D4.3: TRANSLATION PROJECT-LEVEL EVALUATION

Jinhua Du, Joss Moorkens, Ankit Srivastava, Mikołaj Lauer, Andy Way, Alfredo Maldonado, David Lewis

Distribution: Public

Federated Active Linguistic data CuratiON (FALCON)

FP7-ICT-2013-SME-DCA

Project no: 610879

Document Information

Deliverable number:	D4.3
Deliverable title:	**Translation Project level Performance, Usability and Reuse Efficacy Results**
Dissemination level:	PU
Contractual date of delivery:	31st August 2015
Actual date of delivery:	4th November 2015
Author(s):	Jinhua Du, Joss Moorkens, Ankit Srivastava, Mikołaj Lauer, Andy Way, Alfredo Maldonado, David Lewis
Participants:	DCU, TCD
Internal Reviewer:	Skawa
Workpackage:	WP4
Task Responsible:	T4.3
Workpackage Leader:	DCU

Revision History

Revision	Date	Author	Organization	Description
1	2/11/2015	Joss Moorkens, Jinhua Du, Ankit Srivastava, Andy Way	DCU	Draft
2	4/11/2015	Joss Moorkens, Mikolaj Lauer	DCU & XTM	Addition of interim DG-T report
3				
4				

CONTENTS

1. Executive Summary

This document summarises both the final optimisation of the machine translation (MT) systems and the final rounds of user evaluation conducted by the FALCON project. It reports on system testing using external resources, and the engineering that brought about a reduction in MT retraining time by 84% and a 95% reduction in disk space requirements for each MT engine. Thereafter, two user evaluations are described: one three-week evaluation at the project level by the European Commission's Directorate General of Translation, and one small-scale evaluation of the FALCON integrated terminology features.

2. Introduction

This deliverable reports on evaluations and optimisation of the FALCON statistical machine translation (SMT) process, and a second stage of human evaluations carried out using the FALCON portal following up on the work described in D4.2. Section 3 focuses on machine translation (MT) optimisation. As part of the work to maximise the functionality and robustness of the MT components, a great deal of effort was expended in attempting to reduce SMT engine retraining times and to minimise diskspace required in order to meet the operational requirements of running engines integrated with the multi-clinet XTM Cloud system. The dramatic improvements achieved in these efforts are reported in Section 3.1. The following section reports on tests using the automatic MT evaluation metrics BLEU [Papineni et al., 2002] and Translation Edit Rate [TER – Snover et al., 2006] to assess whether BabelNet [Navigli and Ponzetto, 2012] resources are useful as extra training data to augment SMT systems. This is followed by a report of further tests using BabelNet resources to alleviate the impact of out-of-vocabulary (OOV) words that appeared in the MT outputs.

The focus in Deliverable 4.2 was on testing the integration of the various components of the FALCON platform. In Section 4 of this deliverable, we report on evaluations carried out at the project level by translators and project managers at the European Commission's Directorate General of Translation (DG-T). The opportunity to work with the DG-T allowed us to carry out the planned tests of utility and qualitative assessments of the FALCON platform with the assistance of experienced translation staff who work as part of one of the largest translation services in the world. This necessitated a change from the crowd-sourced translation project planned for Task 4.3 and changes to the plan to evaluate the platform over several translation cycles. Instead, the platform was used over a period of three weeks to assist in the translation of live projects, as detailed in Section 4.1. Finally, Section 4.2 reports on a small-scale evaluation of the integration and functionality of terminological components, including IATE automatic term extraction, along with additions and edits to the Termweb dictionary component.

3. Machine translation optimisation

This section will focus on improvements to the MT retraining process and experiments to evaluate the utility (or otherwise) of the augmentation of MT resources using Babelnet.

3.1. Optimisation of SMT retraining

Regarding the post-editing-based SMT (PE-SMT), the incremental retraining can be roughly categorized into two different scenarios, namely the segment-level online incremental retraining (segment mode) and batch-level incremental retraining (batch mode). The former takes one post-edited segment per retraining cycle to immediately update the models, which requires rapid incremental processing of the word alignment, phrase/rule generation, language model and parameters tuning etc., while the latter firstly accumulates a batch of segments, and then performs the incremental retraining process to update the system. The batch-level mode can perform the incremental retraining process in the background while the translators/post-editors continue to work on the next batch of segments. From the point of view of parameter estimation, the former can promptly adapt its feature weights to the newly post-edited segment and learn the translator's knowledge, but the frequent change of weights might make the system unstable; the latter adapts the parameters on an average level of segments in a batch, which can keep the system relatively more robust, however, it cannot learn the knowledge as early as possible and cannot demonstrate a quick response to translator's practice and preference. In our system, we use the batch-level incremental retraining mode in order to keep the system more robust and in a reasonable range of being sensitive to the data change.

The incremental retraining framework for SMT is shown in Figure 1.

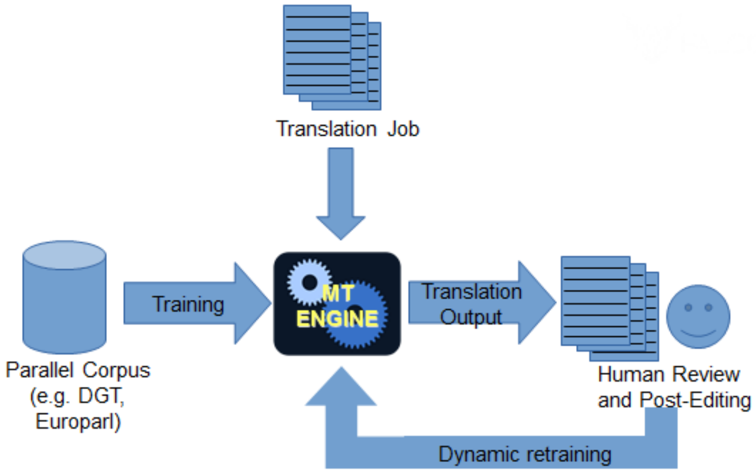


Figure 1 Incremental SMT retraining framework

The 'dynamic retraining' indicates that as long as the newly post-edited sentences are up to a threshold (500 sentences in our experiments), the retraining process will be automatically started.

