<i>Miscellaneous</i>	Entities and placeholders are wrong	Punctuation, entities or placeholders are corrected.
--	<i>Local change (added or deleted words)</i>	Words or chunks were added or omitted.	Words or chunks are deleted or added.

<i>Correct MT</i>	Raw MT is correct.	--
-------------------	--------------------	----

We would like to note that unlike Blain et al.'s work, our analysis was not automatic.

PEA example

Source: Draw Objects

MT: 作図オブジェクト(“Drawing” (noun) “objects”). The order needs to be reversed in Japanese and if the noun is kept for source “draw”, a connector must be inserted between the two words.

PEA: オブジェクトの作図(“objects” “of” “drawing”). The translator reversed the order, kept a noun to translate the source “draw” and inserted a connector. There are two mechanical edits, but only 1 PEA/logical edit of reversing the order of nouns that are syntactically connected to each other, which entails inserting a connector.

The results of the analyses carried out by applying the above-mentioned metrics are presented in chapter 5. The execution of the analyses is described in the next section.

4.5 Execution of the analyses

As was mentioned in the introduction, the analyses are comparative, for they juxtapose three Moses configurations, and twofold: they (a) gather data, mainly on PE effort and (b) study the correlation of PE effort with MT quality and PE productivity.

To establish correlations we first have to gather data. That is what step (a) consisted of. Below, we describe, in chronological order, how we collected not only PE effort data, but also data on MT quality and PE productivity:

MT Quality evaluation had been carried out during my internship, applying the metric described in 4.2. For the present work, it was reviewed and digitized (i.e. annotated in XML format).

PE Productivity calculation had also been partially carried out at the time of my internship, but we expanded it for this work, and calculated productivity of the three different Moses configurations, in jobs and for the four translators.

PE effort measurement was the main analysis of this work. We defined the metric in 4.4 and annotated the post-edited segments for PEAs.

PEA annotation took place in a XML editor and was conducted manually. A simple document-type definition was written, to guarantee a consistent structure in the data, since it was imported from Excel files, and to enable fast tag identification. Indeed, the XML editor was chosen as it was free and input-sensitive, among other reasons. The outcome is an annotated XML file. Fig. 32 shows an example of segment 1 of job tut2a post-edited by four translators and annotated with PEAs in the XML editor:

```
<annotations>
  <segment number="1" translator="1" jobname="tut2a" config="NLP">
    <src>Tutorial 2: Draw and edit in AutoCAD WS 2011</src>
    <MT>チュートリアル 2: AutoCAD WS で作図と編集 <MTerror type="s_term">2011</MTerror></MT>
    <postedited commentedas="example">チュートリアル 2: <pea category="structural" type="local_reordering">AutoCAD WS 2011</pea> で<pea
category="grammatical" type="category_change">作図、編集する</pea></postedited>
  </segment>
  <segment number="1" translator="2" jobname="tut2a" config="NLP">
    <src>Tutorial 2: Draw and edit in AutoCAD WS 2011</src>
    <MT>チュートリアル 2: AutoCAD WS で作図と編集 <MTerror type="s_term">2011</MTerror></MT>
    <postedited>チュートリアル 2: <pea category="structural" type="local_reordering">AutoCAD WS 2011</pea> での<pea category="lexical"
type="nounchange_adsk">描画</pea>と編集</postedited>
  </segment>
  <segment number="1" translator="3" jobname="tut2a" config="NLP">
    <src>Tutorial 2: Draw and edit in AutoCAD WS 2011</src>
    <MT>チュートリアル 2: AutoCAD WS で作図と編集 <MTerror type="s_term">2011</MTerror></MT>
    <postedited commentedas="example">チュートリアル 2: <pea category="structural" type="local_reordering"> AutoCAD WS 2011</pea> <pea
category="grammatical" type="preposition_change">での</pea>作図と編集</postedited>
  </segment>
  <segment number="1" translator="4" jobname="tut2a" config="NLP">
    <src>Tutorial 2: Draw and edit in AutoCAD WS 2011</src>
    <MT>チュートリアル 2: AutoCAD WS で作図と編集 <MTerror type="s_term">2011</MTerror></MT>
    <postedited commentedas="example">チュートリアル 2: <pea category="structural" type="local_reordering"> AutoCAD WS 2011</pea> <pea
category="grammatical" type="preposition_change">での</pea>作図と編集</postedited>
  </segment>
```

Figure 32 - Annotated segment example

Once data was collected, step (b) was applied in the following way: first, we calculated “individual” results for our three elements under analysis (MT quality, PE productivity and PE effort), then we established the correlations between them:

Study of MT Quality as such, to find the best configuration

- Ranking of the three configurations
- Summary of MT errors: quantity and types
- Summary of MT errors in the processed segments

Study of PE productivity as such, to know the speed of translators in PE

- Throughput calculated on the basis of a configuration, job and translator

Study of PE effort as such, to know what kind of PE edits translators performed

- Calculation of an average of PE edits.
- Summary of recurrent PEAs for configurations and text types.
- Summary of number and nature of PE errors and rewritten segments.

Study of correlations between PE effort and MT Quality, to see the impact of linguistic quality on the number and types of PE edits

- Number of error-free segments that were post-edited
- Number of PEAs sorted by ascending MT score (best to worst)

Study of correlations between PE effort and PE productivity, to see the editing speed depending on edit type

- Post-editing productivity sorted by number of PEAs in segments.

This chapter was dedicated to the description of the data set we worked on, the explanation of the metrics used to analyze it and the execution of the analyses. The relevant sections highlighted text characteristics and expected impact on translation quality as well as the statistical relevance of our data; the linguistic metric for MT quality evaluation and system ranking was presented in a detailed way, just as the metric for annotating post-editing actions in Japanese, in order to show the links between them; the execution of the annotation was also briefly illustrated; finally, the method used to analyze the results in a meaningful way concluded this chapter, which is followed by a report of all results.

5 – Results

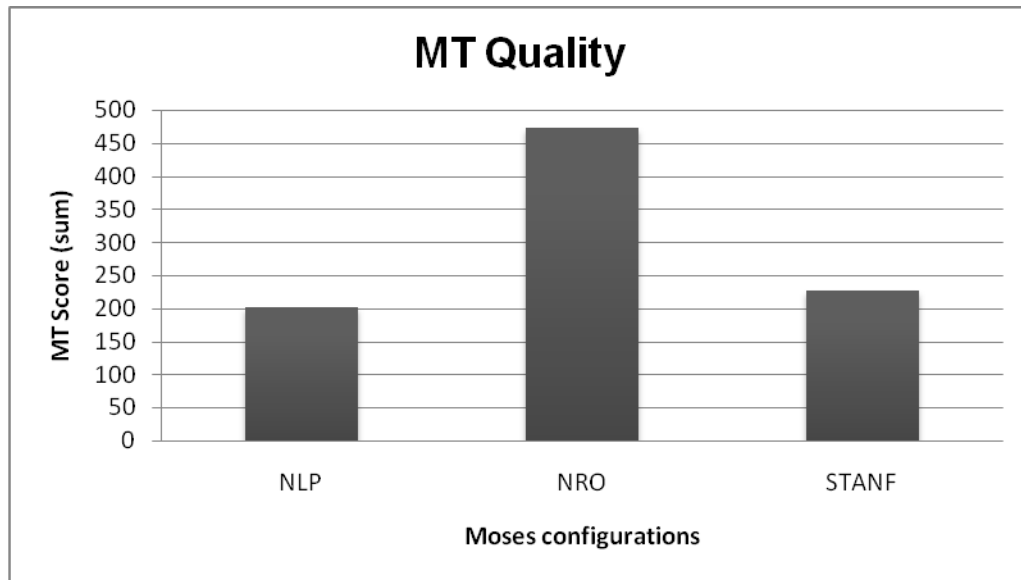
This chapter presents all the results of the analyses on MT Quality, PE productivity and PE effort of the Japanese translators on the analyzed dataset. Section 5.1 presents the MT Quality of the three Moses configurations, the types of errors and the ranking of the configurations according to these results. Section 5.2 shows the productivity of the translators for each configuration and text type. Section 5.3 presents the results of the PEA annotations that give us PE effort information. Averages and most recurrent PEAs are reported, as well as other phenomena such as completely rewritten segments and PE mistakes. Section 5.4 analyses the results further to establish a correlation between MT Quality, PE effort and PE productivity. The chapter concludes with preliminary results (in section 5.5) of the automatic scores computation of textual difference (or edit distance) between MT and PE.

5.1 MT Quality

This section presents the results of the linguistic quality evaluation carried out following the metric described in section 4.2 on the raw output of the three Moses configurations (for a description of the configurations, see section 3.2). Since the main aim of this evaluation was to establish a ranking of the three systems, the approach to present results will be predominantly configuration-based.

Scores ranged from 0 (no error) to 14. 448 segments were evaluated (about 150 per configuration).

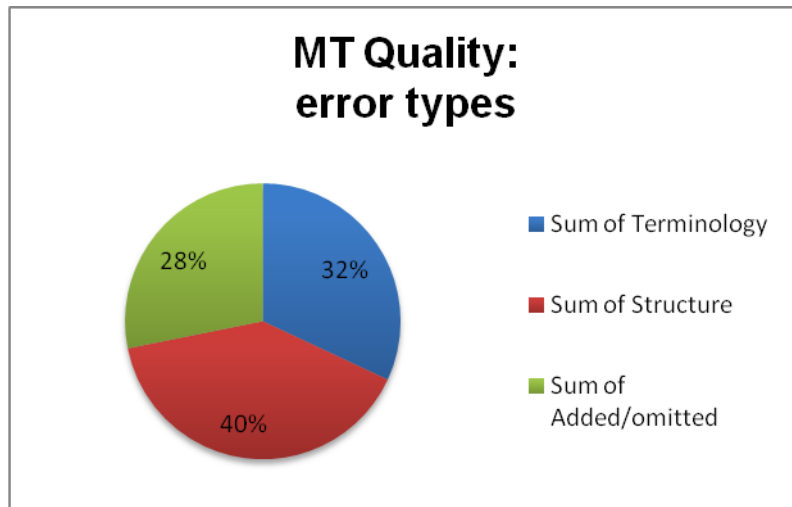
Graph 1 illustrates overall results and shows that NRO was by far the worst-performing configuration, while NLP performed slightly better than STANF:



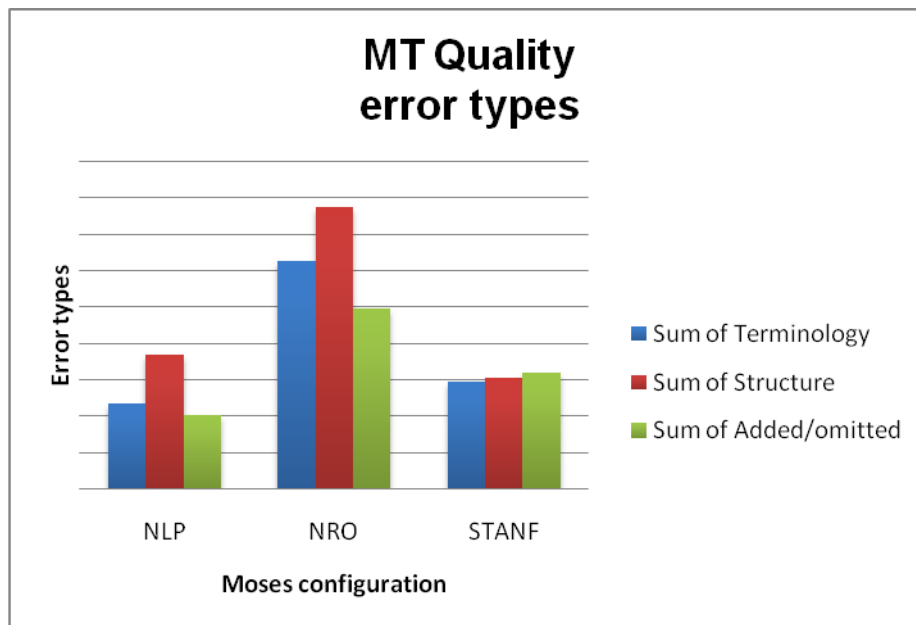
Graph 1 - MT Quality results: all Moses configurations

Consequently, the ranking of the systems is NLP highest (1), STANF medium (2), and NRO lowest (3). NLP and STANF have a very close score, 202 penalties against 227 respectively, while NRO lags far behind with 474 penalties.

