Project Adaptation for MT-Enhanced Computer Assisted Translation

Mauro Cettolo **Nicola Bertoldi** **Marcello Federico**

FBK - Fondazione Bruno Kessler
via Sommarive 18
38123 Povo, Trento, Italy
{cettolo,bertoldi,federico}@fbk.eu

Abstract

The effective integration of MT technology into CAT tools is a challenging topic both for academic research and the translation industry. Particularly, professional translators feel crucial the ability of MT systems to adapt to their feedback. In this paper, we propose an adaptation scheme to tune a statistical MT system to a translation project using small amounts of post-edited texts. By running field tests on two domains with 8 professional translators working with a CAT tool, productivity gains up to over 20% were measured after applying MT project adaptation.

1 Introduction

Computer-assisted translation (CAT) is an important frontier where current MT technology meets professional translators. While MT is generally not yet able to provide output that is suitable for publication without human intervention, CAT is an ideal scenario, because human feedback is always available. In fact, while it has been reported several times that translators can improve their productivity by post-editing MT output, to our knowledge little work has been done to show how MT can benefit from human post-editing. From the viewpoint of professional translators, refinement of the SMT system in response to their corrections is indeed perceived as crucial in order to improve the usability of MT and to further increase productivity and quality of post-editing. Simply stated, while it is acceptable that MT makes mistakes, it is less acceptable that MT does not learn from user corrections.

This paper presents recent results from the MateCat project,[1] which is developing a Web-based CAT tool for professional translators that integrates new MT functions. In particular, we report here on the self-tuning MT feature, that incrementally updates the MT engine by exploiting user post-edits collected during the life of a translation project.[2] Before starting, the MT engine is optimized on the domain of the project. Then, after a day of work by a human translator, knowledge about the newly translated text and user corrections is injected into the SMT system so that, hopefully, improved translations will be proposed the next day. This procedure is continued until the end of the project.

The approach has been validated in laboratory and with two-day field tests, involving the translation of English documents into Italian in two domains, Information Technology (IT) and Legal. The IT and Legal domains represent relevant sectors in the translation industry and are suitable for exploiting statistical MT, since the information source is sufficiently homogeneous, the language is sufficiently complex, and there is sufficient multilingual data available to train and tune MT systems.

Lab tests were performed by comparing different variants of our systems and measuring progress on well-defined development and test sets. Field tests were run with the MateCat tool over two days to compare productivity of human translators before and after adapting the MT systems.

Remarkable improvements in terms of automatic MT metrics were observed in the lab test experiments. These results were also confirmed by the field tests, where significant productivity gains

[1] www.matecat.com

[2] By translation project we mean a set of homogeneous documents assigned to the translator.

Sima'an, K., Forcada, M.L., Grasmick, D., Depraetere, H., Way, A. (eds.)
Proceedings of the XIV Machine Translation Summit (Nice, September 2–6, 2013), p. 27–34.

were measured. In particular, translations speed of the translators increased on average by 11.2% in the IT domain and 22.2% in the Legal domain, while the post-editing effort improved by 6.5% and 10.7%, respectively.

The paper is organized as follows. Section 2 sketches the methods utilized for project adaptation. In Section 3, the data employed in experiments are introduced and analyzed. The complete adaptation scheme and lab test experiments are described in Section 4, while Section 5 is devoted to present the field test experiments. Some final summarizing comments conclude the paper.

2 Adaptation Methods

In this section we describe the techniques employed to adapt our SMT systems.

2.1 Data selection

Data selection is a problem widely investigated by the SMT community, see for example (Yasuda et al., 2008; Matsoukas et al., 2009; Foster et al., 2010; Axelrod et al., 2011). In fact, very often, data available to train MT models come from different sources that may be heterogeneous with respect to size, quality, domain, production period, etc. Data selection is supposed to pick out a subset of training data that is mostly relevant to the task at hand. In our case, we are interested in selecting data that is relevant to a specific translation project. Practically, model features extracted from the most specific data are combined with those extracted from the remaining training data, in a way to optimizes the trade-off between accuracy and coverage of the models.

