



«MOSES, MOSES:  
LET MY PEOPLE GO»

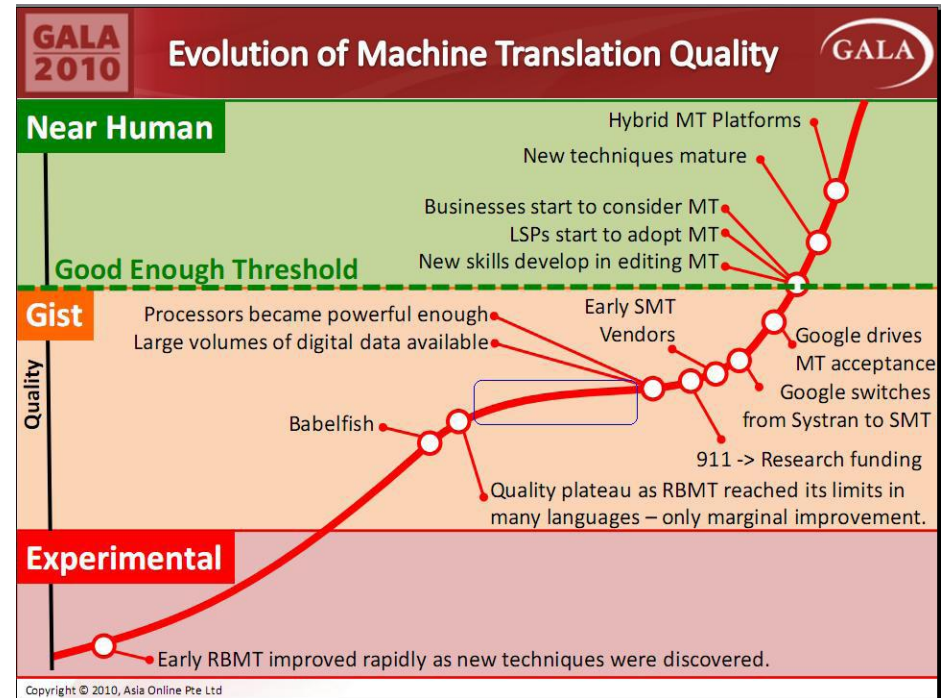
# Moses MT engine feasibility study

Serge Gladkoff, President of Logrus International  
for TAUS Portland, October 2010