Graph 2 shows the percentage of the types of errors overall and graph 3 shows the distribution of error types over the configurations. Overall, on the total penalties (903 points), 40% are structural errors, 32% are terminology errors and 28% are errors due to added or omitted words. In this regard, NLP and STANF perform differently as STANF has about the same number of structural, terminology and added/omitted words errors, while NLP performed well as regards added/omitted words and better than STANF in terminology, but worse as far as structure is concerned. Bad performance on structure by NLP was a somewhat unexpected result and calls for attention at the PEA analysis stage.

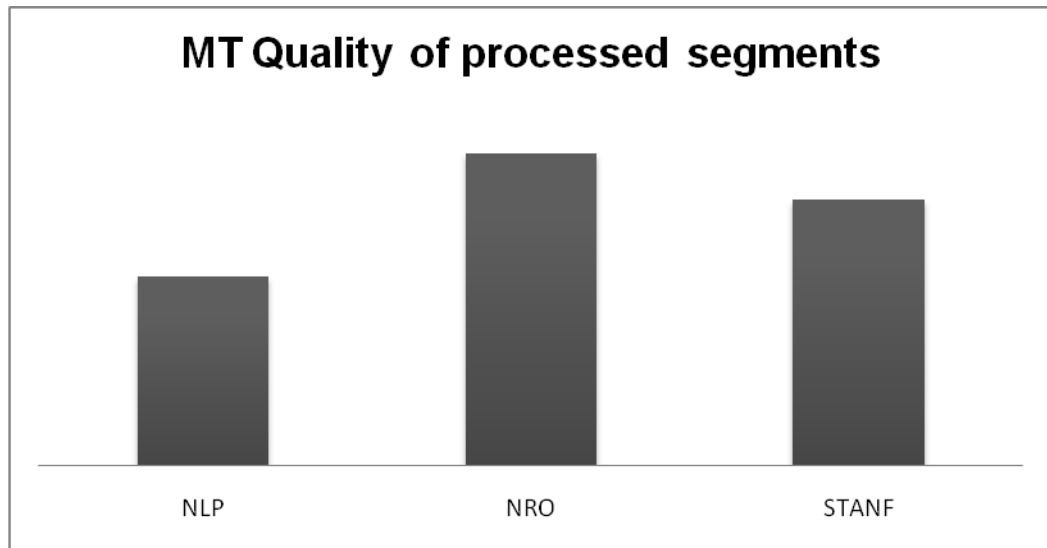


Graph 2 - MT Quality: types of errors (overall)



Graph 3 - MT Quality: types of errors (configurations)

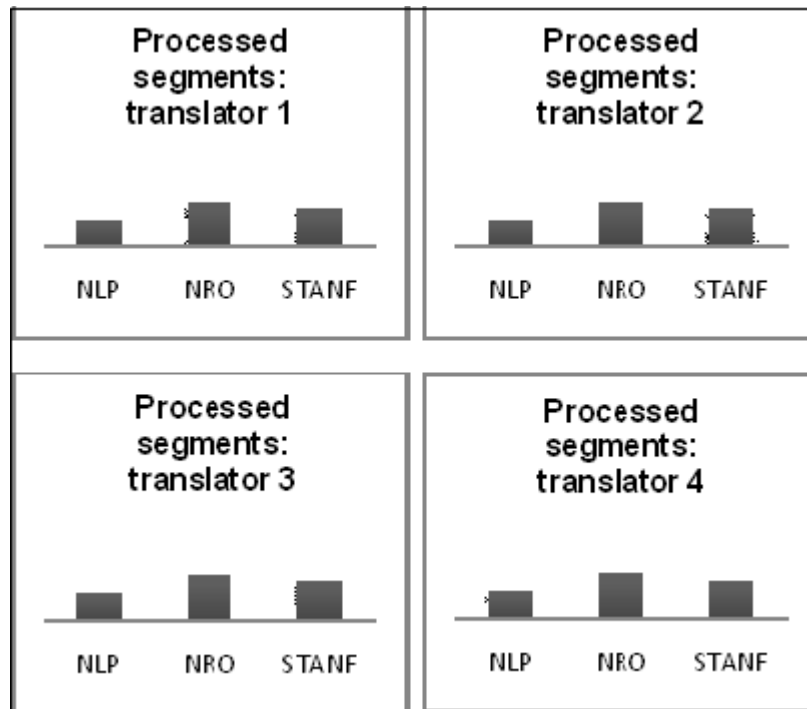
Finally, given the special care that must be paid to comparable data (see 4.1), we also calculated the results on processed segments only, filtering out all segments that translators did not work on. The result of this is shown in graph 4 and the proportions are the same as the evaluation over all segments.



Graph 4 - MT Quality: processed segments only

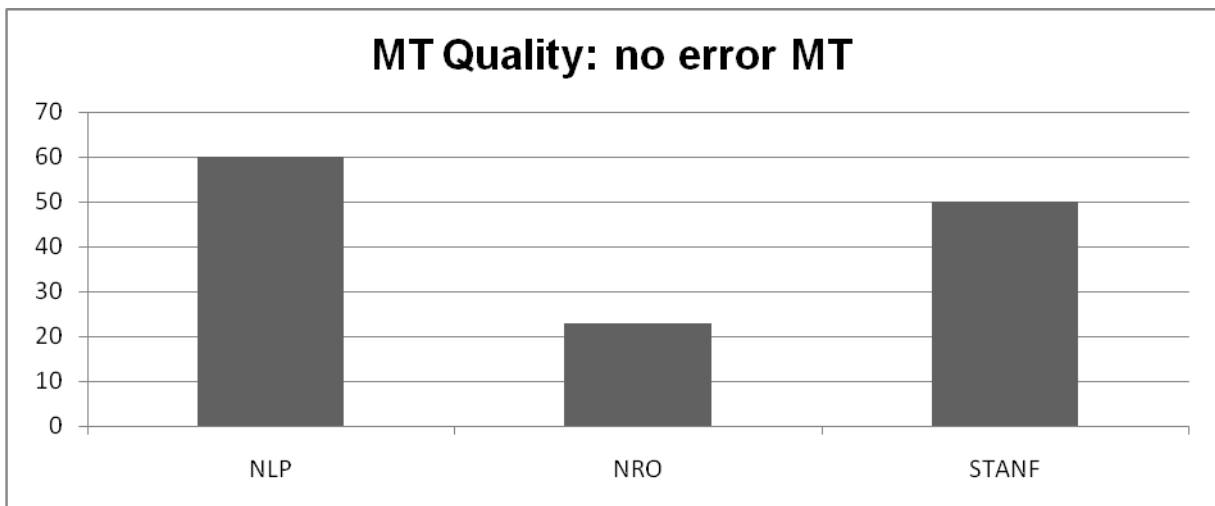
This means the translators were not faced with different quality MT just because they skipped or did not complete a job. This ensures that there is no bias and we can put MT Quality into direct correlation with PEAs.

For more details on the MT Quality each translator actually dealt with, see graph 5. Differences can be explained from the fact that processed jobs do not coincide over the translators (see section 4.1 on the data set). Despite this inequality, the proportions are in line with the evaluation on all segments.



Graph 5 - MT Quality: processed segments (translators)

Finally, we show the number of correct MT segments for each configuration as it is another type of indication about the MT Quality. Graph 6 shows the total number of error-free MT segments for each Moses configuration: 60/148 in NLP, 23/155 in NRO and 50/145 in STANF. It confirms the ranking we established.



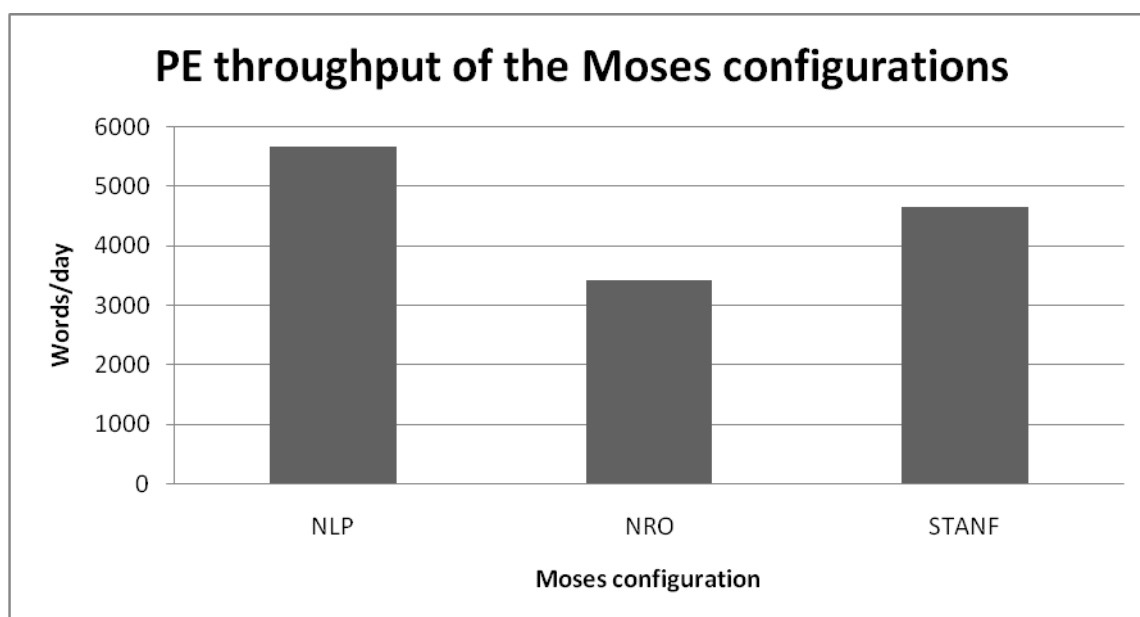
Graph 6 - MT Quality: error free segments in every configuration

As regards error-free segments among the processed segments, in total, 409 segments over 1301 were error-free: 200 by NLP, 153 by STANF and 56 by NRO, which

confirms our ranking while also giving an indication as regards the translation quality which the translators dealt with and the consequent achievable productivity.

5.2 PE Productivity

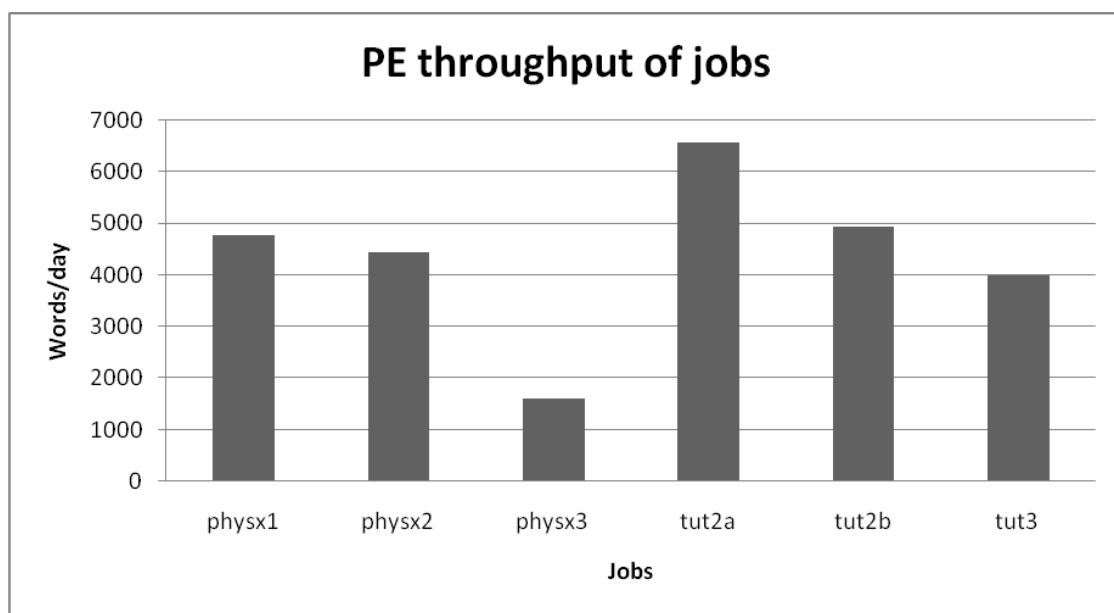
This section presents the results of the productivity calculations carried out following the method described in section 4.3 on the post-edited versions of the three Moses configurations (for a description of the configurations, see section 3.2).