Our retraining experiments simulate the post-editing job, that is, we use the target reference (instead of PEs) and approximate PE time with the TER metric, which is based on previous related work that found the the lower the TER score, the less the PE time.

The initial SMT system is trained using the initial training set and the model parameters are tuned on the devset. Then at the beginning, a batch of 500 source sentences (we call it "incremental sub-set") is selected sequentially from the incremental dataset and translations are obtained with the initial MT system. These translations are post-edited and the corrected translations are added to the training data. We then incrementally train a new MT system on the basis of previous training data. The updated model will be used to translate the next batch. The same process is repeated until the incremental dataset is finished. The workflow of the incrementally retrained experiments is shown in Figure 2.

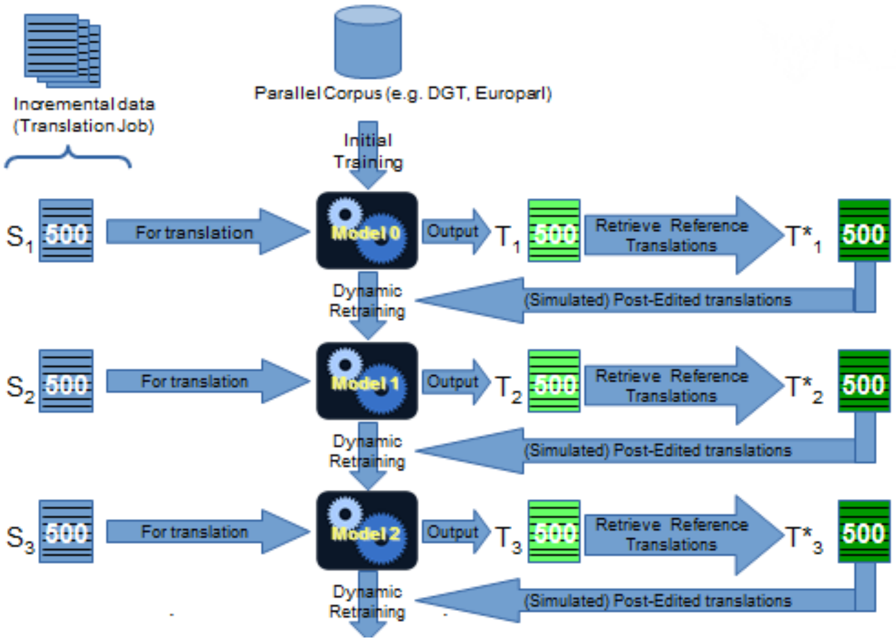


Figure 2 The workflow of the retraining experiments

The detailed description of the incrementally retrained experiments as to translation performance has been reported in the Deliverable 4.1, here we only discuss our retraining strategies in terms of the incremental word alignment.

In this project, we developed two incremental retraining strategies for the SMT systems in FALCON, namely 1) Moses-based incremental retraining framework, and 2) a combined strategy to combine cdec and Moses, where the second strategy can significantly speed up the incremental retraining process compared to the first strategy.

3.1.1. Strategy 1: Moses-based incremental retraining

The Moses-based incremental retraining strategy is shown in Figure 3

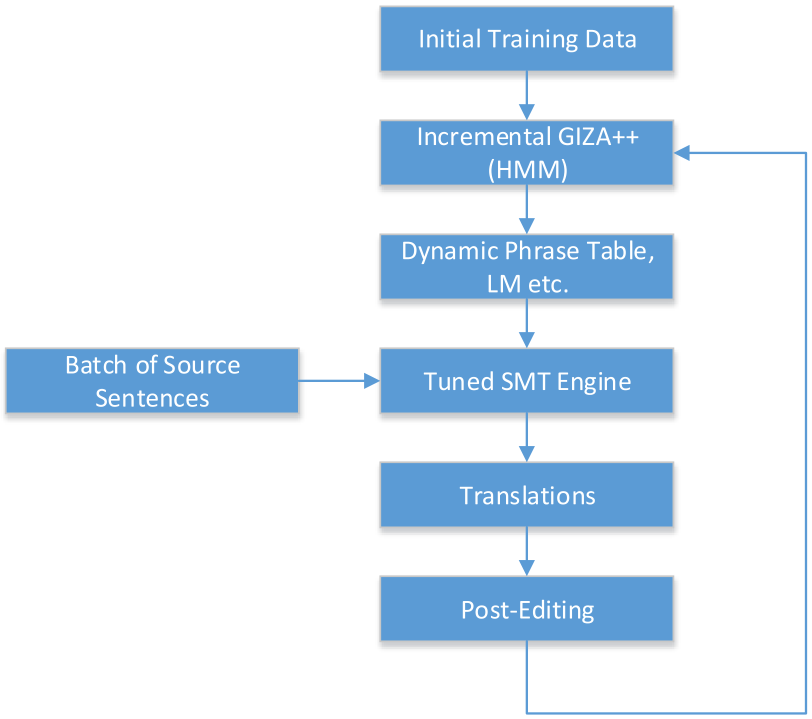


Figure 3 Moses-based incremental retraining workflow

In this strategy, the key steps to perform incremental retraining are:

1) Incremental GIZA++: during the word alignment for the initial training data, the alignment parameters are kept and will be used in the incremental alignment. The incremental GIZA++ can not only align the newly post-edited sentence pairs, but can update the alignment parameters. However, intuitively if the initial parallel data is very large, then the batch of 500 sentences would not have a big impact on the word alignment accuracy.

2) Dynamic Phrase table: it is a virtual phrase table based on sampling word-aligned bitexts and stored in the memory with suffix array. The dynamic phrase table can be quickly updated after the newly aligned bitexts are added to the old aligned bitexts, and the search for phrase pairs are very quick compared to the static phrase table.

We can see from this strategy that all components such as the word alignment, alignment symmetrization, model buildings are related to the Moses package.

In the experiments using the Moses-based strategy, we found that when the initial parallel data is large scale (e.g. 4 million), even the new data is only 500 sentences, the incremental word alignment will take more than 2 hours which is not applicable because the translators or post-editors would not like to wait for such a long time to carry on their work.

