АБУМАТРАН
ABUMATRAN

Abu-MaTran

Automatic building of Machine Translation

PIAP- GA-2012-324414

D4.1b MT systems for the second development cycle

Dissemination level	Public
Delivery date	2014/12/31
Status and version	Final, v1.0
Authors and affiliation	Mikel L. Forcada (UA), Sergio Ortiz-Rojas (Prompsit), Tommi Pirinen (DCU), Raphaël Rubino (Prompsit), Antonio Toral (DCU)

PEOPLE

Project funded by the European Community under the Seventh Framework Programme for Research and Technological Development

Table of Contents

Executive Summary

This deliverable D4.1b describes work done in the area of machine translation development and deployment (work package 4) during the period of the second milestone of the project (from month 7 to month 24). As part of this second development cycle, we have built a new generic MT system by building upon the system we prepared for milestone 1; we have then broadened our scope by (i) building a MT system for a specific domain (tourism) and (ii) experimenting with the application of linguistic knowledge, through a process called morph segmentation, to avoid the data sparsity caused by unknown word forms which can however be morphologically broken into known tokens; we have developed a comprehensive architecture to provide these machine translation systems through a web interface. The deliverable also briefly reports on other related activities in this work package, namely (i) our participation in the WMT14 translation contest and (ii) experiments with a procedure to clean noisy parallel corpora.

1 Introduction

This deliverable reports on the activities carried out within the project regarding the development and deployment of machine translation (MT) systems during the second milestone period (M7 to M24). The main aim of the deliverable is to provide high-quality MT systems for the languages of the project's use case. In order to do so, we exploit the resources acquired (cf. deliverable D3.1b). The evaluation of the systems described here can be found in deliverable D5.1b.

The rest of this deliverable is organised as follows. In Section 2 we report on the MT systems built for the second milestone of the project: (i) a generic system that aims at improving the one we built for milestone 1 (Section 2.1) and (ii) a domain-specific system for the tourism domain (Section 2.2). In Section 3 we report on a new approach to MT for morphologically rich languages,based on morph segmentation. In Section 4 we cover our participation in the WMT14 shared task. Section 5 describes the cleaning of the OpenSubtitles corpus and the MT systems built on clean and unclean versions of this corpus. Finally, Section 6 describes the software platform developed in the project to provide all the MT systems we develop from a web interface.

2 Milestone 2 MT Systems

2.1 General MT System

2.1.1 Context

In the work described in public deliverables D4.1a (Toral et al. 2013a) and D5.1a (Toral et al. 2013b), several MT systems were trained in order to translate general purpose text. Three test sets were used to evaluate the MT systems and one of them, the manual translation into Croatian of the English test set for the 8th Workshop on Statistical Machine Translation WMT13,[1] led to the poorest performances as indicated by the automatic metrics described in D5.1a.

In addition, based on the nature of this test set (taken from 52 online news sources in various languages) and the thematic diversity of its content, we assume that the results obtained with our MT systems are coherent in a realistic and generic scenario. This thematic heterogeneity of the data to translate is also a way to simulate, on a smaller scale, the use of multiple, domain-specific, test sets. We will compare the MT systems described in this deliverable to the ones described in D4.1a in terms of data and training methods. We also focus on phrase-based statistical machine translation (SMT, Koehn, 2010), as it is the most commonly used paradigm for MT nowadays.

The aim of Milestone 2 MT systems is to improve upon the MT systems trained and evaluated for Milestone 1 of the project. In order to achieve this goal, we decide to increase the amount of parallel data used to train the translation models and the amount of monolingual data used to train the language models, described in Sections 2.1.2 and 2.1.3.