We reimplemented the data selection technique by Moore at Lewis (2010) and made it publicly available through the IRSTLM toolkit (Federico et al., 2008). The algorithm can be applied to select both monolingual and parallel data.

To apply the algorithm we need a *foreground corpus*, which represents the addressed task, and a *background corpus*, which is much larger and task independent. The first step consists in creating two language models, one foreground and one background, used to compute a single score for each sentence of the background corpus. This score is the difference between the cross-entropy calculated with the foreground LM and the cross-entropy calculated with the background LM. The background sentences are then ordered according to this score. The selection of useful sentences from the background corpus is achieved by determining the best splitting point of its sorted version. The estimation of the optimal split is performed by minimizing the perplexity of a development set on growing percentages of the sorted corpus. The set of sentences used to train the model with the lowest perplexity are finally selected.

In all our experiments, we have observed the same behaviour which was reported by Moore and Lewis (2010): the perplexity decreases when less, but more appropriate data is used, reaching a minimum between 10 to 20% of the background data. As a positive side effect, the models become considerably smaller which is an important aspect when deploying SMT systems in real applications.

Notice that in our case the selection of parallel text was done by considering only one side of the bitext.

2.2 Fill-up for phrase-based SMT adaptation

Given the scarcity of parallel linguistic resources, in SMT training, the need of combining data of parallel corpora of different sizes and content is rather typical. One way to face this combination issue is the *fill-up* technique, initially proposed by Nakov (2008) and then refined by Bisazza et al. (2011). Fill-up effectively exploits background knowledge to improve translation and distortion models coverage, while preserving the more reliable information coming from the foreground corpus. In practice, the background phrase table is merged with the foreground phrase table by adding only phrase pairs that do not appear in the foreground table. While performing at least as good as other popular adaptation techniques (Niehues and Waibel, 2012), fill-up approach builds models that are more compact and easier to tune by means of the minimum error rate training procedure.

2.3 Mixture LMs

As concerns the LM adaptation, we employed the mixture of LMs since it is a well-established and good-performing method. The mixture model can be used to combine one or more background LMs with a foreground LM representing new features of the language we want to include (Federico and Bertoldi, 2004). The technique consists of the convex combination of the LMs; the mixture weights are estimated on the training data of the foreground LM by applying a cross-validation scheme that simulates the occurrence of new n-grams. The method is available in the IRSTLM toolkit (Federico et al., 2008).

3 Data for Development

For our experiments we relied on existing language resources, including parallel corpora and translation memories. For the IT domain, in addition to small publicly available corpora, proprietary data sets were employed (software documentation in general). For the Legal domain, the publicly available JRC-Acquis collection (Steinberger et al., 2006) was used, which mostly includes EU legislative texts translated into 22 languages. More details are provided in the next sections.

3.1 IT domain

Most of text corpora for this domain were provided by the industrial partner of the MateCat project. In particular, we employed the following resources:

- Translation Memory (TM): a large collection of parallel entries used for training purposes; it mostly consists of real projects commissioned by a specific customer;
- Software manuals from the OPUS corpus (Tiedemann, 2012), namely KDE4, KDE4-GB, KDEdoc, and PHP; they are used for training purposes;
- Generic IT projects (IT-docs): parallel documents coming from six real projects commissioned by various customers; they are intended to be representative of the IT domain and then used for development/investigation purposes;
- Customer-specific project (TM-prjct): a parallel document from the specific customer over-represented in the TM; it is used for development/investigation purposes as well.

Statistics of these corpora are reported in Table 1. From here on, we will refer the union of TM and OPUS corpora simply as TM.

3.2 Legal domain

The JRC-Acquis corpus[3] has been used for both training and evaluation purposes. The collection is provided with two different alignments at segment level computed by means of free tools. A preliminary investigation suggested to re-align it by using the Gargantua software (Braune and Fraser, 2010). From the original corpus, a document has been selected for development/evaluation purposes, of adequate size and member of a not too large nor too small Eurovoc[4] subject domain class, namely the 4040. To be fair, all documents associated with this class have been removed from the training data. The document has been split into two blocks, one used for weight tuning/data selection purposes (`dev`), the other for evaluation (`test`). Table 2 provides some statistics on these parallel texts.