Based on the intuition we mentioned in 1), when the data set is very large, it is not necessary to update the old alignment parameters, we only need to use it to align the new data. Therefore, we propose a combined strategy which uses ‘fast_align’ and ‘force_align’ in cdec and dynamic phrase table in Moses to carry out the incremental retraining.

3.1.2. Strategy 2: Moses+cdec-based incremental retraining strategy

The Moses+cdec-based incremental retraining strategy is shown in Figure 4.

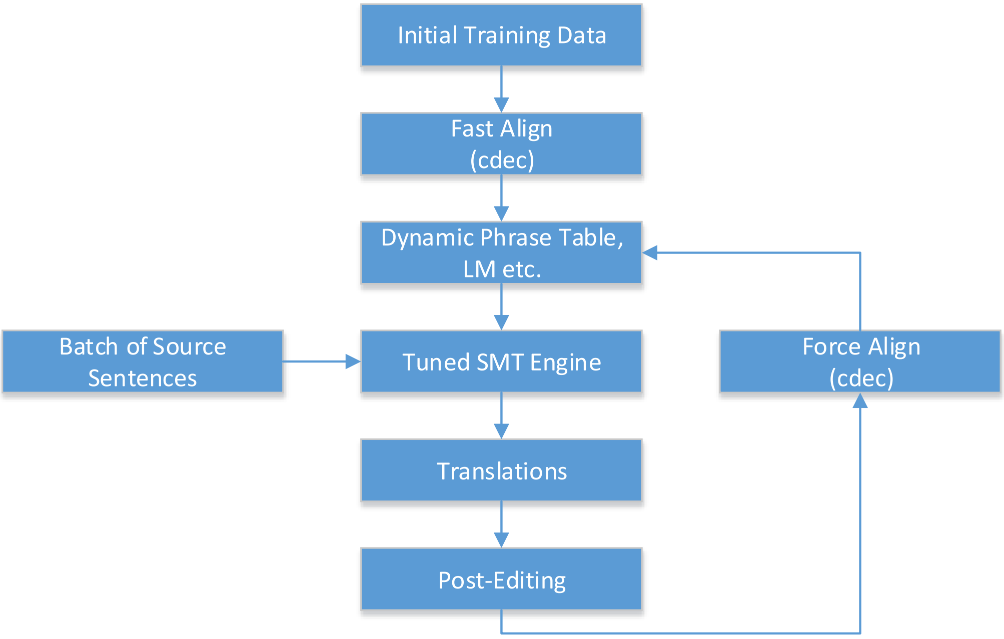


Figure 4 Moses+cdec-based incremental retraining workflow

In this strategy, the key steps to perform incremental retraining are:

1) Fast Align: the word alignment for the initial training data is performed using 'Fast Align' and the alignment parameters are kept for the word alignment of the new data;

2) Force Align: the new sentence pairs from the post-editing are aligned word by word using the stored alignment parameters of the initial training data. This step can save most of the time consumed in the incremental GIZA++ word alignment.

3) Dynamic Phrase Table: after obtaining the forward and inverse word alignment links from the fast align, we use cdec 'symmetrizer' to generate the final alignment links and transform it with the training bitexts into suffix array.

3.1.3. Comparison Experiments Using Two Strategies

We use English—German language pair to perform the experiment. The training data is from Europarl which contains 4,506,826 pairs, and devset and testset are from Newswire which are Newswire2012 and Newswire2013 respectively. The batch for the new data contains 500 sentences.

The main purpose of the experiment is to look at the time spent in the retraining process. The results are shown in Table 1.

Strategy	#Time Consumed	#Disk Space occupied
Moses-based	180 minutes	24G
Moses+cdec-based	***29*** minutes	1.1G

Table 1 Comparison of two incremental retraining strategies

We can see that:

1) The 'Moses+cdec-based' strategy reduced the time consumption of retraining to 1/6 of that of 'Moses-based' strategy.
2) The 'Moses+cdec-based' strategy reduced the disk space occupation to 1/24 of that of the 'Moses-based' strategy. Accordingly, the memory usage will be significantly reduced we infer.
3) From the viewpoint of practical application for the batch-model incremental retraining, 29 minutes for a 4.5 million scale SMT engine is acceptable.

3.1.4. Strategies to speed up the decoder

We also optimized the decoder to speed up the decoding process. Specifically,

1) Increasing the number of 'workers': Our decoder is an incremental retraining adapted decoder which uses the dynamic phrase table, i.e. generating a phrase by dynamically sampling the bitexts with alignment links in the memory. The number of sampling workers plays an important role in decoding time. The default number is 1, and we set it to 500 based on the settings of our machine.

2) Multiple treads decoding: our server running the decoder is a multi-core machine, so we set the number of threads of the decoder to 4 for each language pair. When the concurrent requests are very large, the multi-thread decoder can also process timely.

3) Decoding stack size: searching the best path in decoding is also time-consuming. In order to speed up, we set the stack size to 20 from the default value 100. By comparing the performance between these two settings, we found there is no significant decrease of system performance.

We tested the optimized SMT system on English-French WMT Newswire 2013 test set that contains 3,000 sentences. The average decoding time for the default system and the optimized system is shown in Table 2.

System	#Time (second)
Default	3.87
Optimized	2.56

Table 2 Average decoding time

The time in Table 2 includes: sending time from the client to the server, tokenization time, truecasing time, decoding time, detokenizing time, sending time from the server to the client. The EN-FR system is trained on Europarl data that contains 5.4 million pair of sentences.

3.2. Augmenting SMT training with Babelnet

In this Section, we investigate the use of BabelNet to augment SMT. Two different types of SMT systems are tested, namely the phrase-based (PB) and hierarchical phrase-based (HPB) systems, which are built on Europarl data sets. In addition, some optimization strategies are utilized to clean up the BabelNet dictionaries. Experimental results on English-Chinese and English-Polish language pairs show that BabelNet can augment performance of SMT systems as long as the data is clean and domain adapted.

3.2.1. Methodology

Two hardware platforms are used in the BabelNet experiments, which are:

1) DeskPC: it is a desktop PC with 4G memory and 4 CPU cores. It is mainly used to run "cdec" experiments. However, when the data is large, this PC is apparently not strong enough to load big models.
2) demo-cngl: it is a cluster in DCU containing many powerful servers. The latest 3.0 version of Moses was installed and it is mainly used for "Moses" experiments.