Graph 7 - Productivity (configurations)

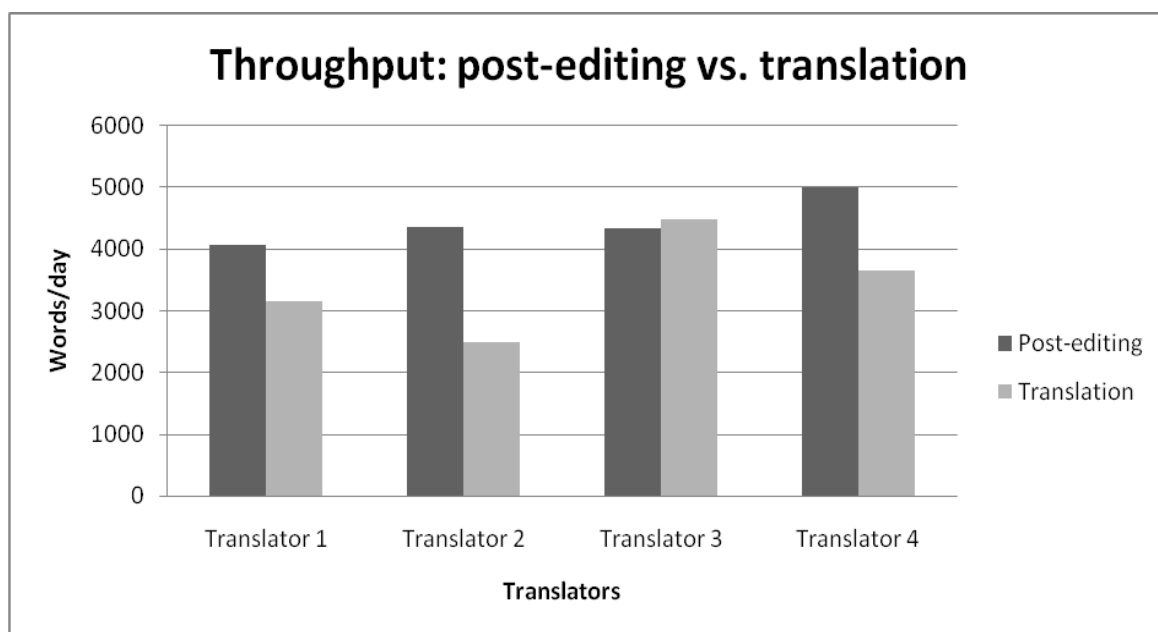
As the graph 7 above shows, productivity across translators and depending on the configurations correlates with translation quality: NLP content was the quickest to post-edit, followed by STANF then NRO.

Concerning jobs, text type 1 yielded better results than text type 2, but variance can be observed among jobs of the same text type (see graph 8). This might be due to the statistical bias in the data that was described in section 4.1 or maybe to an effect of fatigue (the order of the jobs was tut2a, 2b, 3 and then physx1, 2 and 3).



Graph 8 - Productivity (jobs)

In the industry, average daily translation throughput is considered to be 2500 words. Thanks to machine translation, our Japanese translators achieved a productivity that was certainly higher than 2500 words a day³². Three of them were noticeably faster post-editing than translating, as graph 9 below illustrates.



Graph 9 - Productivity: post-editing vs. translation

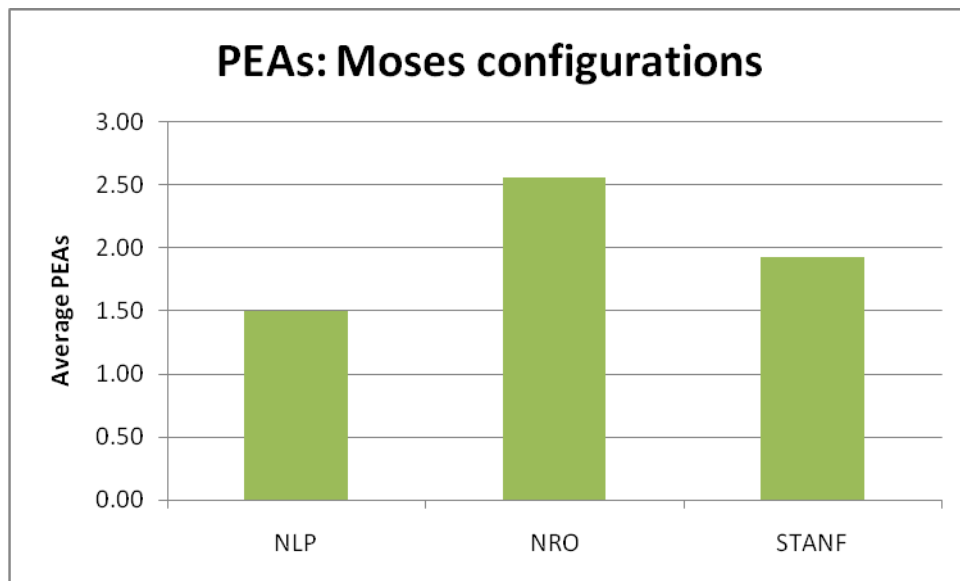
³² The figures in the graph 8 and following point to very high numbers, such as 4000 to 5000 words a day, but these numbers reflect speed trends and not the absolute and actual throughput of translators (i.e. they did not post-edit 5000 words in one day).

5.3 PE effort

This section presents the results of the annotation of post-editing actions. The metric was described in 4.4 and the execution in 4.5. We present averages, recurrent PEAs and other aspects relevant to the analyses.

5.3.1 Average PEAs

Over 1301 annotated segments, 2272 PEAs have been performed. On average, there were 1.5 to 2.57 PEAs per segment. Graph 10 below shows that the most PE effort was spent on NRO:



Graph 10 - Post-editing actions: averages (configurations)

Average PEAs per segment calculated over the translators (table 5) show that individual variance was high, but the number of processed segments was different.

Translators	Average PEAs	Processed segments
1	1.82	257
2	2.14	284
3	1.78	420
4	2.04	340
1.94		

Table 5 - Post-editing actions: averages (translators)

Average PEAs per segment calculated over the jobs (table 6) shows less variance, but physx3 job must be ignored due to the fact that only 1 translator worked on it; average PEAs correlate to productivity over jobs (see graph 8): the more edits, the less productivity.

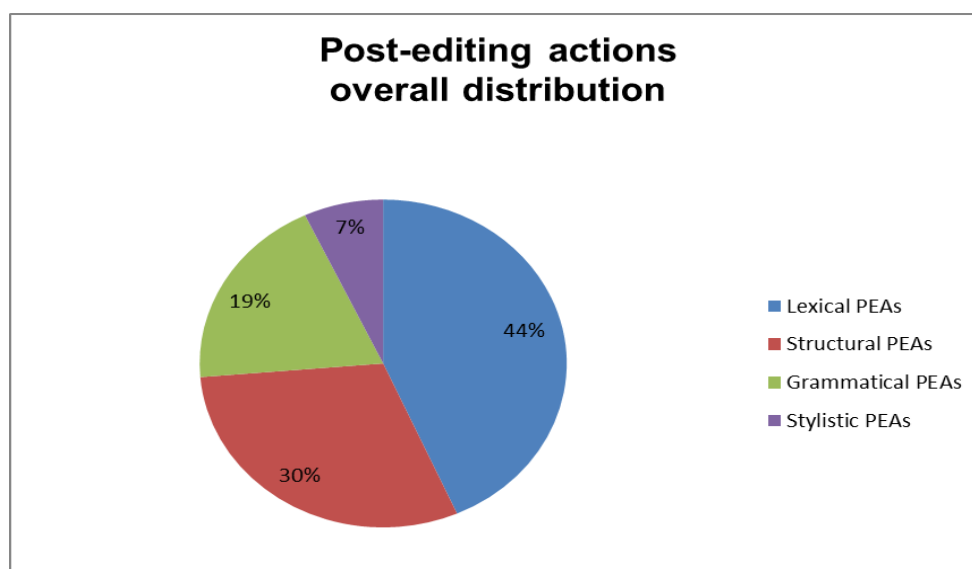
Job name	Average PEAs	Number of translators who processed the job
physx1	1.42	3
physx2	1.86	3
physx3	2.64	1
tut2a	1.56	4
tut2b	1.99	4
tut3	2.55	4
1.94		

Table 6 - Post-editing actions: averages (jobs)

5.3.2 Recurrent PEAs

The analysis of recurrent PEAs brings us to qualitative appreciations. Recurrent PEAs give an indication of the type of linguistic elements that needed more editing. We propose an aggregate view of recurrent PEAs (categories) and a detailed list of PEA types in descending order of frequency, overall, for configurations and for text types.

Overall, 44% of the PEAs were lexical, 30% related to sentence structure, 19% to grammar and 7% were stylistic (graph 11). Interestingly, the MT Quality evaluation found more structural problems than terminology problems (see graph 2), but this discrepancy is mainly due to a methodological problem: the MT evaluation had been designed for different reasons and since its results were not meant to be one day crossed with those of a PE analysis, its criteria were fewer (see page 66) and thus not entirely comparable with PE analysis criteria.



Graph 11 - Post-editing action categories: overall

PEA (category), type	Total
(none) added/deleted words	475
(structural) local-level reordering	349
(lexical) verb changes	313
(lexical) noun change (Autodesk terms)	233
(none) correct MT	191
(grammatical) verb mode changes	167
(structural) clause-level reordering	165
(none) completely rewritten	129
(lexical) noun changes (general language words)	115
(grammatical) category changes	95
(none) miscellaneous	83
(stylistic) lexical stylistic change	80
(lexical) adjective change	76
(grammatical) preposition change	49
(stylistic) clause-level stylistic change	39
(grammatical) verb tense changes	17
(none) PE error	16

Table 7- Post-editing action types: overall

Table 7 groups PEA types for all configurations. The most frequent PEA concerns adding or deleting words. This might not only be related to the fact that SMT fails to produce the right number of target words, but might indicate that a relatively high amount of editing was necessary. The second and third most frequent PEAs are, not surprisingly, reordering and changing the lexical choice of verbs, i.e. the two single most difficult aspects of English to Japanese SMT. Note, however, that in the top 5 we also have noun changes of Autodesk terminology and correct MT, which proves that there was much editing of company terms but also a significant amount of publishable-quality raw MT segments. The sixth and seventh most frequent PEAs confirm the trend stated above that reordering and verbs pose a problem. There were only a few stylistic changes, at least in phrasing, and PE mistakes.

Looking at the different Moses configurations (table 8 below), we can see that there were fewer lexical PEAs in NLP than in NRO and STANF, but that, surprisingly, NLP had more structural and grammatical PEAs than STANF. The “worst” configuration for structure was therefore NRO, and the fact that it had also few stylistic changes might mean that it was of such a quality that fixing the output to make it understandable was a priority, leaving no room for stylistic changes.

Categories	NLP	NRO	STANF	Total
Lexical PEAs	189	269	279	737
Structural PEAs	159	235	120	514
Grammatical PEAs	122	94	112	328
Stylistic PEAs	47	24	48	119

Table 8 - Post-editing action categories: configurations

For details, the tables 9, 10 and 11 below present the PEA types for each configuration.