		#seg	#src wrds	#tgt wrds
TM+OPUS				
all entries (wd)		5.5M	63.8M	66.6M
no duplicates (wod)		1.9M	27.8M	29.0M
IT-docs		4.1k	56.0k	60.5k
	data-sel	1.4k	18.0k	19.3k
	dev	1.4k	21.1k	22.9k
	test	1.4k	16.9k	18.3k
TM-prjct		1.8k	18.0k	18.7k
	dev	800	5.1k	5.4k
	test	989	12.9k	13.3k

Table 1: Overall statistics on English–Italian parallel data of the IT domain used for training and testing the SMT system. Counts of (English) source words and (Italian) target words refer to tokenized texts. Symbols k and M stand for 10^3 and 10^6, respectively. For the meaning of `wd` and `wod` (with/without duplicates) see Section 3.3.

		#seg	#src wrds	#tgt wrds
Acquis	(wod)	1.5M	47.6M	49.3M
Legal-prjct		769	23.2k	24.2k
	dev	290	10.0k	10.6k
	test	479	13.2k	13.6k

Table 2: Overall statistics on English-Italian parallel data of the Legal domain used for training and testing the SMT system.

3.3 Data analysis

Before performing experiments, we analyzed data at disposal. We focused on IT corpora, since Legal training, development and test sets are expected to be homogeneous coming from the same source.

Table 3 shows perplexity (PP) and out-of-vocabulary rate (OOV) of the union of the six IT-docs and of TM-prjct computed on 6-gram LMs smoothed via the Kneser-Ney method and estimated on some variants of the TM. The first two columns compare the training on the whole TM

[3] optima.jrc.it/Acquis/JRC-Acquis.3.0/corpus/

[4] eurovoc.europa.eu

PP/%OOV	TMwd	TMwod	TMwod+$\overline{\text{prjct}_i}$	closeTMwod	farTMwod
IT-docs	618/1.93	575/1.93	492/1.63	469/2.79	1031/2.78
TM-prjct	151/0.55	143/0.55	142/0.55	305/2.28	152/0.57

Table 3: Perplexity (PP) and out-of-vocabulary rate (OOV) of project texts (target side) over 6-gram LMs estimated on some variants of the TM; see text for details.

target text (`TMwd`) and on the same text after the removal of source-target duplicates (`TMwod`). As the deletion of duplicate entries yields a 5-7% relative reduction of PP, all the experiments presented in the rest of the paper involved the TM without duplicates. Anyway, it can be noted that the PP of IT-docs is rather high, differently from what we observe for TM-prjct. This is consistent with the *a priori* knowledge about the content of the TM, which is known to include many real projects commissioned by the customer of TM-prjct.

In order to verify whether IT-docs are somehow linguistically related among each other, the PP/OOV of each of them have been computed over the union of TM and the other five projects, following a cross-validation scheme. The average value is reported in the first row of the column `TMwod+`$\overline{\texttt{prjct}_i}$ of Table 3. A 14-15% relative reduction of both PP and OOV with respect to the use of `TMwod` only, indicates that IT projects are similar to each other. On the contrary, `TM-prjct` seems far from them since by adding all of them to `TMwod` for training does not change its PP/OOV values (142/0.55 vs. 143/0.55).

Finally, TM segments have been sorted according to their closeness to (the source side of) IT-docs via the data selection method described in Section 2.1. Then the bilingual TM has been split into two parts, one including the closest segments to IT-docs for a total of 5 million source words, the other including the remaining segments (22.8 million source words). The usual target LMs have been estimated on such a partition; PP/OOV values of IT-docs are shown in columns `close/farTMwod` of Table 3. It is evident that a significant portion of the TM is quite close to IT-docs: by properly selecting about 20% of the whole TM, the PP of IT-docs is globally reduced by 18% relative (from 575 to 469). The excluded TM text is important for lexical coverage however, as evidenced by the OOV increase, from 1.93% to 2.79%. On the other hand, the partition is not well suited to the TM-prjct, as expected given its distance from IT-docs which data selection was performed on.