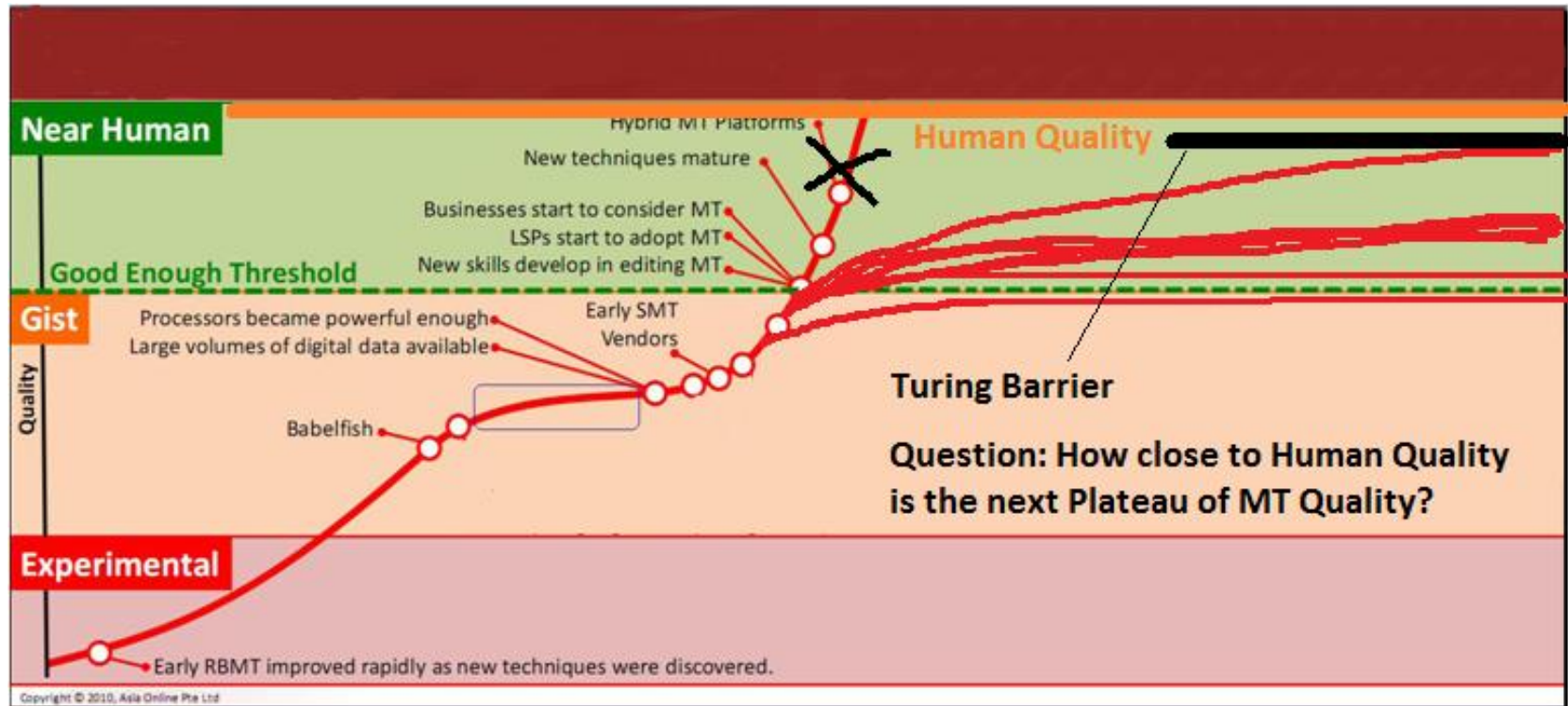
# Do we witness a breakthrough?



1. There's a hype. The number of publications and actual deployments is growing. (Twitter posts on MT made 50% of all tweets on LocWorld Berlin 2010!).
2. Statistical systems seem to demonstrate real quality progress. Marriage with RBMT is especially promising.
3. Reports claim economic efficiency of some MT deployments, although few reliable ROI data is available.
4. New factors: a) huge amount of linguistic data available, b) further leaps in processing power, c) cost and time pressures.
5. OpenSource decreases the cost of development and lowers the barrier to entry into new technology.
6. Key clients have already integrated MT in their workflows and some projects are coming with components of MT output.



# More realistic picture



# HYPOTHESIS and goals of the experiment



1. TEST HYPOTESIS – Is the Plateau of quality behind? How far is the next plateau of quality from Human Translation?
2. Is the MT technology available for LSPs without multimillion funding?
3. Is OpenSource SMT Moses a viable place to start?
4. Can we base our own productivity solutions on Moses in any near future?
5. What are the possible LSP MT-related process implications?
6. What technical issues could arise?
7. What are the methods to calculate ROI?
8. What are post-editing differences in MT scenario?
9. Is my company capable of handling this?

# Resources



1. ONE (very powerful) computer.
2. ONE (very experienced) Project Manager – MT skeptic!
3. ONE (very experienced), talented programmer.
4. Very limited task definition and scope.
5. NO prior experience with MOSES – starting from zero.
6. Intentionally very limited Moses tweaking – **no** going deep into Moses code.
7. TAUS DATA Founding Member status + €15,000 to get full access to corpus.
8. Corporate culture of scientific research, automation, innovation.
9. Total number of internal hours for this pilot =~ 360 man-hours.

# First steps with Moses



1. We downloaded the system, installed and set the minimum standard, default configuration.
2. Data was largely taken from TDA corpus, but we have also used our translation memories on IT (English - Russian pair).