2.1.2 Translation Model

Milestone 1 SMT systems, trained for direct English–Croatian translation (in both directions), are built upon three bilingual parallel corpora, namely SETimes, hrenWaC and TED Talks, which are also used to train Milestone 2 SMT systems described in this deliverable. In addition to these corpora, we take advantage of the latest language technologies releases from the Joint Research Center:[2]

[1] http://www.statmt.org/wmt13/

[2] https://ec.europa.eu/jrc/en/language-technologies/

- The Translation Memory of the DGT,[3] the Directorate General for Translation of the European Commission.
- the Croatian translation of the Joint Research Centre's (JRC) Acquis Corpus[4]

We also use the OpenSubtitles 2013 corpus, available from the OPUS platform.[5] This corpus appears to be noisy (i.e. wrong alignments, mistranslations, mixed languages); it therefore requires an intensive cleaning process before it is usable for training MT systems (see Section 5).

In addition to this specific cleaning step applied to OpenSubtitles 2013, we perform several steps of pre-processing (before training the SMT systems) on the corpora: punctuation normalisation, tokenisation, removing sentences shorter than 1 and longer than 80 tokens, true-casing[6] and escaping problematic characters for the SMT tools used. Details about the corpora used to train the Milestone 2 SMT systems are presented in Table 1.

The tools used to train the SMT systems are out-of-the-box open-source tools widely used in the SMT research and industry communities:

- MGIZA++ for word alignment (Gao & Vogel, 2008),[7]
- Moses v2.1.1 for phrase extraction and alignment, as well as decoding.

We train individual SMT systems on each of the parallel corpora to get an indication of the quality of the training data when used for SMT. We then concatenate all the parallel corpora except OpenSubtitles 2013, and finally we concatenate all the parallel corpora including also this latter corpus.

	before pre-processing			**after pre-processing**		
Label	**Sentence pairs**	**Tokens EN**	**Tokens HR**	**Sentence pairs**	**Tokens EN**	**Tokens HR**
DGT TM	276,502	4,717,544	4,158,312	271,857	5,097,554	4,599,970
HrEnWaC	99,001	2,338,723	1,999,359	97,888	2,531,285	2,190,638
JRC Acquis	683,924	10,312,020	8,849,841	677,453	11,454,284	10,033,433
OpenSub	17,243,328	127,852,973	104,287,409	16,950,842	160,128,956	127,513,902
SETimes	201,910	4,162,738	3,896,904	201,800	4,834,147	4,503,386
Ted	86,348	1,287,656	1,092,742	86,186	1,504,996	1,269,554

Table 1: Parallel Corpora used in the Milestone 2 MT systems

[3] https://ec.europa.eu/jrc/en/language-technologies/dgt-translation-memory

[4] http://tinyurl.com/CroatianAcquis

[5] http://opus.lingfil.uu.se/

[6] using a true-case model trained on all the monolingual and bilingual data except OpenSubtitles 2013

[7] http://www.kyloo.net/software/doku.php/mgiza:overview

2.1.3 Language Model

The language model is the component of an SMT system which helps in generating fluent text by modelling the target language with sequences of words called *n*-grams. Milestone 1 language models were built on the target side of the parallel data used to train the translation model, or a large monolingual resource, namely hrWaC1.0 for Croatian and the WMT13 data, both described in deliverable D3.1a (Toral et al. 2013c). The evaluation of both data sources to train language models (see deliverable D5.1a, Toral et al., 2013b) indicates that larger monolingual corpora lead to the best performances compared to using the target side of the parallel corpora, according to three automatic metrics.

We decide to use both large monolingual corpora for both English and Croatian language models, but also to use the target side of the parallel corpora when Croatian is the target language. Adding the Croatian side of the parallel data to the language models allows us to cope with the lack of data in this particular language. When English is concerned, for instance, the amount of monolingual data is large enough to train robust language models. Thus, we investigate the use of target side parallel data in addition to monolingual data to clear the low-resource hurdle implied by the Croatian language. This method could be applied to any low-resource target language.

The corpora used to train the Croatian language model are:

- DGT Translation Memory, JRC Acquis, OpenSubtitles 2013, SETimes and TED Talks, using the Croatian side of these bilingual corpora,
- hrWaC2.0, the new version of the Croatian Web-crawled corpus[8], as an example of large monolingual corpus.

The corpora used to train the English language model are:

- all the monolingual data released for the WMT14 translation task[9]
- the English side of each parallel corpus released for WMT14, removing overlap with monolingual corpora.