We used two standard automatic evaluation metrics for a fair and objective comparison. The metrics are as follows:

1) BLEU-4: We use the script "mteval-v13a.pl" to obtain the BLEU score;
2) TER: We use the script "tercom v6b.pl" to obtain the TER score.

3.2.2. Experiments

Europarl data for English–Polish pair

The language pair is English–Polish (En-Pl) that is from Europarl data and contains 518,155 sentence pairs for training, and 2,000 sentence pairs for devset and 2,000 sentence pairs for testset (one reference for each source sentence). The statistics of the data are shown in Table 3 and Table 4.

English – Training Data			Polish – Training Data		
#sen	#word	#entry	#sen	#word	#entry
518,155	11,270,214	52,247	518,155	9,743,192	144,146

Table 3 Statistics of Europarl En–Pl data for the model training

English – Test Set			Polish – Test Set		
#sen	#word	#entry	#sen	#word	#entry
2,000	47,194	4,063	2,000	39,956	7,451

Table 4 Statistics of Europarl En–Pl data for the test set

The results of different systems using Europarl En-Pl data are shown in Table 5.

Systems	Type	Machines	LM		BLEU4 (%)	TER (%)
			Type	Order		
Moses	PB	Demo-cngl	KenLM	5	24.86	58.37

Cdec	HPB	DeskPC	KenLM	5	***24.99***	***57.12***

Table 5 Results of different SMT systems on Europarl En-Pl data

It can be seen that "cedc" performs best on this data set. However, the difference between the "CDEC" and "Moses" are not significant.

Europarl+BabelNet data for English-Polish pair

The data includes Europarl (the same as in Section 2.3) and BabelNet dictionaries (extracted from BabelNet and they are raw data without any clean-up) for English–Polish pair. The entries in the BabelNet dictionaries are added into the training data in this experiment.[1] Statistics of the training data are shown in Table 6. The test set is the same as in Table 4.

English – Training Data			Polish – Training Data		
#sen	#word	#entry	#sen	#word	#entry
6,714,558	26,333,229	943,611	6,714,558	23,095,319	675,661

Table 6 Statistics of Europarl+BabelNet En–Pl data for the model training

The results of different systems usingEuroparl+BabelNetEn-Pl data are shown in Table 7.

Systems	Type	Machines	LM		BLEU4 (%)	TER (%)
			Type	Order		
Moses	PB	Demo-cngl	KenLM	5	24.67	58.67
cdec	HPB	DeskPC	KenLM	5	***24.64***	***57.98***

Table 7 Results of different SMT systems on Europarl+BabelNet En-Pl data

It can be seen that:

1) Both systems perform worse than the corresponding system on "Europarl" data set, which indicates that adding raw BabelNet data into Europarl data is not helpful in improving system performance. Furthermore, it significantly increases the training time and decoding time as well as memory usage.
2) Moses performs better than cdec in terms of BLEU score, but worse in terms of TER score.

Europarl+BabelNet cleaned data for English-Polish pair

We did some pre-processing for the EN–PL BabelNet dictionaries in this section as follows:

[1] The BabelNet data is only repeated once when added into the Europarl data.

[2] We use BabelNet 2.5.1 API in http://babelnet.org/download.

1) if the EastAsian characters are included in either English or Polish side, we remove thispair;
2) if the English side contains some symbols which are not letters, digits, then we removethis pair;
3) if the Polish side contain the punctuations, then we remove this pair;
4) if the English side is the same as the Polish side, we remove this pair;
5) if the character length (Bytes) of the string in each side is less than 3, then we remove thispair;
6) if the ratio of the sentence lengths (word-level) between the English side and the Polishside is in (0.5, 2), then we keep this pair, otherwise, we remove it. This rule is based onthe fact that 99% pairs of sentences in Europarl EN-PL fall in this range.

After the clean-up, we obtain2,215,248 pairs from the original 6,199,234 pairs in the EN–PL dictionaries. We can see that almost 2/3 was removed. The statistics of the cleaned EN–PL dictionaries are appended to the Europarl data are shown in Table 8.

English – Training Data			Polish – Training Data		
#sen	#word	#entry	#sen	#word	#entry
2,733,403	15,697,254	462,826	2,733,403	13,995,654	438,283

Table 8 Statistics of Europarl+BabelNet cleaned data for the model training

The results of different systems using Europarl+BabelNet cleaned data are shown in Table 9.

Systems	Type	Machines	LM		BLEU4 (%)	TER (%)
			Type	Order		
Moses	PB	Demo-cngl	KenLM	5	24.30	58.92
cdec	HPB	DeskPC	KenLM	5	***24.63***	***57.91***

Table 9 Results of different SMT systems on Europarl+BabelNet cleaned data

It can be seen that the cleaned data did not increase system performance on Moses and cdec. Even if we removed a larger part of the dictionary pairs, the performance does not decrease too much which shows that the BabelNet dictionary indeed contains too much noise or its domain is much different from the Europarl data so that it can't contribute to the system.

FBIS Data for Chinese–English Pair

The language pair used here is Chinese–English (Zh-En). The training data comes from NIST FBIS that contains 270,794 sentence pairs, the devset is NIST 2006 current set that includes 1,664 sentences with 4 references for each, and the testset is NIST 2005 current set that contains 1,082 sentences with 4 references for each. The statistics of the data are shown in Table 10 and Table 11.

Chinese – Training Data			English – Training Data		
#sen	#word	#entry	#sen	#word	#entry
270,794	9,582,189	102,035	270,794	10,319,019	81,036

Table 10 Statistics of FBIS Zh–En data for the model training

Chinese – Test Set			English – Test Set		
#sen	#word	#entry	#sen	#word	#entry
1,082	30,489	5,684	1,082	142,794	7,552

Table 11 Statistics of FBIS Zh–En data for the test set

The results of different systems usingFBIS Zh–Endata are shown in Table 12.

Systems	Type	Machines	LM		BLEU4 (%)	TER (%)
			Type	Order		
Moses	PB	Demo-cngl	KenLM	5	***28.30***	***66.11***
cdec	HPB	DeskPC	KenLM	5	27.49	69.01

Table 12 Results of different SMT systems on FBIS Zh–En data

We can see that Moses performs better than the cdec system in this task.