The data has to be cleared from garbage to train Moses – we removed complex formatting and tags, also simply dropped invalid segments. This has been identified as major obstacle and project effort.

Currently TDA is working on cleaning the data, but at that moment they were far from clean (10-20% of all units were irrelevant for Moses training)

- Removed tags for RTF, HTML, XML
- Removed/fixed strings with wrong encoding
- Removed untranslated strings (source = target);
- Removed strings with no text (numbers, dates) or empty target

# Playing around



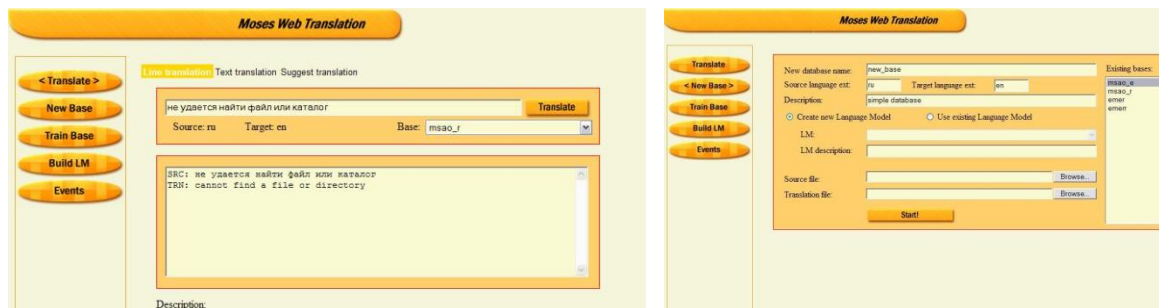
1. We evaluated not only language quality as such, but also the convenience of the system installation and setup, difficulty of fine-tuning and use, and system stability.
2. We have tried various Moses distributions, language models and Linux distributions.
3. We have decided that IRST is the optimal language model.
4. Various language parameters of Moses have been played with, to get the optimal result.

· -weight-d (-d): weight(s) for distortion (reordering components)	[input-factors]
· -weight-e (-e): weight for word deletion	0
· -weight-file (-wf): file containing labeled weights	[mapping]
· -weight-generation (-g): weight(s) for generation components	T 0
· -weight-i (-I): weight for word insertion	[ttable-file]
· -weight-l (-lm): weight(s) for language models	0 0 5 /home/atma/BASES/db/msao_e/Ingbase/msao_e-MAIN
· -weight-t (-tm): weights for translation model components	[lmodel-file]
· -weight-w (-w): weight for word penalty	1 0 3 /home/atma/BASES/db/msao_e/lm/1.gz
· -phrase-drop-allowed (-da): if present, allow dropping of source words	[ttable-limit]
· -distortion: configurations for each factorized/lexicalized reordering model.	20
· -distortion-limit (-dl): distortion (reordering) limit in maximum number of words (0 = monotone, -1 = unlimited)	0
· -max-phrase-length: maximum phrase length (default 20)	[weight-d]
· -mbr-size: number of translation candidates considered in MBR decoding (default 200)	0.6
· -minimum-bayes-risk (-mbr): use minimum Bayes risk to determine best translation	[weight-l]
· -monotone-at-punctuation (-mp): do not reorder over punctuation	0.5000
· -n-best-factor: factor to compute the maximum number of contenders (=factor*nbest-size). value 0 means infinity, i.e. no threshold. default is 0	[weight-t]
	0.2
	0.2
	0.2
	0.2
	0.2
	[weight-w]

# Final Logrus Moses system configuration



1. Hardware: Quad-Core AMD-64, 16 Gbytes of RAM, 64-bit Linux Ubuntu, 100 Gbytes of disk space.
2. Moses configuration: the most recent build, IRST language model, GIZA++ (training component).
3. We have created Web interface to the system.
4. Moses training has to be automated, and was automated, to try the system with different SME corpuses
5. Other SW: our proprietary scripts developed to train Moses, its setup and remote (Web) client access.