PEA (category),type	NRO
(none) added/deleted words	171
(lexical) verb changes	143
(structural) local-level reordering	142
(structural) clause-level reordering	93
(none) completely rewritten	70
(grammatical) verb mode changes	59
(lexical) noun change (Autodesk terms)	58
(lexical) noun changes (general	44

language words)	
(grammatical) category changes	29
(lexical) adjective change	24
(none) correct MT	22
(none) miscellaneous	19
(stylistic) lexical stylistic change	18
(stylistic) clause-level stylistic change	6
(none) PE error	5
(grammatical) preposition change	4
(grammatical) verb tense changes	2

Table 9 - Post-editing action types: NRO configuration

In NRO, the top 3 is the same as the overall figures of table 7, but in the top 5 we have yet another reordering PEA and, more importantly, completely rewritten segments.

PEA (category), type	STANF
(none) added/deleted words	193
(lexical) noun change (Autodesk terms)	103
(lexical) verb changes	96
(structural) local-level reordering	80
(none) correct MT	62
(grammatical) verb mode changes	60
(none) completely rewritten	53
(lexical) adjective change	40
(lexical) noun changes (general language words)	40
(structural) clause-level reordering	40
(grammatical) category changes	37
(none) miscellaneous	27
(stylistic) clause-level stylistic change	25
(stylistic) lexical stylistic change	23
(grammatical) verb tense changes	10
(none) PE error	6
(grammatical) preposition change	5

Table 10 - Post-editing action types: STANF configuration

In STANF, the second most frequent PEA is lexical, but reordering, verb changes and correct MT are in the top 5, closely followed by completely rewritten segments.

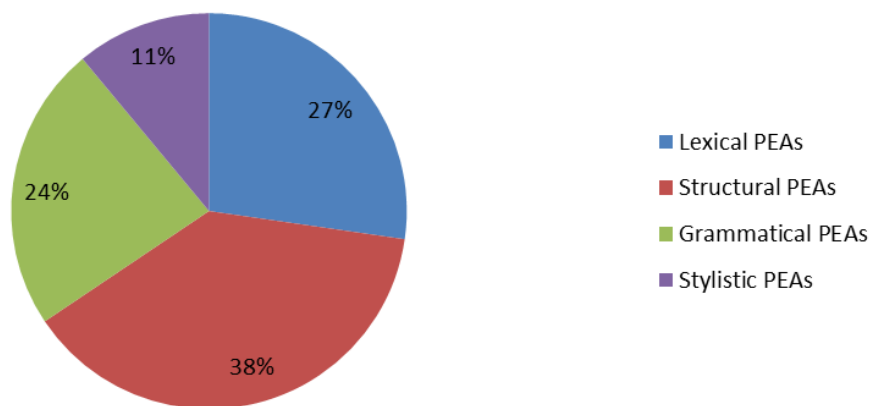
PEA (category), type	NLP
(structural) local-level reordering	127
(none) added/deleted words	111
(none) correct MT	107
(lexical) verb changes	74
(lexical) noun change (Autodesk terms)	72
(grammatical) verb mode changes	48
(grammatical) preposition change	40
(stylistic) lexical stylistic change	39
(none) miscellaneous	37
(structural) clause-level reordering	32
(lexical) noun change (general languagewords)	31
(grammatical) category changes	29
(lexical) adjective change	12
(stylistic) clause-level stylistic change	8
(none) completely rewritten	6
(none) PE error	5
(grammatical) verb tense changes	5

Table 11 - Post-editing action types: NLP configuration

In NLP, reordering accounted for the greatest number of PEAs, but correct MT is in the top 3, which probably explains why it yielded better productivity. Verb changes are not far behind, still in the top 5 with noun changes of Autodesk terms.

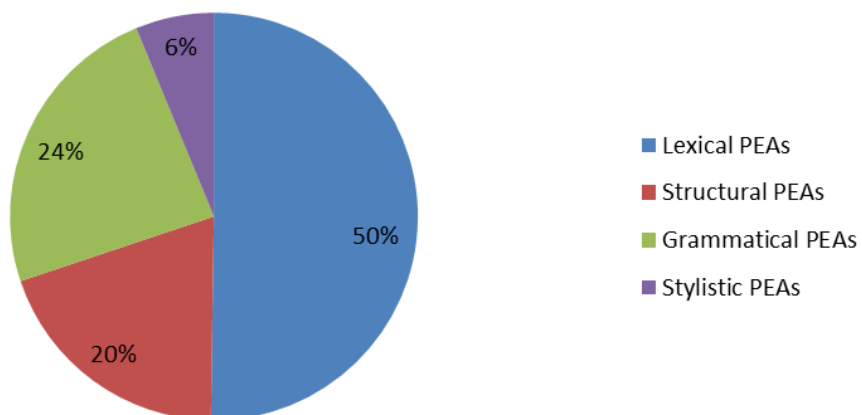
Now looking at the same figures but for text types (graph 12 and 13), the only noticeable difference lies in lexical and structural PEAs. As expected, text type 1 (described in 4.1), similar to already localized content, required fewer lexical edits.

Post-editing actions distribution in text type 1



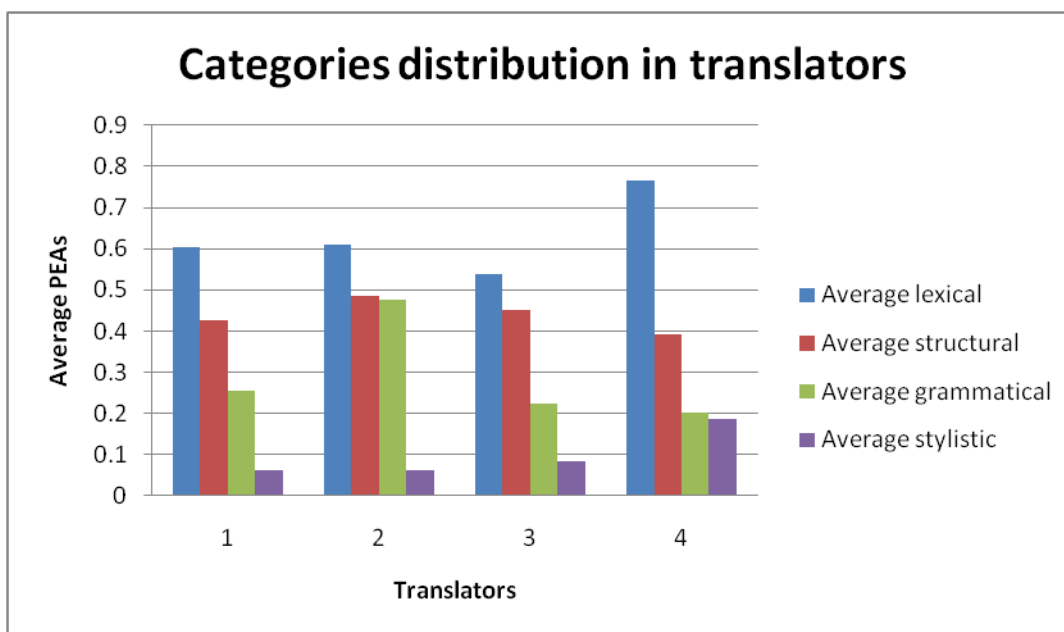
Graph 12- Post-editing action categories: text type 1

Post-editing actions distribution in text type 2



Graph 13 - Post-editing action categories: text type 2

Finally, we would also like to present the results from the point of view of translators, because it must be kept in mind that the relatively low number of participants in the test introduces a bias in the data: there is a risk that the conclusions we draw inform us about the participants' profile in post-editing, rather than about the activity of post-editing itself. The translators all dealt with the same MT quality (see graph 5), with the notable exception of translator 3, who worked on one additional job belonging to text type 2 and the NRO configuration. Graph 14 illustrates the number and type of post-editing actions performed by our four translators. For the statistical reasons explained in section 4.1, for comparisons between translators we must split the group into two and juxtapose translators 1 and 2 to translators 3 and 4. In the first sub-group, only the number of grammatical PEAs is significantly different, but this is probably due to the fact that translator 2 systematically changed the mode of the verb from indicative into imperative for source segments with a command/request. For example, in segments such as “Reset the simulation” the English imperatives were (correctly) machine translated with the indicative mode in Japanese (シミュレーションをリセットします); translators 1, 3 and 4 left the indicative as it was, but translator 2 always changed it into a polite request expressed with the Japanese imperative (シミュレーションをリセットしてください). Source segments with a command/request were numerous, as the aim of the texts is to explain the functionalities of the software they describe. In the second sub-group, there is a significant difference in the number of lexical PEAs. Although we cannot ignore the fact that translator 4 did one job fewer than translator 3, at the same time, we cannot avoid noticing that translator 4 also performed many stylistic changes; therefore, we tend to think that graph 14 shows the profile of translators, and namely that translator 4 prefers (over)editing to leaving the MT proposal as it is.



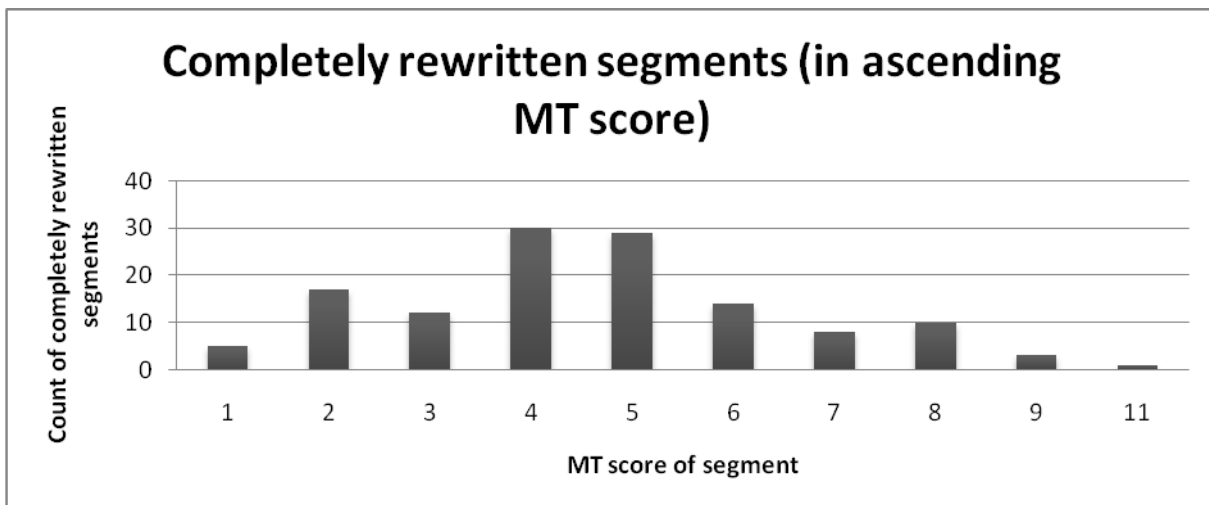
Graph 14 - Post-editing action categories: translators

It is interesting to note that translator 4, who performed the highest number of stylistic changes, was also the most productive translator (see graph 9). This might be interpreted as proof that stylistic changes do not have a negative impact on PE productivity.

5.3.3 PE mistakes and rewritten segments

The reason why we also wanted to look at PE mistakes is that very good MT can be misleading because it looks perfect even though there are errors. In our data, there were only 16 cases of PE mistakes.

In graph 15 below we present segments that were not analyzable for post-editing actions because they were too different from the MT proposal. These segments were filtered out from the results.