More details about this language model are given in Rubino et al. (2014).

The Croatian language model is trained on the concatenation of the bilingual and monolingual corpora described in this section, after applying the same pre-processing steps that we applied on the parallel corpora, except the filtering of long sentences. We train a 5-gram smoothed Kneser-Ney language model using the KenLM toolkit[10] (Heafield 2011). Table 2 reports the size of the two language models used for the Milestone 2 SMT systems in terms of number of *n*-grams. There are more 4-grams and 5-grams in the Croatian language model because we did not prune it as we did for these high order *n*-grams appearing less than twice in the English language model.

	1-gram	2-gram	3-gram	4-gram	5-gram
English	13.4M	198.6M	381.2M	776.3M	1,068.7M
Croatian	10M	173.3M	621.6M	1,032.2M	1,215.2M

Table 2: Size of the language models of the Milestone 2 MT systems (in number of n-grams)

2.2 Tourism

In this section we describe briefly the methodology followed to build domain-specific MT systems for tourism for the language direction Croatian to English. Toral et al. (2014) describe this task in further detail.

[8] http://nlp.ffzg.hr/resources/corpora/hrwac

[9] http://www.statmt.org/wmt14/translation-task.html

[10] https://kheafield.com/code/kenlm/

We have built a number of MT systems using the domain-specific parallel corpus for tourism (cf. Section 3.1 in Deliverable 3.1b) as well as the generic parallel corpora used in our MT system for the first milestone (i.e. HrEnWaC, SETimes and Ted). In particular we have built systems that fall into the following three categories:

- Generic. This system uses solely generic corpora (this is the SMT system trained for Milestone 1 and described in Deliverable 4.1a).
- Specific. These systems use only domain-specific data. We have four such systems, which use the corpora obtained by each of the two crawlers, the union of both corpora and the intersection of both corpora.
- Generic+Specific. This system uses the concatenation of the generic corpora and the union of the domain-specific corpora acquired by both crawlers.

All these systems share the same English language model, which is the one we built for our participation in the WMT'14 shared task (cf. Section 4). All systems are tuned on a subset of the domain-specific corpus (825 sentence pairs). Finally, all the systems are evaluated on another subset of the domain-specific corpus (816 sentence pairs).

As part of the presentation of project Abu-MaTran (Abu-MaTran, 2014) at the 18th Annual Conference of the European Association for Machine Translation (EAMT 2014, Dubrovnik), we demonstrated a mobile application that uses the best MT system we built for the tourism domain (generic+specific). It is based on the open-source Mitzuli translation app for Android adapted to our task.[11] While the original system uses the Apertium MT engine (Forcada et al., 2011) to provide off-line translation and its web-service to provide online translation possibilities, we extended Mitzuli in collaboration to its developer (Mikel Artetxe) by implementing a backend that works with our web-service (see Section 5).

3 Morph Segmentation

Morph segmentation is a process in the morphological analysis of word-forms where the word is broken into sub-parts called *morphs* based on some criteria. Its purpose is to solve the following problems:

- vocabulary size increases with morphological complexity but the sizes of morph sets do not vary much across languages (see Figure 1 for English to Croatian comparison); unknown word-forms may usually be represented as combination of known morphs,
- the translations between languages will be close to 1-to-1 when represented in terms of morphs or morph phrases (see also fig. 2). As an example, the Finnish phrase *pöydällä* prototypically translates into English as 'on (a) table', while it is likely that the exact correspondence for this word is common enough to appear in any parallel corpus and therefore its translation can be found by the statistical translation model; it may however be the case that such a correspondence cannot be found for rarer word-forms such as *Seinellä* 'on Seine' while both *Seine* and *-llä* may have been seen.