In summary, the main outcomes of the analysis are:

- it is useful to remove duplicates from the TM
- the model of each IT-doc can be improved by exploiting the other IT-docs
- it is confirmed that TM mostly consists of documents from a specific customer; in fact:
  - there is a significant mismatch between IT-docs and the TM
  - the customer specific project TM-prjct matches the TM, not generic IT-docs
- the IT-docs/TM mismatch can be reduced by properly selecting a portion of the TM.

4 Lab Test Results

Lab tests have been performed on data sets described in Section 3. Performance are given in terms of BLEU and TER, computed by means of the `MultEval` script (Clark et al., 2011) that also provides the standard deviation σ, and of GTM.[5]

4.1 Baseline SMT for the IT domain

An IT baseline system has been built upon the open-source MT toolkit Moses (Koehn et al., 2007). The translation and the lexicalized reordering models are trained on the parallel training data available (Table 1); a 6-gram LM smoothed through the improved Kneser-Ney technique (Chen and Goodman, 1999) is estimated on the target side via the IRSTLM toolkit (Federico et al., 2008). The weights of the log-linear interpolation model are optimized by means of the standard MERT procedure provided within the Moses toolkit.

For the experiments, the set of IT-docs has been split into three equally sized blocks: the first is used for data selection (employed for adaptation, not for baselines), the second for weight tuning (dev), the third for test. Summarizing, the main features of the IT baseline system are:

- single TM, reordering model (RM) and LM, all estimated on `TMwod`

[5] nlp.cs.nyu.edu/GTM

IT system	test set	BLEU (σ)	TER (σ)	GTM
baseline	IT-docs	23.56 (0.68)	55.34 (0.65)	57.58
	TM-prjct	49.24 (0.86)	36.28 (0.76)	72.95
domain-adapted	IT-docs	27.56 (0.91)	53.14 (0.72)	59.96
	TM-prjct	44.57 (0.88)	39.53 (0.77)	70.69

Table 4: Performance of the IT systems.

- MERT on the dev block of IT-docs.

Its automatic scores on the test block of IT-docs and of TM-prjct are provided in the row `baseline` of Table 4. The large difference of scores measured on the two sets makes evident the problem of generalization of SMT models estimated on the TM.

4.2 Domain adapted SMT system (IT domain)

Let us suppose to have the goal of building an SMT engine for the translation of generic IT documents, that is with no *a priori* knowledge on them, and that the sets of Table 1 are given for training/development purposes. The main problem to face is the bias of TM towards a specific customer, as evidenced by the baseline performance in Table 4. Assuming the IT-docs as generic representative of the IT domain and the analysis outcomes of Section 3.3, an attempt to generalize the IT baseline of the previous section could rely on the following architecture:

- foreground (FG) models on the closest portion of the TM to the IT-docs
- background (BG) models on the remaining part of the TM.

Such FG/BG-based adaptation scheme has been implemented by using the TM and the six IT-docs as follows: First, the TM has been sorted with respect to the data selection block of IT-docs; then, the best ranked segments for a total amount of around 5 million source words have been used as FG data according to the PP computed on the dev block; as FG data, the data selection block has been used as well; the remaining text of the TM has been used to train BG models. The FG and BG TMs/RMs have been combined by means of the fill-up technique (Section 2.2). A single 6-gram LM has been trained on the target side of the whole TM. MERT has been run on the dev block of IT-docs. In summary, the main features of the adapted system are:

- TM and RM: fill-up of FG and BG models
- LM on the whole TM
- MERT on the dev block of IT-docs.