# Training, Training, Training



Training and data cleaning was lengthy and iterative process, had to be automated.

**Moses Web Translation**

Translate  
New Base  
Train Base  
Build LM  
< Events >

```
[05.05.2010 17:11:27] tt2: begin upload new database
[05.05.2010 17:11:28] tt2: source upload finished
[05.05.2010 17:11:28] tt2: translation upload finished
[05.05.2010 17:11:33] tt2: cleaning finished
[05.05.2010 17:11:38] tt2: build lm finished
[05.05.2010 17:15:48] tt2: training finished
[05.05.2010 17:15:48] tt2: starting language server
[05.05.2010 17:17:14] rtaus: begin upload new database
[05.05.2010 17:17:28] rtaus: source upload finished
[05.05.2010 17:17:36] rtaus: translation upload finished
[05.05.2010 17:18:04] rtaus: cleaning finished
[05.05.2010 17:34:14] rtaus: begin upload new database
[05.05.2010 17:34:44] rtaus: source upload finished
[05.05.2010 17:35:07] rtaus: translation upload finished
[05.05.2010 17:35:33] rtaus: cleaning finished
[05.05.2010 17:39:08] rtaus: build lm finished
[11.05.2010 15:16:47] emer: begin upload new database
[11.05.2010 15:16:48] emer: source upload finished
[11.05.2010 15:16:48] emer: translation upload finished
[11.05.2010 15:16:53] emer: cleaning finished
[11.05.2010 15:16:58] emer: build lm finished
[11.05.2010 15:18:58] emer: training finished
[11.05.2010 15:18:58] emer: starting language server
[20.05.2010 16:37:04] emerr: begin upload new database
[20.05.2010 16:37:04] emerr: source upload finished
[20.05.2010 16:37:04] emerr: translation upload finished
[20.05.2010 16:37:09] emerr: cleaning finished
[20.05.2010 16:37:14] emerr: build lm finished
```

Logrus 2010

# Results



Language pair \ MT system (developer)	Edit Distance (average value of Character Edit Rate <sup>1</sup> )						
	SMT	SMT (*)	SMT (**)	RBMT	RBMT	<# 4>	<# 5>
	(Bing — Microsoft)	Logrus Moses	Logrus Moses	(ProMT v8)	(SYSTRAN)	(SDL)	(Google)
English – French	0.45	-	-	0.54	0.51	-	-
English – Italian	0.46	-	-	0.59	0.51	-	-
English – Portuguese	0.42	-	-	0.58	0.58	-	-
English – Russian	0.53	0.57	0.61	0.63	0.69	-	-
English – Spanish	0.35–0.37	-	-	0.46	0.45	0.40	0.35

(\*) Logrus Moses after deep fine-tuning, trained on very narrow IT domain.

(\*\*) Logrus Moses after deep fine-tuning, trained on much wider IT domain (TAUSDATA)

1. We have obtained Moses translation quality results in standard configuration, without deep fine-tuning.
2. The same after deep fine-tuning.
3. The MT output results from various systems were evaluated on one and the same sample.
4. Translation quality has been evaluated with Edit Distance algorithm (the software tool was developed in Logrus).
5. The “pure” result was better than with untrained PROMT 8 and was close to BING Translator from Microsoft.

<sup>1</sup> rate of the number of changes introduced after post-editing, to the sentence length.

# Problems of MT integration into actual process



1. Large paragraphs (>255 characters). Moses fails to build a match table on them, so we had to break them down.
2. Punctuation marks are treated as letters. Multi-sentence segments are not treated correctly and add noise.
3. Basic Moses configuration only work with lowercase letters. Additional processing is required to restore case.
4. RTF, XML, HTML tags are not supported.
5. Training corpus often contains ambiguous, obsolete terms. Extensive terminology support is required.
6. Every Moses user has to develop his own interface between Moses and his own workflow, CMS, TMs, projects, etc.