Morph segmentation can be supervised or unsupervised: the unsupervised approach uses large amounts of text to find ways to minimise the complexity of text over some statistical factors. From this style of morphology, Morfessor[12] is the best known and generally considered state-of-the-art as evidenced by its lead in earlier MorphoChallenge shared tasks.[13] Morfessor has several different algorithms that use different statistical formulas to find the optimal segments; from these we use out-of-the-box versions of Morfessor 2.0 Baseline (Virpioja et al. 2012), which optimises over minimum description length, i.e., trying to find the minimum amount of different segments that cover the whole text is trained with, and Morfessor Flatcat (Grönroos et al., 2014), which aims to be

[11] https://github.com/flammie/mitzuli

[12] http://www.cis.hut.fi/projects/morpho/

[13] http://research.ics.aalto.fi/events/morphochallenge/

more linguistically accurate using context-based parameters as well. Another approach is using traditional rule-based morphological analysers, such as the ones defined in Beesley and Karttunen (2003), here each word-form is analysed along linguistic principles to produce etymologically relevant morphs, but the approach requires an existing computational linguistic description of the language.

Figure 2 is a graph showing the number of new unique words in the Europarl corpus (Koehn 2005), that is, how many new words are seen as you read Europarl from the start onwards, both using words in text as they are seen by baseline SMT system and after pre-processing with unsupervised morphological segmentation. This should illustrate neatly how the morphological complexity affects the rate of unseen tokens therefore potentially lowering the quality of SMT. The figure shows un-preprocessed English in red and un-preprocessed Croatian in green. The pre-processing of Croatian with two different Morfessor algorithms, Morfessor 2.0 baseline and Morfessor Flatcat, shown in blue and yellow respectively, results in growth patterns that look more similar to those of English. In fact, Morfessor 2.0 baseline, whose goal is to minimise the amount of different tokens has much smaller unique token count by default.