The row `domain-adapted` of Table 4 provides automatic scores of the adapted system computed on the test block of IT-docs and of TM-prjct. Concerning IT-docs, the adapted system outperforms the baseline by more than 4 BLEU points, corresponding to a relative improvement of over 17%, showing the better generalization capabilities of the adapted system. On the other hand, a significant degradation of performance is observed on TM-prjct, as with the adapted system we exactly wanted to smooth the bias of the baseline towards the specific customer of the TM and TM-prjct.

The IT domain adapted system has been used during the first day of the MateCat field test as reference MT engine; more details will be provided in Section 5.

4.3 Baseline SMT for the Legal domain

The Legal baseline system has been trained on data whose statistics are reported in Table 2. It is in all respects analogous to the baseline for the IT domain, apart from the weight tuning: in fact, during the development of systems for the Legal domain, we noted that default weights provided by Moses were at least as effective as those estimated via MERT. In addition, we observed that the BLEU score on development sets measured at each MERT iteration did not change from nor improve too much the initial value computed with default weights. Given such experimental evidence and with the goal of keeping the adaptation scheme as simple as possible, we decided not to run MERT and to use the default weights in our Legal systems.

Differently from the IT domain, the Legal training data can be considered general enough for building SMT models able to well capture the linguistics features of Legal documents. Hence, this baseline has been used during the first day of the

domain	test set	Day 1			Day 2		
		BLEU (σ)	TER (σ)	GTM	BLEU (σ)	TER (σ)	GTM
IT	TM-prjct	44.57 (0.88)	39.53 (0.77)	70.69	49.54 (0.92)	34.53 (0.73)	73.69
Legal	Legal-prjct	32.26 (0.98)	49.62 (0.89)	63.29	33.14 (1.00)	47.88 (0.89)	64.57

Table 5: Lab performance of the IT and Legal systems developed as for the field test. See text for details.

MateCat field test as reference MT engine; more details will be provided in Section 5.

4.4 Project adapted SMT systems

In this section we report on lab experiments about the adaptation of SMT models towards specific documents. In both domains, the project adaptation scheme resembles that of Section 4.2 used to adapt to the IT domain the IT baseline system. The scheme can be sketched as follows: from the training data, the closest portion to the source side of the dev block of the project under processing has been selected and used, together with the dev set itself, to train FG models. The remaining portion of the training data is used to estimate BG models. Translation and reordering models are built by filling-up FG with BG models as described in Section 2.2; LM is built as a mixture of FG and BG LMs as described in Section 2.3.

For the IT domain, the system has been adapted to the `TM-prjct` whose statistics are provided in Table 1. Models have been interpolated by reusing weights of the domain adapted system.

For the Legal domain, the system has been adapted to the `Legal-prjct` whose statistics are provided in Table 2. In this case, default Moses weights have been used for model interpolation, as explained in Section 4.3.

Column `Day 2` of Table 5 shows the scores of the IT and Legal systems specifically adapted to the projects `TM-prjct` and `Legal-prjct`, respectively. The scores in column `Day 1` refer to the IT domain-adapted system (Section 4.2) and to the Legal baseline (Section 4.3), respectively. Note that `Day 1` systems are those actually used during the first day of the field test. `Day 2` systems resemble the actual field test systems but differ in the adaptation data: here lab test sets were used (`TM-prjct` and `IT-prjct`), there the documents translated during Day 1 were used.

From Table 5, it results that for all tasks and for all metrics the project adapted systems consistently outperform the reference systems, proving the effectiveness of the proposed adaptation scheme.

5 Field Test Results

The field test was run with the MateCat tool, an open-source web-based CAT tool, under development within the MateCat project, integrating new MT functions and built on top of state-of-the-art MT and CAT technologies, such as Moses (Koehn et al., 2007), IRSTLM (Federico et al., 2008) and MyMemory.[6] Given a source segment to translate, the tool suggests an engine-generated translation that comes from either the TM, in case of fuzzy match higher than a threshold (that can be chosen by the translator), or the MT otherwise. During the field test, the fuzzy match threshold was set to 85% by the organizers, resulting in a clear predominance of MT suggestions in both domains (88-89% in IT, 97-98% in Legal).