FBIS+BabelNet for Chinese–English Pair

We append word/phrase pairs from the Chinese–English dictionaries of BabelNet to the initialFBISdata. These pairs are repeated only once. We did some pre-processing for the BabelNet Chinese–English dictionaries as follows:

1) The original word/phrase pairs are full of noises and many Chinese characters are encoded as UTF-8 Traditional format (BIG5), so we have to convert them to UTF-8 Simplified format (GBK) first and then clean up the noises as possible as we can. The encoding conversion tool we used is “ConvertZ”.
2) We split the Chinese side and English side based on the bound symbol “” into two individual parts for next processing.

3) We use two simple rules to clean up the data. The first one is that the pair without any Chinese character in the Chinese side will be removed, and the second one is that the pair containing other symbols except the digits, letters in the English side will be removed.
4) We use “ICTCLAS” segmentor to segment Chinese side of the dictionary, and then lowercase the English words in the Chinese side.

   Finally, we obtain 5,501,451Chinese–English pair from the original 5,975,619 pairs. Table 13 shows the statistics of the new training data. The testset is same as in Table 9.

Chinese – Training Data			English – Training Data		
#sen	#word	#entry	#sen	#word	#entry
5,501,451	24,928,462	278,302	5,764,117	24,381,754	660,836

Table 13 Statistics of FBIS+BabelNet Zh–En data for the model training

The results of different systems usingFBIS+BabelNet Zh–Endata are shown in Table 14.

Systems	Type	Machines	LM		BLEU4 (%)	TER (%)
			Type	Order		
Moses	PB	Demo-cngl	KenLM	5	***28.36***	***65.47***
cdec	HPB	DeskPC	KenLM	5	26.76	70.28

Table 14 Results of different SMT systems on FBIS Zh–En data

It can be seen that:

1) Moses at the first row in Table 12 improves a little compared to Moses with the same set-up in Table 10 in terms of BLEU score, but much decrease in terms of TER score. Althoughthe improvements are not significant, the results show the effectiveness of BabelNet for phrase-based SMT if the data is clean.
2) cdec at the bottom row in Table 12 performs significantly worse than the same system in Table 10 in terms of BLEU and TER scores, which indicate that even the cleaned BabelNet data is not helpful for HPB system. We think that the HPB rules are non-terminal templates, while the data in BabelNet are phrases or words, so it might have some problems when extracting rules.

3.3. Augmenting SMT output with Babelnet

In this Section, we use BabelNet[2] in the post-processing stage to verfiy its impact on the OOVs in the translation results. The statistics of OOVs in the WMT Newswire2013 test set in terms of Europarl data set and the DGT data set for every language pairs in our experiments are shown in Table 15 and 16, respectively.

Language Pair	#OOVs	#Trans. Of OOVs	Ratio (%)
EN-DE	887	155	17.47
EN-FR	737	96	13.03
EN-ES	771	103	13.36
EN-FI	1676	693	41.35

Table 15 Statistics of OOVs and the translations by BabelNet on Europarl data set

In Table 15, the first column '#OOVs' indicates the number of OOVs (including duplicates) in the test set in terms of the training data, and the second column '#Trans. Of OOVs' indicates the number of OOVs that can be translated by the BabelNet API, and the last column show the ratio of (#Trans. Of OOVs)/(#OOVs).

Language Pair	#OOVs	#Trans. Of OOVs	Ratio (%)
EN-FR	3,229	1,515	46.92
EN-ES	3,256	1,417	43.52

Table 16 Statistics of OOVs and the translations by BabelNet on DGT data set

From Table 15 and 16 we can see that only part of the OOVs can be retrieved and translated by calling the BabelNet API. The ratios of handling the OOVs on DGT data set is higher than those on Europarl data.

The comparative results of Europarl and DGT data sets are shown in Table 15 and 16, respectively.

Language Pair	OOVs Contained		OOVs Removed		OOVs Processed	
	BLEU4	TER	BLEU4	TER	BLEU4	TER
EN-DE	11.54	75.93	11.50	75.58	11.55	75.66
EN-FR	20.46	67.05	20.65	66.85	20.60	66.86

[2] We use BabelNet 2.5.1 API in http://babelnet.org/download.

EN-ES	23.75	64.07	23.90	63.90	23.92	63.90
EN-FI	5.11	91.98	5.53	87.70	5.34	90.24

Table 17 Results of using BabelNet on Europarl data

In Table 17, the language pairs are English-German (EN-DE), English-French (EN-FR), English-Spanish (EN-ES) and English-Finish (EN-FI) respectively. ‘OOVs Contained’ indicates the OOVs in the source sentences are kept to the translations during the decoding, and ‘OOVs Removed’ indicates the OOVs in the source sentences are automatically removed in the translations during the decoding, and ‘OOVs Processed’ indicates that the OOVs in the translations are processed usbing BabelNet resources. All the results are evaluted in Truecased format.

Language Pair	OOVs Contained		OOVs Removed		OOVs Processed	
	BLEU4	TER	BLEU4	TER	BLEU4	TER
EN-FR	4.20	1.05	4.38	1.01	4.28	1.04
EN-ES	11.07	74.20	10.95	73.44	11.22	73.60

Table 18 Results of using BabelNet on DGT data

The test set to evaluate the OOVs on Europarl data and DGT data is the WMT Newswire2013 current set for EN-DE, EN-FR and EN-ES. As to En-FI, we use the WMT2015 EN-FI test set as our test set.

From Table 17 and 18 we can see that in all cases, the BLEU and TER scores in terms of ‘OOVs Processed’ are higher than those of ‘OOV Contained’, which show that the BabelNet is useful to handle the OOVs issue in SMT. Further experiments and investigation will be carried out in future.

4. Project-level human evaluation

4.1. Qualitative results of project-level evaluation with the Directorate-General for Translation

In fulfilment of Task 4.3, the FALCON platform was due to be tested at a project level via crowd-sourcing. However, when the opportunity arose to run a project-level evaluation with the European Commission’s Directorate General of Translation (DG-T), it was decided to proceed with this instead, despite the extra lead time required and the specific translation requirements within the DG-T.