# Moses Features, Advantages, and Disadvantages



## 1. ADVANTAGES:

- Free
- Amazingly high quality of basic system after proper tweaking
- Minimal setup provides competitive language quality level
- Practical solutions can be based on Moses

## 2. DISADAVTAGES:

- Moses is not a ready tool – it's a construction set
- Serious efforts are required to bring it to practical use

## 3. PROBLEMS:

- Training corpuses require cleaning
- TMX is poorly supported by existing software
- Domains of feasibility are so specific that it is not clear whether the effort worth it yet for “generic” LSP

# Conclusions – “scientific” aspects



1. Hypothesis is true – it does look like SMT breaks the MT quality barrier known until recently. Another plateau is within usable domain in certain situations.
2. We experienced the benefits of statistics-based method and its limitations.
3. We obtained information on areas where SMT produces best results.
4. We have found the configuration setting and training technique – Moses experience that we can take further.
5. We concluded that further research is required and beneficial.
6. Terminology requires special attention. Large corpus is not always better than smaller one. Very serious inconsistencies in terminology in corpus, particularly detrimental for SMT engines.
7. SMT vs RBMT choice criteria refined.

# Conclusions – technical aspects



1. TAUS DATA: insufficient number of data categories (“genres”, content types). Extremely diverse data are grouped together, requires domain separation.
2. Too few TMX editors on the market, most of them too expensive. Multiple problems with TMX support, language support and/or stability of these editors.
3. Multiple problems with Import/Export in all these editors. Logrus had to develop a TMX editor of its own, and we are planning to provide it for free to cooperating parties in the near future. Other useful proprietary tools and techniques have been developed can be shared.
4. No MT – TM – Translation Environment integration yet.
5. Terminology-Related Problem: Linking to Various Data Corpora  
RBMTs require grammatical attributes, while STAT-based MTs require fine, “genre”-related classification.