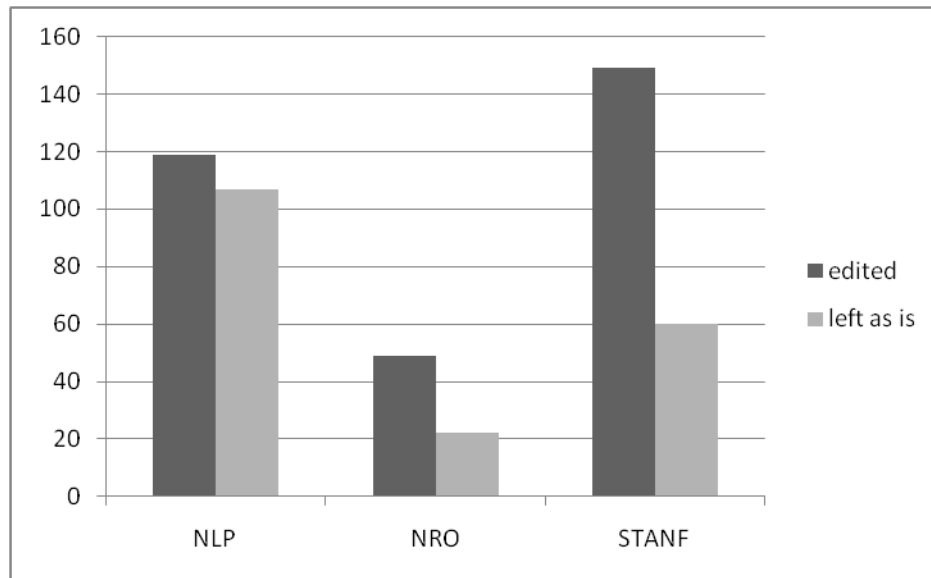


Graph 15 - Completely re-edited segments

As can be seen, only segments with a MT score of 1 to 9 and 11 were concerned by this phenomenon. 0-score segments, i.e. error-free MT, were not concerned, as expected, but even segments with a low MT score (1, 2, 3) were sometimes completely re-edited, which was a surprising result. Despite the fact that the graph suggests that segments with as many as 6 to 11 errors were less concerned by re-writing than segments with a MT score of 4 or 5, we have to consider that: 1) in total only 129 segments over 1301 (the “processed segments”, see 4.1) had been completely rewritten, according to our annotation 2) the number of segments with a MT score between 7 and 11 is relatively low: over the total 450 source segments sent for translation, 5 had a score of 7 and 8, 3 had a score of 9 and 2 had a score of 11.

5.4 Correlation of PE effort with MT quality and PE productivity

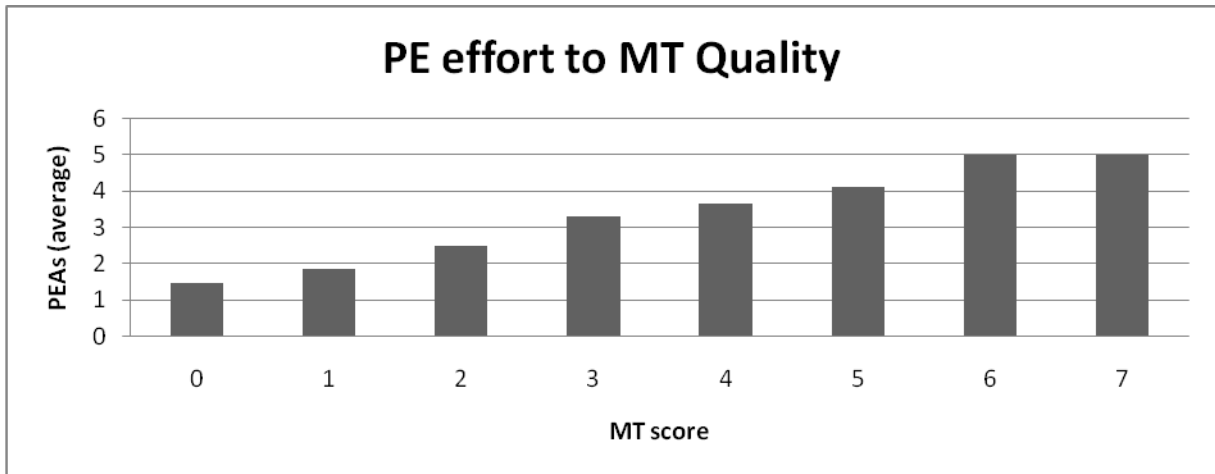
This section presents some combined results: PEAs and MT quality in more detail (graphs 16 to 18) and the relationship between PEAs and productivity.



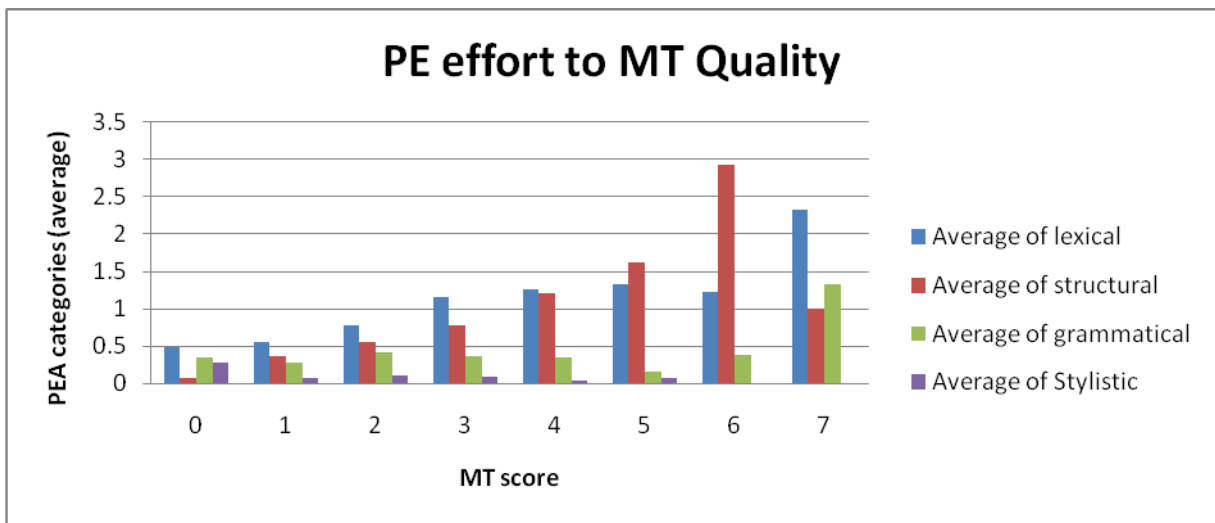
Graph 16 - 0 score MT: amount of edited and not edited segments

Graph 16 above was calculated over 0-score MT segments, i.e. error-free segments. Across all engines, the trend was to edit rather than leave “perfect” segments as they were, which can be interpreted in two ways: either the MT evaluation was not strict enough, or translators felt that they should edit “good” MT to make it even better – or both. The latter potential conclusion is confirmed by graph 18 where we can see that stylistic changes are more frequent in good than in bad MT. Productivity figures over the engines (see graph 7, section 5.2) on this data seem to be proof that such an attitude of overcorrecting does not have a negative impact on speed – at least not to the point of compromising productivity.

If we consider the number of PEAs as an indicator of "PE effort", the graphs 17 and 18 below show results for the average number of PEAs depending on the MT score of the segment. (All segments that scored 8 to 11 were completely re-edited i.e. not annotated for post-editing actions because too different from MT proposal).



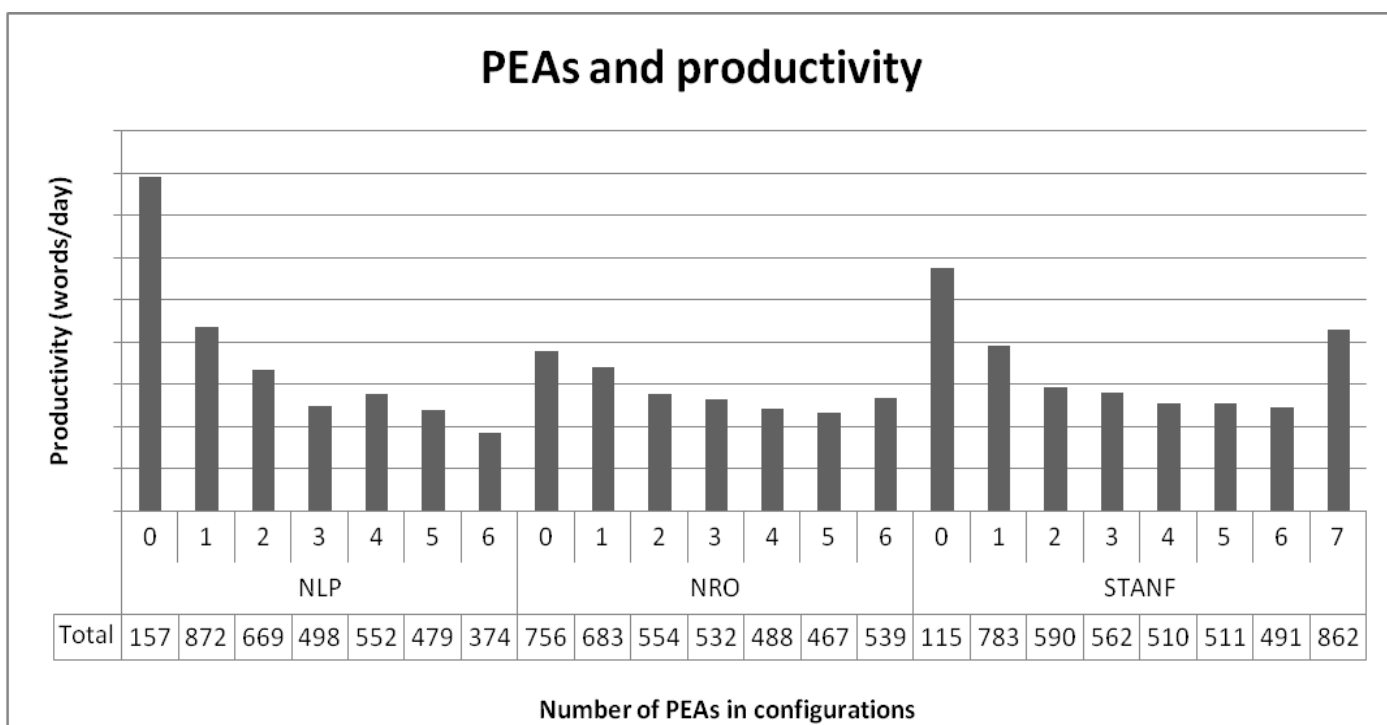
Graph 17 - PE effort related to MT quality



Graph 18 - PE effort (categories) to MT quality

Graph 18 shows the following trends: even “perfect” segments are edited and there is an increasing number of lexical and structural PEs for MT scores from 1 to 5, but a decreasing number of grammatical and stylistic PEs for the same scores. Stylistic changes do not appear on bad-quality MT segments, but rather on good-quality MT segments.

To answer the question of the relationship between PEs and productivity, we present the “cost” of having PEs in the segments, that is to say the post-editing time juxtaposed with the number of PEs. Results (see graph 19) are in words per day and show the cost of having 0 to 6 PEs with all configurations. In other words, graph 19 shows the productivity of segments annotated with 0 to 6 PEs.