In preparation for this evaluation, two training sessions for participants took place. The management of the DG-T Information Technology group and their selected translator participants took part in online training sessions organised by XTM on September 2nd 2015, and on September 4th 2015. The first session was dedicated to presenting the FALCON interface with a comprehensive explanation of its various features and applications. The second session focused mainly on the SlimView component and on TermWeb integration.

The online training sessions were followed by a one-day on-site presentation of Falcon in Luxembourg on September 17th, when the integration with Easyling/SlimView and TermWeb was again explained in detail. The evaluation of Falcon with SlimView was planned to commence on October 15th 2015, and the evaluation participants were sent detailed instructions on how to work with the system. The actual testing ran from October 16th to October 30th inclusive, during which time seven participants worked on two different UK English to French website translation projects. Project FR1 contained 151 segments to translate and FR2 contained 1113 segments. Feedback was gathered from the DG-T participants using surveys and interviews regarding their impression of the evaluation.

One of the drawbacks that became apparent from the feedback following the evaluation was that the language pair – stipulated by the DG-T – was not necessarily suitable for the evaluation participants. Participant 1, for example, said that “it would have been much more useful if I was able to test in my usual target language: Bulgarian”. Other participants are native speakers of Finnish, Danish, Swedish, Russian, and Dutch. While this meant that participants could test the functionality of the platform, there were elements that they did not feel able to evaluate. One participant said that he could not give an opinion on the quality of the SMT “since the MT was in French which is by no means my strongest language”.

Participants had mixed feelings about using a cloud-based tool, as “on one hand I like the idea of a cloud-based tool that can constantly be updated and improved”, however, the participant feels that “cloud-based solutions always have issues with speed and then there's the problem with slow or no internet connection that might render the tool useless in times of connectivity issues”.

The main criticism of the platform from the participants was to do with speed when working with Slimview. One participant had to regularly refresh his Slimview browser window, and “loading everything in order to start to translate took a lot of time”. While this problem did not occur in other tests, we have since begun work to recreate and improve any lag in performance. Another participant commented that “segment versioning is a really nice feature”, although he would like the addition of an overall change tracker for review purposes. The overarching view of participants surveyed thus far is that the platform works well, but requires some further testing and development in order to help translators to create a “faster or better translation”.

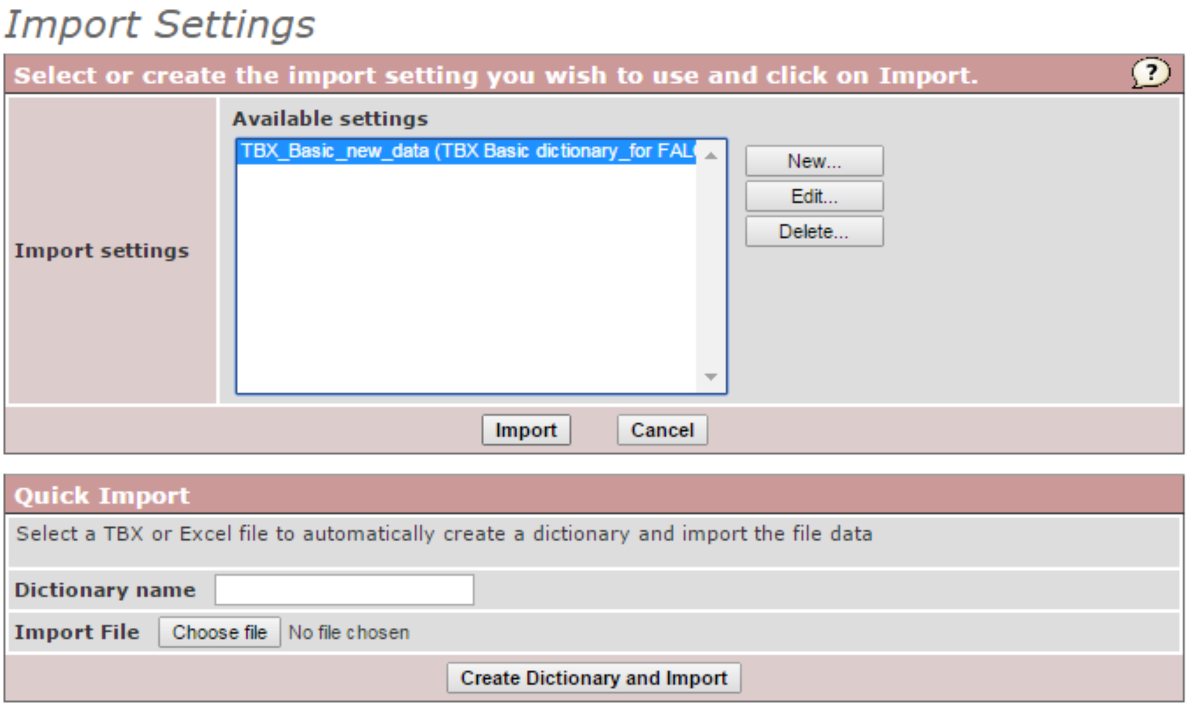
Further input is being collected and analysed an will be provided to an update of this document.

4.2. Human evaluation of terminology features

Following on from the evaluation of system usability and re-use efficacy in Task 4.2, several participants from that task were invited to complete a translation task focussing specifically on the functionality and efficacy of the terminology tools integrated in the FALCON platform. Four participants, with the language pairs of English-Spanish and English-French, agreed to take part in this short study.

4.2.1. Evaluation methodology

Each participant was assigned an 1800-word (+/-5%) task to post-edit, from which 50-60 terms had been automatically extracted within the FALCON platform. The participants were asked to fill in target terms on a spreadsheet downloaded from FALCON with the help of the source text, and then to upload the completed spreadsheet using the upload dialog as seen in Figure 5. The text to post-edit was a technical text taken from the CNGL Centre for Global Intelligent Content annual report.



Import Settings

Select or create the import setting you wish to use and click on Import.