The field test was organized over two days in which a document for each domain had to be translated by four translators. During the first day, for the translation of the first half of the documents, translators received MT suggestions by the reference engines described in Sections 4.2 and 4.3; during the second day, MT suggestions came from systems adapted to the text of the first day following the scheme proposed in Section 4.4. The impact of the project adaptation was measured by comparing productivity of translators during the first and the second day. Productivity was measured by two key performance indicators: average translation time for each word (time to edit) and average estimated number of edit operations applied on the suggestions (post-editings effort). The two metrics are described in Section 5.1.

Statistics on the test documents translated during the field test are reported in Table 6. Figures on the source side (English) refer to the texts the users are requested to translate (`#src wrds`). Figures on the target side (Italian) refer to the suggestions given by either the TM or the MT engine (`#sugg wrds`), and to the actual post-edits provided by the translator (`#tgt wrds`).

[6] mymemory.translated.net

field test	user	Time-to-edit (sec/word)				Post-editing effort			
		Day 1	Day 2	p-value	Δ	Day 1	Day 2	p-value	Δ
IT	t1	4.70	3.36	0.001	28.51%	34.27	30.99	0.060	9.57%
	t2	2.26	2.47	0.220	-9.29%	38.50	39.52	0.330	-2.65%
	t3	3.17	3.11	0.450	1.89%	32.53	30.17	0.133	7.25%
	t4	4.77	3.64	0.006	23.69%	32.22	28.44	0.040	11.73%
Legal	t1	5.20	5.63	0.222	-8.27%	26.47	24.57	0.212	7.18%
	t2	5.42	3.92	0.002	27.68%	29.11	26.25	0.140	9.82%
	t3	5.86	4.32	0.000	26.28%	35.65	34.11	0.247	4.32%
	t4	6.60	3.73	0.000	43.48%	22.72	18.07	0.011	20.47%

Table 7: Time-to-edit and Post-editing effort for each field test and for each translator in Day 1 and Day 2. The difference of these measures achieved in Day 1 and Day 2 and its significance p-value are also reported.

field test	test set	#seg	#src wrds	#sugg wrds	#tgt wrds
IT	Day 1	177	3,332	3.488	3,544
	Day 2	176	3,066	3,168	3,336
Legal	Day 1	91	2.960	3.056	3,202
	Day 2	90	3,007	3,153	3,421

Table 6: Statistics on test sets used in Day 1 and Day 2 of the field test. All figures refer to tokenized texts.

5.1 Key performance indicators

We used two key performance indicators to measure the effectiveness of our adaptation scheme:

- **Time to edit (TTE)**, which is the average translation drafting speed by the translators.
  TTE aims at measuring the average productivity of translators. In particular, we measure the average time taken by the translator to complete a segment in seconds per word.

- **Post-editing effort (PEE)**, which is the average percentage of word changes applied by the translators on the suggestions provided by the CAT tool.
  PEE aims at defining the quality of the matches provided by MT engine. We measured the percentage of words edited in a segment by comparing the match provided by the system and the edited segment submitted by the translator. A proprietary function was used which compares two segments and assigns a match percentage based on factors such as same words in the two segments and word order.

Table 7 reports results of key performance indicators for all field tests and for all translators. Significant TTE and PEE improvements can be observed between Day 1 and Day 2. In particular, on the IT domain, two translators out of four improved significantly in terms of both measures (t1 and t4), while on the Legal domain this was the case for three of four (t2-t4). All observed TTE reductions were statistically significant, while the same hold only for three out of the observed PEE variations. By looking at the average productivity gains, on the IT domain we observed 11.2% gain in TTE and a 6.5% in PEE, while on the Legal domain we observed a 22.2% gain in TTE and a 10.7% in PEE. Finally, the good correlation observed between PEE and TTE under the different conditions show that very likely the translators were able to took advantage of MT suggestions, and that the adapted MT engine suggestions were in general better. In fact, better PEE effort was observed for 7 translators of 8.