Graph 19 – Productivity (wpd) and PEAs (number)

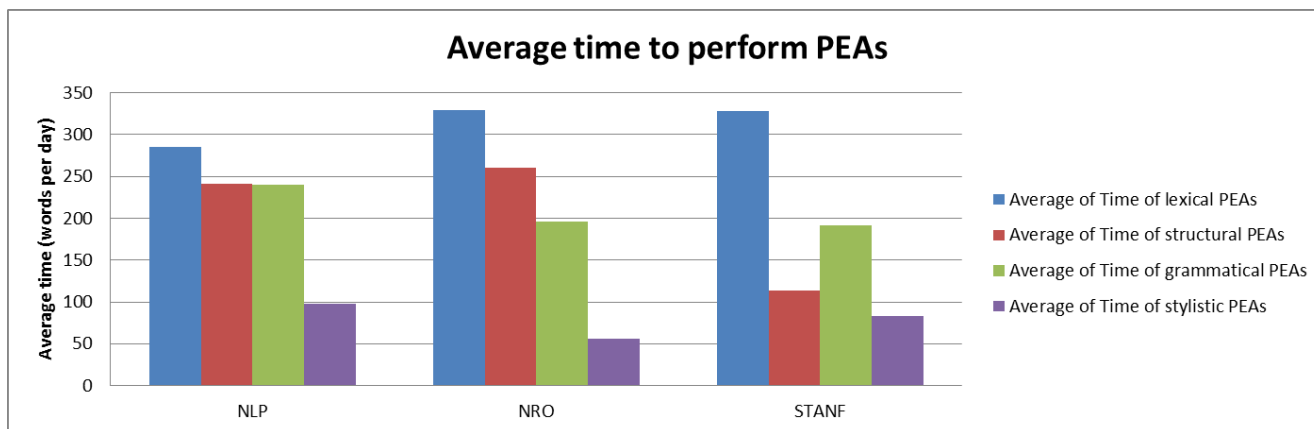
Productivity is highest in the segments where the number of PEAs is lowest, independently of the number of source words. To interpret graph 19, we have to consider the distribution of the number of PEAs in the segments. As shown in table 12 below, this distribution varies over the configurations and STANF was the only configuration with a case of segments with 7 PEA annotations.

PEAs/segment	NLP	NRO	STANF
0	25%	24%	25%
1	33%	21%	23%
2	21%	17%	26%
3	14%	13%	14%
4	4%	14%	8%
5	2%	7%	2%
6	1%	4%	2%
7	--	--	0-1%

Table 12 - Distribution of PEAs over configurations (percentage)

We tentatively calculated the time spent to perform lexical, structural, grammatical and stylistic PEAs by dividing the editing time by the number of PEAs in each segment, and sorting the results by PEA category. The results, expressed in words per day, are presented below in graph 20. The calculation being similar to the one for productivity, high numbers indicate high speed. Again, segment length was not taken into account. We can observe

that, except for structural PEAs, which took more time to perform on STANF segments, on the whole we cannot conclude that one category of PEA takes much more or much less time to be performed than the others. The reason is that lexical PEAs were the most frequent, followed by structural and grammatical in second position and, lastly, stylistic ones (see. graph 11). Graph 20, therefore, suggests that on average, there is a more or less uniform editing time no matter which PEA category is performed.



Graph 20 - Average time to perform PEAs

5.5 Automatic scores vs. PEAs

Although it did not belong to the initial scope of this work, we tentatively calculated the edit distance between raw MT and post-edited versions with GTM³³ and BLEU. The purpose of this was to establish the degree of textual difference with automatic scores, which can be regarded as a traditionally-measured PE effort based on mechanical edits, and see the correlation with productivity; by doing so we can explore the correlation between amount of editing (PE effort) measured by PEAs OR automatic scores and time to perform the edits (productivity). This in turn should give us information if not on the validity of the PEA approach, which is a logical-edits-based, qualitative approach, based on logical edits, at least on the type of results we can expect *in comparison to* mechanical-edits based approaches. It is, in other words, a way to test logical edits against mechanical edits.

³³General Text Matcher, available at <http://nlp.cs.nyu.edu/GTM/>. Turian, J.P., Shen, L., Melamed, I.D., (2003), Evaluation of Machine Translation and its Evaluation in *Proceedings of MT Summit IX*, 23-27 September 2003, New Orleans, USA, pp.386-393.

According to our preliminary results summarized in table 13, GTM scores calculated on all segments of all jobs correlate with productivity better than PEAs do. BLEU scores calculated on a job basis only on text type 1 seemed to correlate with productivity figures just as average PEAs correlate with them.

	Amount of editing (PE effort)	Time to edit (Productivity)	Job based	Segment based
Calculated by	PEAs (number of logical edits)	Words per day	Correlates to some extent	Does not really correlate
Calculated by	GTM (textual difference based on mechanical edits)	Words per day	-	Correlates to some extent
Calculated by	BLEU (textual difference based on mechanical edits)	Words per day	Correlates to some extent-	-

Table 13 - Correlation of edit distance with productivity (tentative)

However, due to time limitations we were not able to explore this topic in a significant manner. Also, for the same reasons we were not able to verify if edit distance was higher with what was defined by PEAs “completely rewritten” segments. Such investigations pertain to potential future work in order to better measure the usefulness of qualitative analyses of post-editing.

In this chapter, we presented the results of our work: linguistic quality of raw MT of the three configurations, which ranked NLP(1), STANF(2) and NRO(3), confirming the importance of reordering, since NRO does not contain this pre-processing step; productivity of the translators in post-editing content translated by these configurations, which confirmed the above-mentioned ranking; PE effort as expressed by the quantity and type of PEAs, for which we provided averages, most recurrent and rewritten segments. Quantitatively, the number of PEAs performed to fix the output of the configurations confirmed the ranking established by the MT evaluation. However, qualitatively, it emerged that the linguistic elements that constituted mistakes according to the MT quality evaluation did not always fully correspond to the linguistic elements that were the most post-edited according to the PEA analysis. The answers that emerge from this work, in other words, are: that MT quality

correlates with productivity by impacting upon it positively, and that PEAs correlate to some extent with productivity; but we could not establish a correlation between MT quality and PEAs. Finally, PEAs are linguistically informative and provided a translator profile.

6 – Conclusion

This study was an attempt to look at a practical application of statistical machine translation from a global perspective, from the early stages of configuration set-up (e.g. choice of reordering rules) to the end of the chain (post-editing) for a language pair that has always been defined “problematic”, with the benefits (and disadvantages) of human evaluation.

We tried to answer the question of what the relationship between MT quality, PE effort and PE productivity is, in the context of English to Japanese post-editing of statistical machine translation of company documentation. We aimed in particular at comparing the configurations of the SMT system and at understanding what happens during post-editing, to maybe guide future implementation strategies for that language pair. To do this, we used a novel post-editing analysis method based on logical edits assuming it would yield more informative results in comparison to edit distance.

Although there were some limitations due to statistical issues pertaining to data scarcity, the questions were all answered: our results all point to the conclusion that, in the described context, MT quality has a direct, positive impact on PE productivity and that PE effort measured by PEAs correlates to some extent with productivity figures, although results were not conclusive. We came to realize that to establish a direct correlation between MT quality and PEAs, the two evaluations should have been designed differently.

The qualitative analysis of Post-editing Actions showed that the aspect of raw MT that needed more editing was different from the main problem identified during the linguistic evaluation of the output, but that there is a direct relationship between MT score and number of post-editing actions. It also highlighted that the typical problems that SMT encounters with our language pair were object of much post-editing, which gives indications on the maturity of the configurations in handling these challenges. Moreover, it helped identify elements that distance-based metrics cannot find, such as post-editing errors, stylistic changes, completely re-edited sentences, and the category of the words the edits were performed on. When putting these results into relationship with time-based calculations of productivity, we were able to draw further conclusions, namely about the cost, in terms of productivity, of having any number and type of PEA. We saw that over our data set, the time to perform any type

and number of PEA was more or less uniform. However, as the tentative measurement of the correlation with automatic scores was not conclusive, future work should certainly explore this matter, as well as the question of the amount of time needed to perform specific PEA types.

We used PEA annotation despite the fact that our MT quality was probably not high enough. This constitutes a drawback because when the MT quality at hand is not high, it is not possible to automate PEA analysis nor to make it very precise, which in turn leads to generalized results, limited to indications as to whether the edits were terminology- or structure-related. To be able, at least, to fully leverage this kind of information, it would be useful to collaborate with a terminology team.

However, we believe that manual PEA annotation for PE effort measurement can be promising in the area of lexical PE edits evaluation, because of the possibility to quantify stylistic changes and identify precisely the grammatical category of the lexical changes. There would be two conditions to obtain meaningful results: first of all, this kind of work needs a more scientific approach to terminology, in order to separate company terms from general language words. Secondly, future work in this sense should consider designing a MT evaluation that involves more than one evaluator and correlates more clearly with the post-editing action categories and types.

Such an evaluation to identify lexical changes could give valuable information concerning the actual appropriateness of SMT in a restricted domain where company terminology has to be enforced. It could also foster informed decisions on whether translators need PE guidelines to help them identify what is considered as a stylistic (and therefore unnecessary) change and what is not, and where the weaknesses of the MT system lie.

In the end, if post-editing actions could be collected over enough translators' work, then ideally it would not always be necessary to have a MT evaluation, depending on requirements.

Bibliography

[Aikawa and Rarrick, 2011] Aikawa, T. and Rarrick, S. (2011) [Are numbers good enough for you? A linguistically meaningful MT evaluation method](#). In *Proceedings of the Thirteenth Machine Translation Summit*, Asia-Pacific Association for Machine Translation [AAMT], September 19-23, Xiamen, China, pp.332-337.

[Aikawa et al., 2007] Aikawa, T., Schwartz, L., King, R., Mo, C.O. and Lozano, C. (2007) Impact of Controlled Language on Translation Quality and Post-editing in a Statistical Machine Translation Environment. In *Proceedings of MT Summit XI*, Copenhagen, Denmark, pp. 1-7.

[Allen, 2003] Allen, J. (2003) Post-editing. In Somers, H. [ed] *Computers and Translation: A Translator's Guide*. John Benjamins Publishing, Amsterdam, pp. 297-317.

[Arnold et al., 1994] Arnold, D., Balkan, L., Meijer, S., Humphreys, R. and Sadler, L. (1994). *Machine Translation: an Introductory Guide*. Blackwells-NCC, London.

[Blain et al., 2011] Blain, F., Senellart, J., Schwenk, H., Plitt, M. and Roturier, J. (2011) [Qualitative analysis of post-editing for high quality machine translation](#). In *Proceedings of the Thirteenth Machine Translation Summit*, Asia-Pacific Association for Machine Translation [AAMT], September 19-23, Xiamen, China, pp.164-171.

[Bourland, 2011] Bourland, W. (2011) Why MT gets more talk than action. In *MultiLingual Computing*, July/August, p.62

[Brown et al., 1990] Brown, P. F., Cocke, J., Della Pietra, S. A., Della Pietra, V. J., Jelinek, F., Lafferty, J. D., Mercer, R. L. and Roossin, P. S. (1990). A statistical approach to machine translation. In *Computational Linguistics*, vol. 16, n°2, pp. 79– 85.

[Dugast et al., 2007] Dugast, L., Senellart, J. and Koehn, P. (2007) Statistical Post-Editing on SYSTRAN's Rule-Based Translation System. In *Proceedings of the Second Workshop on Statistical Machine Translation*, Prague, pp. 220-223.

[EAGLES, 1999]EAGLES Evaluation Working Group (1999) The EAGLES 7-step recipe, online document in University of Geneva, TIM/ISSCO, Research, Projects,

Completed projects, last visited in December 2011,
<<http://www.issco.unige.ch/en/research/projects/eagles/ewg99/7steps.html>>

[Esselink, 2000] Esselink, B. (2000). A Practical Guide to Localization. Language International World Directory, John Benjamins, Amsterdam.

[Estrella et al., 2005] Estrella, P., Popescu-Belis, A., Underwood, N. (2005) Finding the System that Suits you Best: Towards the Normalization of MT Evaluation. In *Proceedings of the Twenty-seventh International Conference on Translating and the computer*, November 24-25, London.

[Flanagan, 1994] Flanagan, M. (1994). Error classification for MT evaluation. In *Proceedings of the AMTA Conference*, Columbia, Maryland.

[Flournoy and Duran, 2009] Flournoy, R. and Duran, C. (2009). Machine translation and document localization at Adobe: from pilot to production. In *Proceedings of the twelfth Machine Translation Summit*, Ottawa, Canada, pp. 425- 428.