Import settings

Available settings

TBX_Basic_new_data (TBX Basic dictionary_for FAL

Quick Import

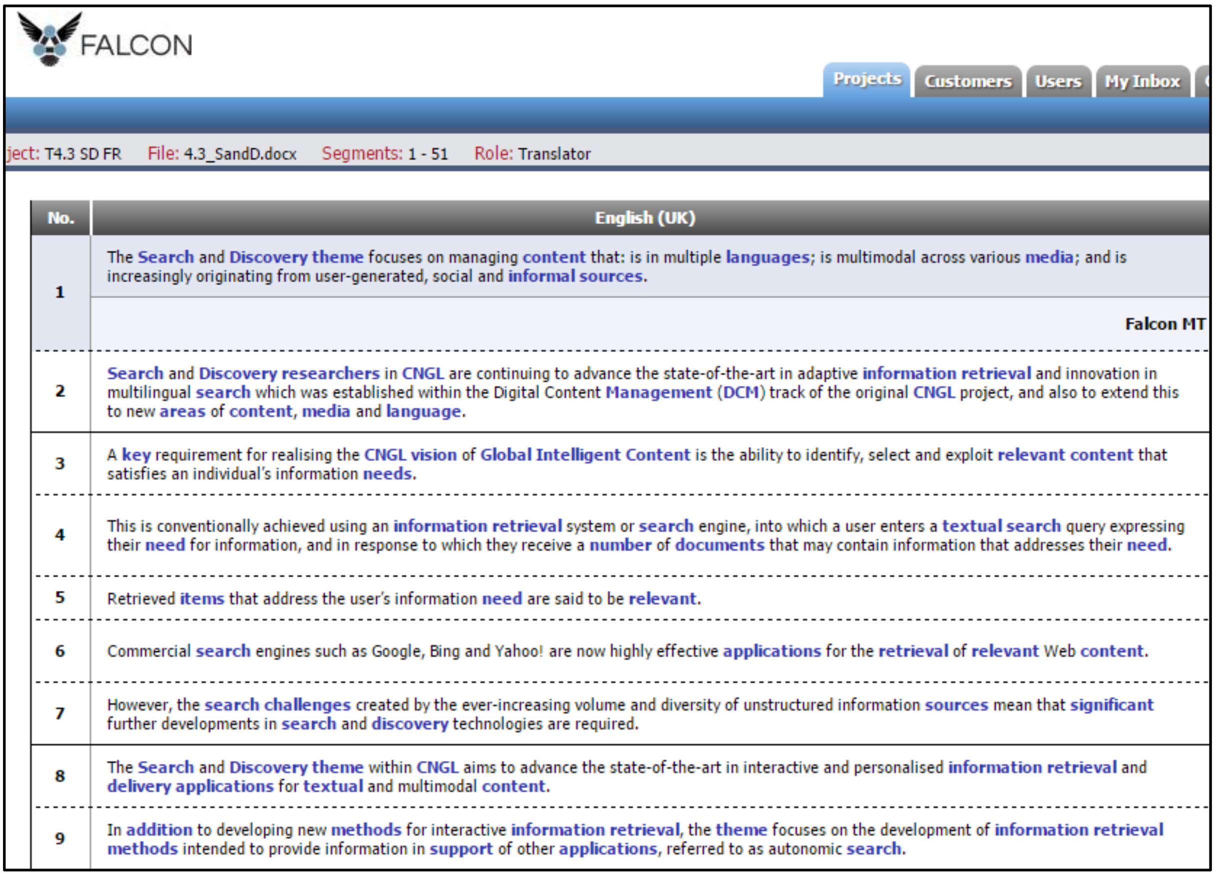
Select a TBX or Excel file to automatically create a dictionary and import the file data

Dictionary name

Import File Choose file No file chosen

Figure 5 FALCON term upload dialog

Participants were requested to post-edit the text within the FALCON/XTM editor. The uploaded terminology was highlighted in the source text as shown in Figure 6.



ject: T4.3 SD FR File: 4.3_SandD.docx Segments: 1 - 51 Role: Translator

No.	English (UK)
1	The Search and Discovery theme focuses on managing content that: is in multiple languages; is multimodal across various media; and is increasingly originating from user-generated, social and informal sources. Falcon MT
2	Search and Discovery researchers in CNGL are continuing to advance the state-of-the-art in adaptive information retrieval and innovation in multilingual search which was established within the Digital Content Management (DCM) track of the original CNGL project, and also to extend this to new areas of content, media and language.
3	A key requirement for realising the CNGL vision of Global Intelligent Content is the ability to identify, select and exploit relevant content that satisfies an individual's information needs.
4	This is conventionally achieved using an information retrieval system or search engine, into which a user enters a textual search query expressing their need for information, and in response to which they receive a number of documents that may contain information that addresses their need.
5	Retrieved items that address the user's information need are said to be relevant.
6	Commercial search engines such as Google, Bing and Yahoo! are now highly effective applications for the retrieval of relevant Web content.
7	However, the search challenges created by the ever-increasing volume and diversity of unstructured information sources mean that significant further developments in search and discovery technologies are required.
8	The Search and Discovery theme within CNGL aims to advance the state-of-the-art in interactive and personalised information retrieval and delivery applications for textual and multimodal content.
9	In addition to developing new methods for interactive information retrieval, the theme focuses on the development of information retrieval methods intended to provide information in support of other applications, referred to as autonomic search.

Figure 6 Automatically extracted terms highlighted in the FALCON UI

Participants were requested to make some changes to the termbase during the course of their post-editing task. They were asked to:

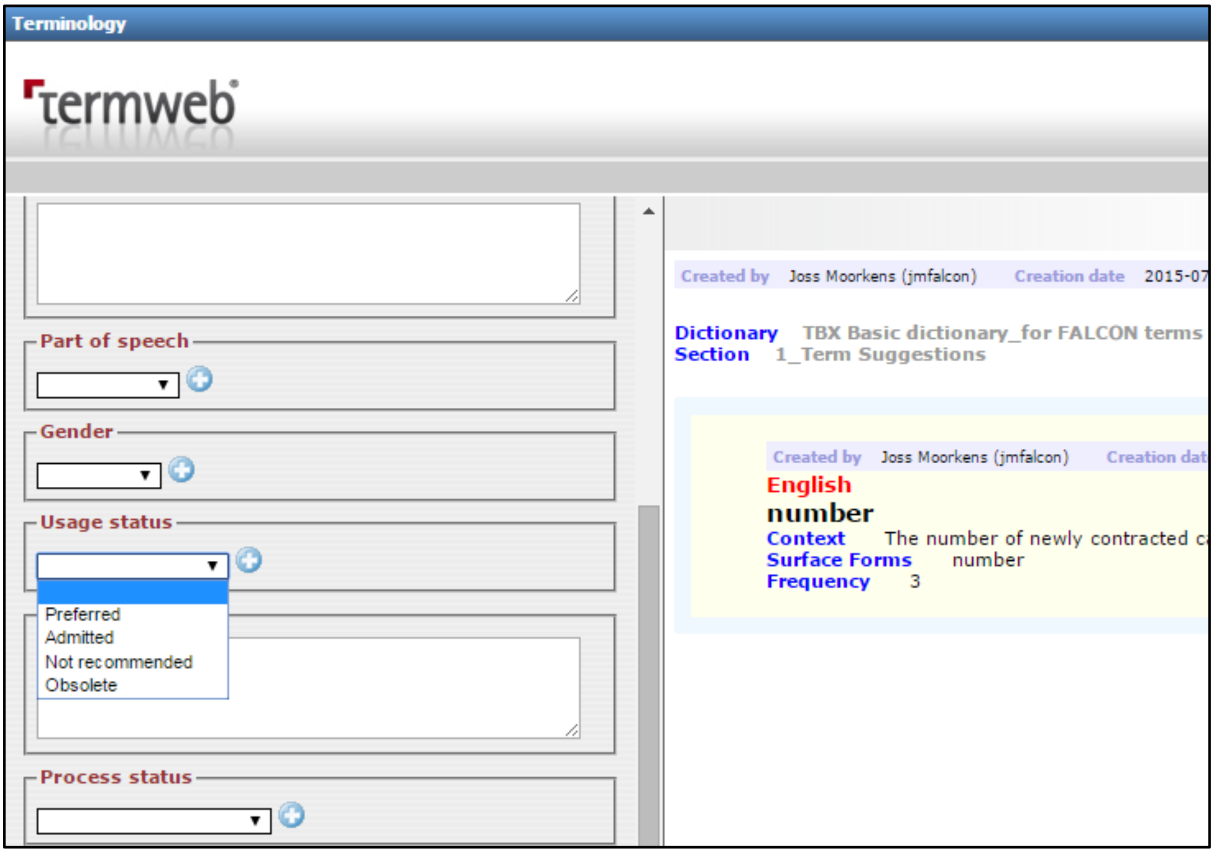
- Change the status of a highlighted term. (i.e. change Usage status to "Preferred", "Admitted", "Not Recommended", or "Obsolete")
- Suggest a new term that was not suggested by the terminology extraction mechanism
- Ignore or delete unwanted terms

Once the task had been completed, participants were asked to respond to a short survey to evaluate their experience of this task.

4.2.2. Evaluation findings

Participants responded very positively about the benefits of the integrated terminology features, writing that they would help them to "perform translations in less time with more accuracy" (User ES3). User ES2 wrote that the features "helped in terms of consistency and allowed me to work more efficiently and to speed up the post-editing task". One participant (ES1), who was happy with Termweb as a standalone tool, appeared unclear about the value of the automated term extraction, and wrote "I hate automated translation". She disliked having to complete an exercise in post-editing, and did not engage with the task to the same extent as the other participants, who were enthusiastic about automatic extraction. For example, ES2 welcomed this addition, adding that "building up a terminology database can be a lengthy and tedious process, but it ultimately speeds up the translation/editing task". Two participants said that this exercise improved their opinion of automated term extraction. ES3 wrote: "When the correct/accurate term was not found, I was able to update it accordingly, which in turn improved the results at a later stage, and I believe will continue to do so in subsequent translations".

Out of the three participants who engaged fully, 43% of terms were marked Preferred in the status window (see Figure 7), 37% were marked Admitted, and 20% were either marked Obsolete or deleted completely. One user found it easiest to keep the terminology tab open in a separate browser, and wrote that making a status change was "quite straightforward, albeit a bit laborious" (ES2).



Created by Joss Moorkens (jmfalcon) Creation date 2015-07

Dictionary TBX Basic dictionary_for FALCON terms
Section 1_Term Suggestions

Created by Joss Moorkens (jmfalcon) Creation dat

English
number
Context The number of newly contracted c
Surface Forms number
Frequency 3

Figure 7 FALCON/Termweb status window

Participants made an average of seven additions to the termbase. They said that they found this operation "really simple" (FR1), although one participant found that "even though the term was added straightaway, there was a slight delay updating the translated segment" (ES2). This delay appears to have been caused by performing operations in two browser windows, and caused her to re-enter one term before she realised that she needed to refresh the Editor window. This was one of two problems that participants encountered. ES3 wrote that "sometimes the system wouldn't register the changes at all and I had to go back and do it again, which made me lose time". Following some investigation of this issue, it may be due to reduced Termweb permissions, required to isolate this study from the concurrent study in the DG-T.

The three participants who engaged fully with the task all said that the terminology features were effective in reducing post-editing effort. Two participants said that the integrated terminology features were novel and far more user-friendly than in their current translation workflow. ES2 wrote:

> I particularly liked the possibility of searching for a term in both source and target language, as well as being able to make changes to both. This is a feature that my usual translation platform does not provide, and I found it especially helpful. The search option is not case-sensitive, being a major time saver. The editing window allows me to enter information regarding grammar and context (part of speech, gender,

term type) not just the word definition, also very useful in terms of accuracy.

Overall, participants said that the terminology tool is “really beneficial” (FR1) and “provides the translator with a great management tool which allows to perform translations in less time with more accuracy”.

5. References

Navigli, Roberto, Simone Paulo Ponzetto. 2012. BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. Artificial Intelligence 193, 217-250.

Papineni, Kishore, Slaim Roukos, T.Ward, and W.J. Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In Proceedings of Association for Computational Linguistic (ACL 2002), 311–318.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In Proceedings of Association for Machine Translation in the Americas (AMTA 2006), 223–231.