6 Conclusions

In this paper we have faced an hot research topic for CAT industry: how to make self-tuning on the user feedback the SMT systems equipping CAT tools. Self-tuning can be seen at two different scales: at the domain level or simply at the project level. At the larger scale, the goal is to focus general purpose models towards the specific domain of interest; for example, this could be applied for preparing the MT system to be employed at the beginning of the translation process once the domain of the translation project is known. At the lower scale, the goal is to further focus in-domain models towards the specific translation project, once the source text is available and the post-editions

start to come; this kind of self-tuning can be applied in any time, provided that enough fresh data is at disposal for updating the models according to the needs of the methods employed.

For handling both types of self-tuning, we have proposed an adaptation scheme which has been tested in a terrific experimental framework, consisting of not only reproducible lab tests but even field tests which involved professional translators and the industrial partner of MateCat, the project inside which this work has been conducted.

The collected experimental results proved the effectiveness of the proposed scheme used to integrate project adapted SMT systems into the CAT workflow: gains of human translator productivity up to over 20% were measured.

Nevertheless, several still open issues deserve to be investigated in the future. First of all, gains observed in Day 2 could be partially due to the familiarization of the users with the system and with the specific project; in order to exclude this effect and to precisely measure the net contribution of the project adaptation method, in Day 2 of forthcoming field tests translation suggestions will be generated by both the project-adapted and the reference systems. Secondly, our field test experiments showed an average productivity improvement but also a performance degradation for one translator: this odd behavior will be carefully analyzed. Finally, we will also investigate the adaptation rate of SMT models for CAT in two respects: from one side, what is the amount of post edits required to achieve the best trade-off between performance and computational costs; from the other side, what is the learning curve when more daily adaptation steps are performed.

Acknowledgments

This work was supported by the MateCat project, which is funded by the EC under the 7^{th} Framework Programme.

References

Axelrod, A., X. He, and J. Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *EMNLP*, Edinburgh, UK.

Bisazza, A., N. Ruiz, and M. Federico. 2011. Fill-up versus interpolation methods for phrase-based SMT adaptation. In *IWSLT*, San Francisco, US-CA.

Braune, F. and A. Fraser. 2010. Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora. In *Coling: Posters*, Beijing, China.

Chen, S. F. and J. Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 4(13):359–393.

Clark, J., C. Dyer, A. Lavie, and N. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *ACL*, Portland, US-OR.

Federico, M. and N. Bertoldi. 2004. Broadcast news LM adaptation over time. *Computer Speech and Language*, 18(4):417–435, October.

Federico, M., N. Bertoldi, and M. Cettolo. 2008. IRSTLM: an open source toolkit for handling large scale language models. In *Interspeech*, Melbourne, Australia.

Foster, G., C. Goutte, and R. Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *EMNLP*, Cambridge, US-MA.

Koehn, P., H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *ACL Companion Volume Proc. of the Demo and Poster Sessions*, Prague, Czech Republic.

Matsoukas, S., A.-V. I. Rosti, and B. Zhang. 2009. Discriminative corpus weight estimation for machine translation. In *EMNLP*, Singapore.

Moore, R. C. and W. Lewis. 2010. Intelligent selection of language model training data. In *ACL Short Papers*, Uppsala, Sweden.

Nakov, P. 2008. Improving English-Spanish statistical machine translation: experiments in domain adaptation, sentence paraphrasing, tokenization, and recasing. In *WMT*, Columbus, US-OH.

Niehues, Jan and Alex Waibel. 2012. Detailed analysis of different strategies for phrase table adaptation in SMT. In *AMTA*, San Diego, US-CA.

Steinberger, R., B. Pouliquen, A. Widiger, C. Ignat, T. Erjavec, D. Tufi, and D. Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *LREC*, Genoa, Italy.

Tiedemann, J. 2012. Parallel data, tools and interfaces in OPUS. In *LREC*, Istanbul, Turkey.

Yasuda, K., R. Zhang, H. Yamamoto, and E. Sumita. 2008. Method of selecting training data to build a compact and efficient translation model. In *IJCNLP*, Hyderabad, India.