[Font Llitjós et al., 2005] Font Llitjós, A., Carbonell, J.G. and Lavie, A. (2005) [A framework for interactive and automatic refinement of transfer-based machine translation](#). In *Proceedings of the 10th EAMT conference "Practical applications of machine translation"*, May 30-31, Budapest, pp. 87-96.

[Grove and Schmidtke, 2009] Groves, D. and Schmidtke, D. (2009) Identification and analysis of post-editing patterns for MT. In *Proceedings of MT Summit XII*, Ottawa, Ontario, Canada, pp. 429-436.

[Groves, 2008] Groves, D. (2008). Bringing Humans into the Loop : Localization with Machine Translation at Traslan. In *Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas*, Waikiki, Hawaii, pp. 11–22.

[Guerberof, 2008] Guerberof, A.A. (2008) Productivity and quality in the post-editing of outputs from translation memories and machine translation. In *The International Journal of Localisation*, vol. 7, nº1, pp. 11-21.

[Guzmán, 2007] Guzmán, R. (2007) Automating MT post-editing using regular expressions. In *MultiLingual Computing*, nº90, vol.18, issue 6, pp. 49-52.

[Hutchins and Somers, 1992] Hutchins, W. J. and Somers, H. L. (1992). An Introduction to Machine Translation. Academic Press.

[Hutchins, 1986] Hutchins, W. J. (1986). Machine translation: past, present, future. John Wiley & Sons, Inc., New York, NY, USA.

[Isozaki et al., 2010] Isozaki, H., Hirao, T., Duh, K., Sudoh, K. and Tsukada, H. (2010) Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, Cambridge, MA, October, Association for Computational Linguistics, pp. 944–952.

[King et al., 2003] King, M., Popescu-Belis, A., Hovy, E. (2003) FEMTI -Creating and using a framework for MT evaluation. In *Proceedings of the ninth machine translation Summit*, September 23-27, New Orleans, pp. 224–231.

[King, 2005] King, M. (2005) Accuracy and suitability: new challenges for evaluation. In *Language Resources and Evaluation*, vol.39, pp.45–64.

[Koehn et al., 2007] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E. (2007) Moses: Open Source Toolkit for Statistical Machine Translation. *Annual Meeting of the Association for Computational Linguistics [ACL]*, demonstration session, Prague, Czech Republic.

[Koehn, 2010] Koehn, P. (2010). Statistical Machine Translation. Cambridge University Press.

[Krings, 2001] Krings, H.P. (2001) Repairing Texts: Empirical Investigations of Machine Translation Post-Editing Processes. The Kent State University Press, Kent, Ohio.

[Lavie, 2010] Lavie, A. (2010) Essentials of machine translation evaluation. Online article in *TAUS Translation Automation*, last visited in October 2011, <<http://www.translationautomation.com/best-practices/essentials-of-machine-translation-evaluation.html>>

[Manning and Schütze, 1999] Manning, C. and [Schütze](#), H. (1999) Foundations of Statistical Natural Language Processing, MIT Press. Cambridge, MA.

[McEnery, 2003] McEnery, T. (2003) Corpus Linguistics. In Mitkov, R. [ed] *The Oxford Handbook of Computational Linguistics*. Oxford University Press, Oxford, pp. 448-463.

[Morland, 2002] Morland, V. (2002) Nutzlos, Bien pratique, or Muy Util? Business Users Speak Out on the Value of Pure Machine Translation. In *Proceedings of the Twenty-fourth International Conference on Translating and the computer*, November 21-22, London.

[Naskar et al., 2011] Naskar, S.K., Toral, A., Gaspari, F. and Way, A. (2011) [A framework for diagnostic evaluation of MT based on linguistic checkpoints](#). In *Proceedings of the Thirteenth Machine Translation Summit*, Asia-Pacific Association for Machine Translation [AAMT], September 19-23, Xiamen, China, pp.529-536.

[O'Brien, 2004] O'Brien, S. (2004) Machine Translatability and Post-Editing Effort: How do they relate? In *Proceedings of the Twenty-fifth International Conference on Translating and the computer*, November 18-19, London.

[Plitt and Masselot, 2010] Plitt, M. and Masselot, F. (2010) A productivity Test of Statistical Machine Translation Post-Editing in a Typical Localisation Context. In *The Prague Bulletin of Mathematical Linguistics*, vol. 93, pp. 7-16.

[Roturier, 2009] Roturier, J. (2009) Deploying novel MT technology to raise the bar for quality: a review of key advantages and challenges. In *Proceedings of MT Summit XII*, Ottawa, Ontario, Canada, pp. 1-8.

[Schenker, 2001] Schenker, J. (2001) The gist of translation. Online article in *TIME Magazine*, last visited in December 2011, <<http://www.time.com/time/world/article/0,8599,2047621,00.html>>.

[Schiaffino and Zearo, 2009] Schiaffino, R. and Zearo, F. (2009) Translation quality measurement in practice.

[Somers, 2003] Somers, H. (2003) Machine Translation: Latest Developments. In Mitkov, R. [ed] *The Oxford Handbook of Computational Linguistics*. Oxford University Press, Oxford, pp. 512-528.

[Specia, 2011] Specia, L. (2011) Quality Estimation for Machine Translation: different users, different needs. Presentation at the *JEC Workshop*, October 14, Luxembourg.

[Talbot et al., 2011] Talbot, D., Kazawa, H., Ichikawa, H., Katz-Brown, J. Seno, M. and Och, F.J. (2011) [A lightweight evaluation framework for machine translation reordering](#). In

Proceedings of the 6th Workshop on Statistical Machine Translation, July 30-31, Edinburgh, Scotland, UK, pp.12-21.

[Tatsumi and Roturier, 2010] Tatsumi, M. and Roturier, J. (2010) Source Text Characteristics and Technical and Temporal Post-Editing Effort: What is Their Relationship? In *Proceedings of the Second Joint EM-CNGL Workshop [JEC '10]*, November 4, Denver, CO, pp. 43-51.

[Thicke, 2011] Thicke, L. (2011). Do-it-yourself machine translation at Autodesk. In *MultiLingual Computing*, September 2011, pp.15-17.

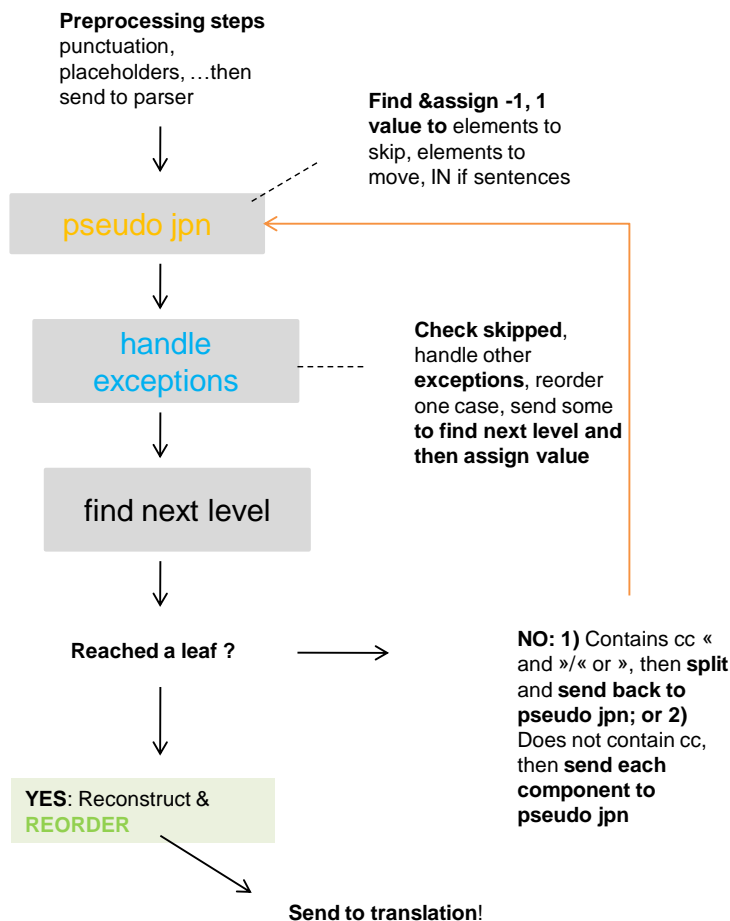
[Van der Meer, 2003] Van der Meer, J. (2003) The business case for Machine Translation. In *Proceedings of the Twenty-fifth International Conference on Translating and the computer*, November 20-21, London.

[Van Slype, 1979] Van Slype, G. (1979) Critical Study of Methods for Evaluating the Quality of MT. Technical Report BR 19142, European Commission, Directorate for General Scientific and Technical Information Management [DG XIII] Available from <www.issco.unige.ch/projects/isle>.

[Vilar et al., 2006] Vilar, D., Xu, J., D'Haro, L.F. and Ney, H. (2006) [Error analysis of statistical machine translation output](#). In *Proceedings of LREC-2006: Fifth International Conference on Language Resources and Evaluation*, May 22-28, Genoa, Italy, pp.697-702.

Appendix A – Reordering rules

In this appendix we present the simplified flowcharts of the NLP reordering rules' script.



To skip (1): VP (VBN ; VP (VBG; VP (VBD
To move to end (-1): VB, VBD, VBG, VBP, VBZ, VP, IN, MD, TO to, RB not, RB 't
To move to beginning (1): SBAR, S, VP (TO to) (VP, VP (TO to) (VB, VP (TO to)(VP (TO to) (ADVP, WDT, PP(IN, PP(TO

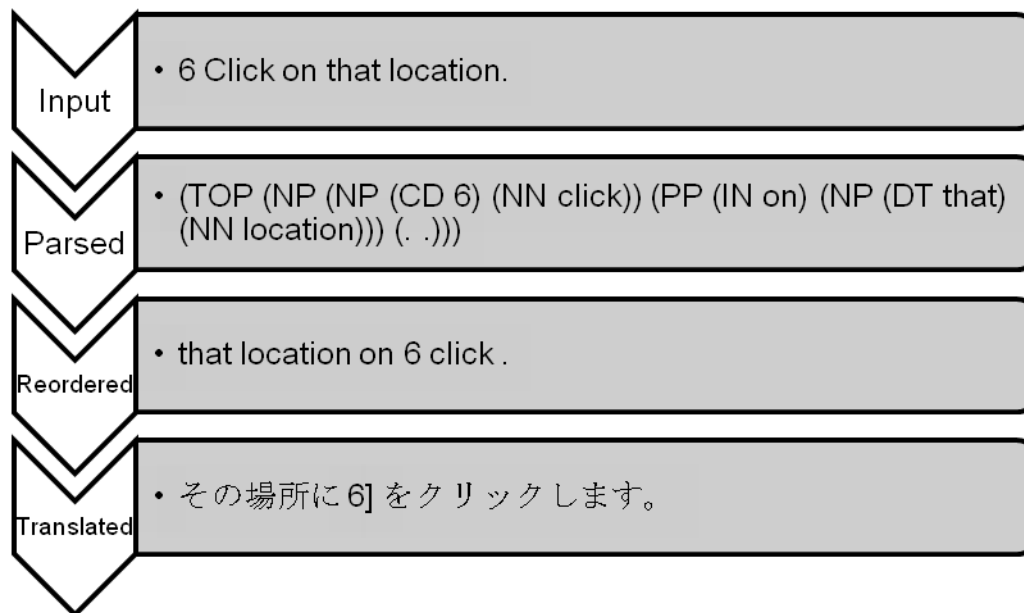
Skipped, to move to end: VP (VBD
Exception (to locally reorder): VB(X) CC VB(X)
To send to find next level: SBAR, S, VP (TO to) (VP, VP (TO to) (VB, VP (TO to)(VP (TO to) (ADVP, WDT, PP(IN, PP(TO and then assign value

1 values → append to array « hardClause » then reverse order
-1 values → append to array « verb » and then reverse order
0 → is « easyClause »
Segment = hardClause – easyClause - verb