# Conclusions – practical aspects

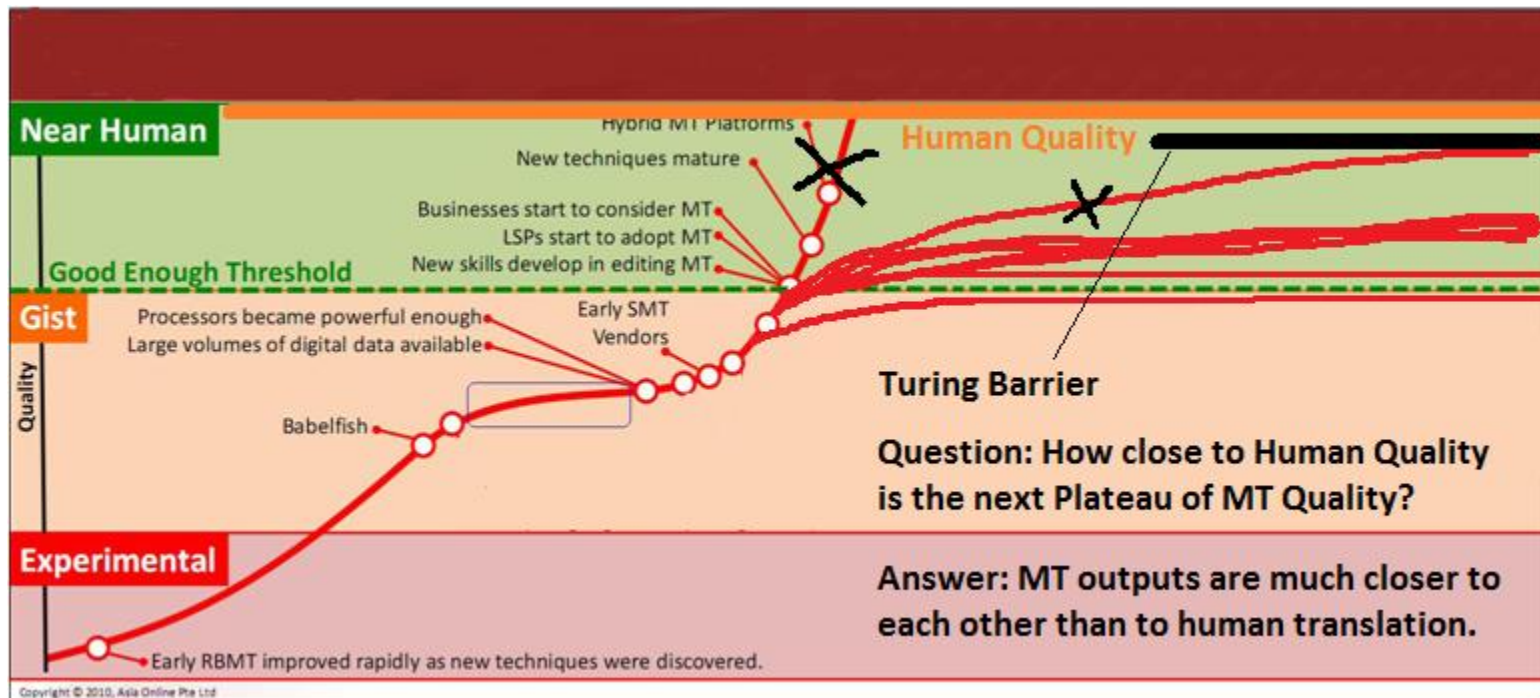


1. Overall linguistic quality of MT output is still low: **Output from MT engines are much closer to each other than to final quality human translation.** (We are still very far from Turing barrier.)
2. Mass post-editing is required. **Editing with reference to the source is a MUST.**
3. Dedicated terminology support is a must. Complete, 100% terminology post-verification is required.
4. MT post-editing is different from human editing, and editors must be trained.
5. With "Difficult" language pairs the cost of post-editing would be 50%-75% of the cost of human translation, **in best case scenario, after costly preparation.** Overall ROI must include internal R&D and/or license and deployment costs.
6. Continuous internal R&D is required to make use of the technology.
7. ROI as a consequence is very specific for particular project and company; further research into methods to evaluate ROI is required.
8. The domain of positive ROI increases. LSPs should pay attention.
9. Significant localization process reengineering is required for MT deployment on all stages: corpus, MT setup and administration, deployment, vendor selection, post-editing processes.
10. Languages vary – the process would be different for each language (SMT vs RBMT, parameters, quality level, etc.). MLVs are at disadvantage – SLVs have opportunity again with more focus into particular language.

# Last Mile Phenomenon



1. Outputs from MT engines are much closer to each other than to final quality human translation.



# Last Mile Phenomenon



2. Proximity phenomenon: the closer you are to the goal, the more subtle details are coming to stand out and still divide you and the prize. “Final 10% to the goal” is the Last Mile you need to cover with the largest effort and cost, deployment, preparation, process change and 100% editing.



# HYPOTHESIS and goals of the experiment



1. TEST HYPOTESIS – Is the Plateau of quality behind? How far is the next plateau of quality from Human Translation?  
**Yes, but we are about to hit another plateau, within usable domain. Human quality is still very far, the last mile gap is WIDE. Current predictions of machines passing Turing Test are 2029.**
2. Is the MT technology available for LSPs without multimillion funding?  
**Not multimillion, but continuous, significant effort is required.**
3. Is OpenSource SMT Moses a viable place to start?  
**YES and NO. For immediate deployments other commercial products may be more feasible. Moses is however very promising. A very good training ground to prepare for MT or evaluate current MT level.**
4. Can we base our own productivity solutions on Moses in any near future?  
**Timeframe is several years still. Numerous technical problems are to be solved.**
5. What are the possible LSP MT-related process implications?  
**We've got some answers to ourselves.**
6. What are the methods to calculate ROI?  
**We have developed our method.**
7. What are the post-editing differences in MT scenario?  
**Under development.**
8. Is my company capable of handling this?  
**YES.**



Thank you!

Questions?