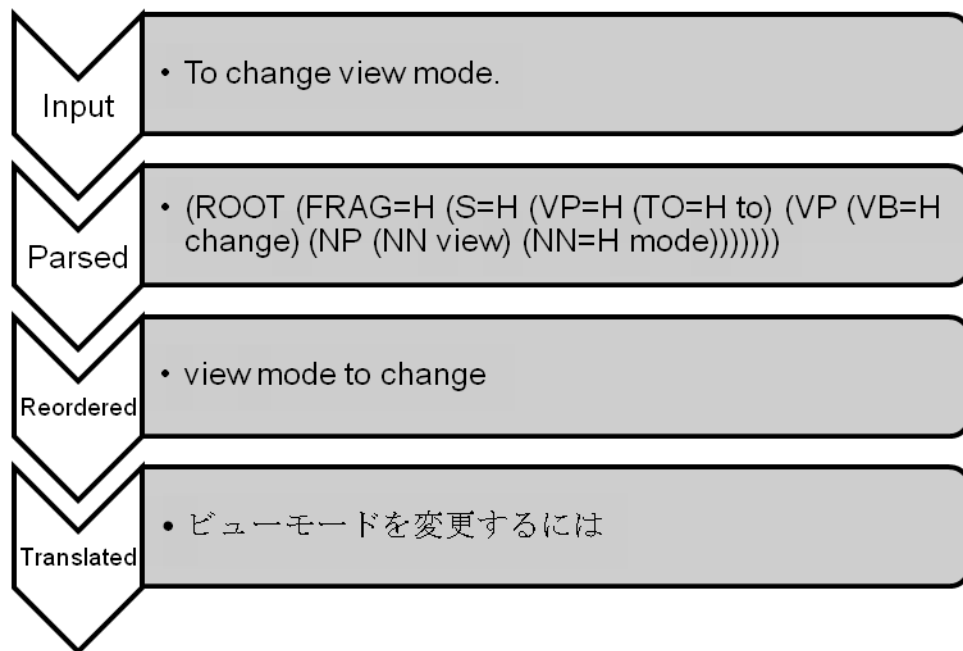
Flowchart 1 - NLP reordering rules

Appendix B – Translation examples

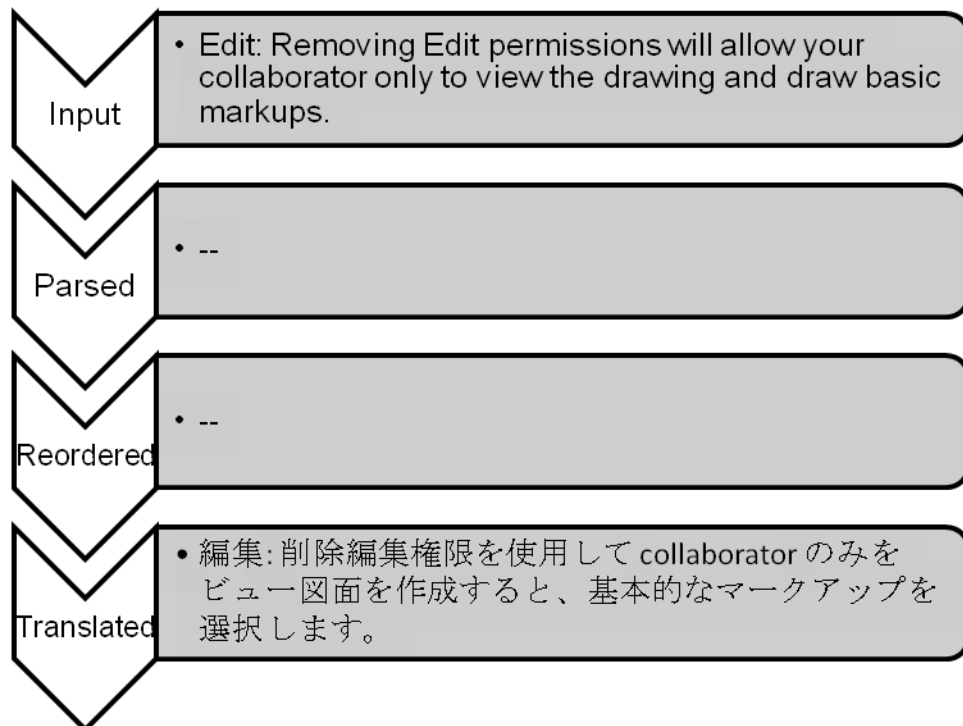
In this appendix we would like to present examples taken from the thesis' data to show the entire translation process from input to output for two text types with all configurations (for details on the data set, refer to section 4.1).



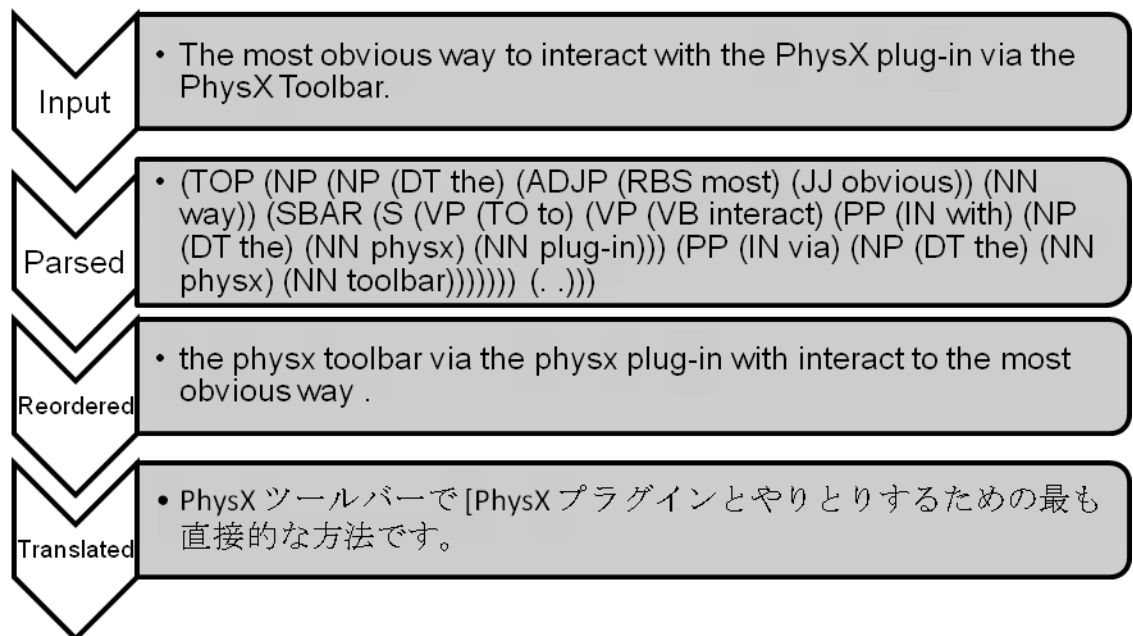
Translation process example: text type 1, NLP configuration



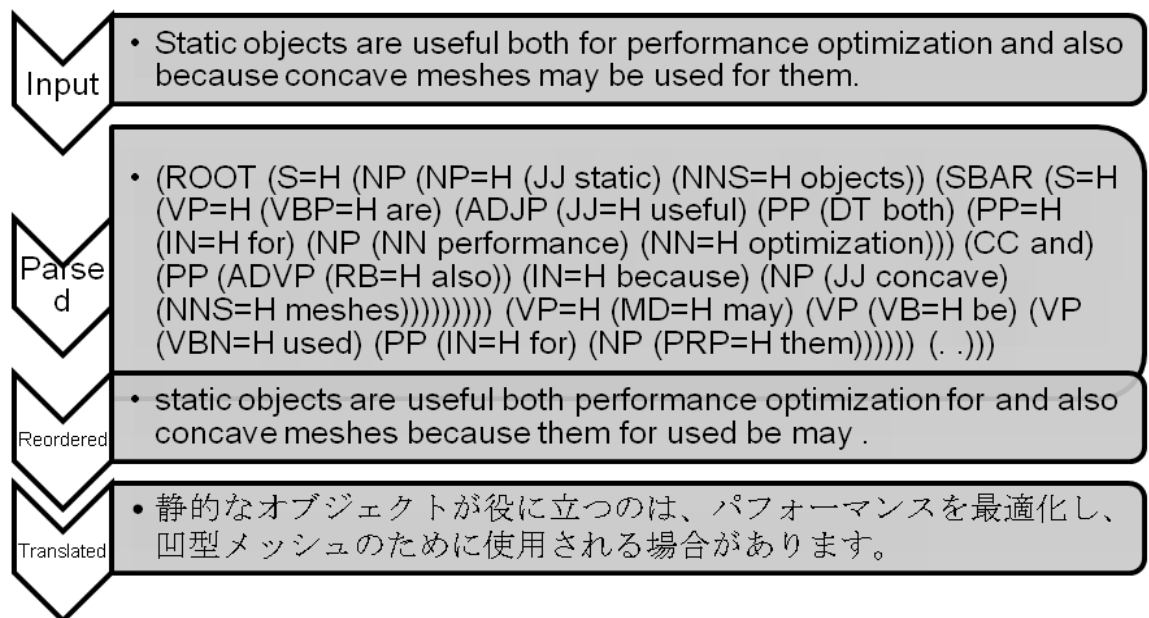
Translation process example: text type 1, STANF configuration



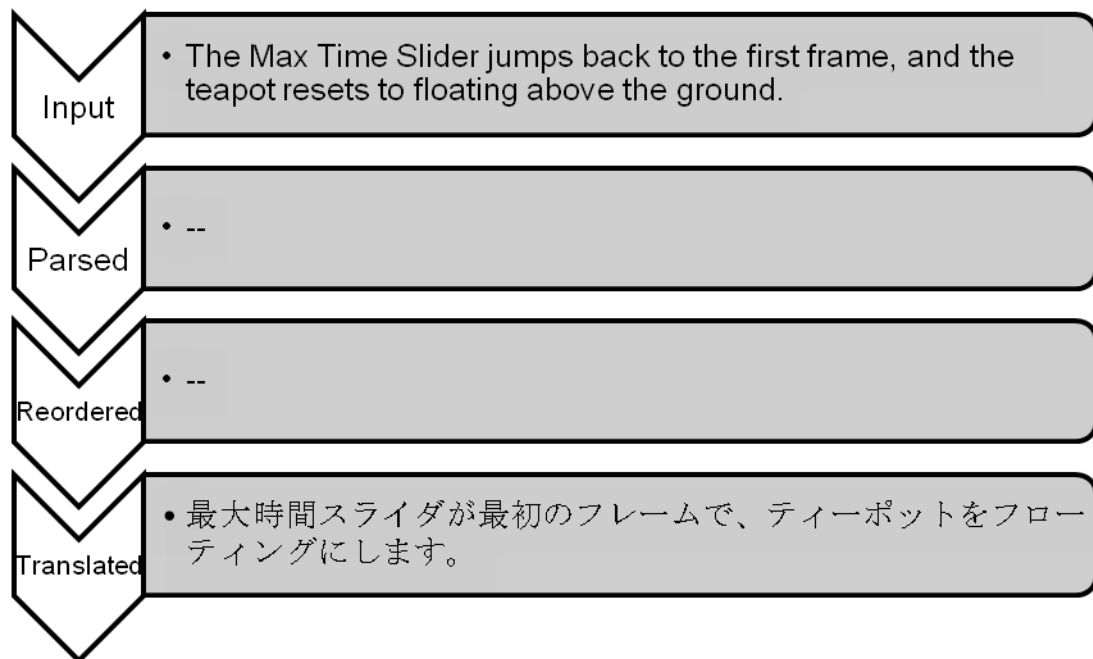
Translation process example: text type 1, NRO configuration



Translation process example: text type 2, NLP configuration



Translation process example: text type 2, STANF configuration



Translation process example: text type 2, NRO configuration

Appendix C – XML files

The annotated XML format data can be obtained on demand by writing to the author at stephanie.dirosa_at_gmail.com.