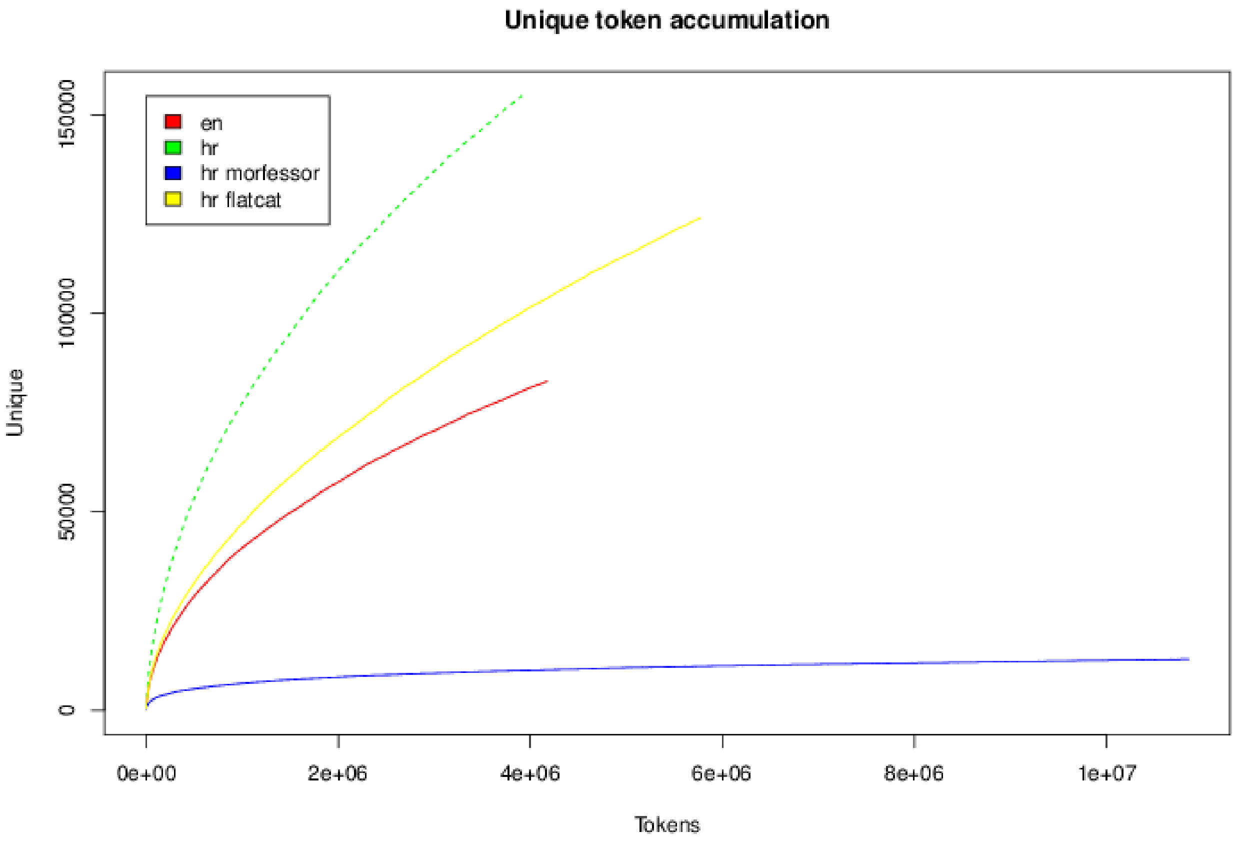


Figure 1: The accumulation of new, unseen word-forms when reading the Europarl corpus from beginning to end: the red and green lines represent the number of English and Croatian word-forms respectively, while the blue and yellow show the number of Croatian morphs as pre-processed with Morfessor 2.0 Baseline and Morfessor Flatcat respectively (see text).

For example, when translating between Finnish (an example of a morphologically rich language) and English (example of a morphologically simpler language),[14] it is common for one word to

[14] Finnish, although not being one of the case study languages of Abu-MaTran, is used in this deliverable as an example of morphologically rich language and the need for morph segmentation. Furthermore we have resources already available that are not ready yet for Croatian, such as mature rule-based segmentation or an established SMT corpus like europarl.

correspond to more than one word. For instance, suffixes typically align to English words, e.g. the suffix *-sta* will most commonly align to the English preposition *from*. As shown in Figure 2, however, this is not always the case, and constructions like “I like the dog” literally translate in Finnish as I like from dog.

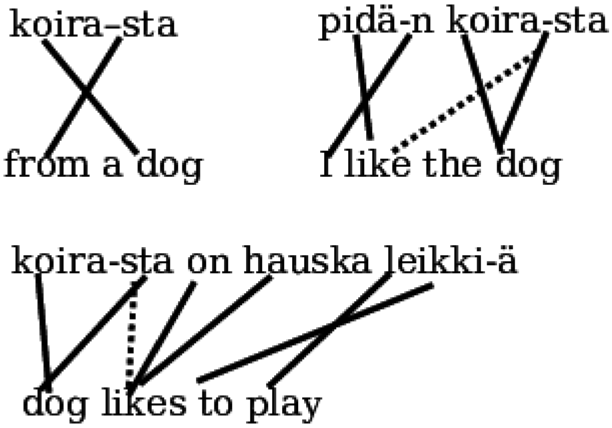


Figure 2: aligning Finnish morphs to English words; the prototypical case aligns a suffix to a preposition whereas certain constructions require specific forms, i.e. the suffix is dependent on the main verb construction and untranslated

The task of morphological segmentation is not straightforward, as there are multiple plausible segmentations for each word and multiple approaches to select the segmentations to begin with. This poses an interesting research question: to find the most suitable criteria for segmentation for MT. We have approached the task by running experiments on several segmentation criteria and methods to find the best segmentation among the results.

The use of morph segmentation extends the whole SMT pipeline by two main operations:

- Pre-processing segmentation, which is used prior to training for the morphologically complex language (in both translation directions) and prior to translation from the morphologically complex language into the morphologically simpler language.
- Post-processing de-segmentation or joining, which is used after decoding in the morphologically simple to complex translation task. This has a minimal effect on the standard main pipeline of Moses and can be attached to the output of the pipeline easily.

For example, let us consider a randomly selected sentence pair from the English--Croatian corpus SETimes (described in Deliverable D3.1a):

> Pozvao je političare da obave svoj dio i osiguraju civiliziranu kampanju, te slobodne i poštene izbore.
>
> =
>
> He called on politicians to do their part in ensuring a civilised campaign and free and fair elections.

We segment the Croatian side with Morfessor into morphs using the marks (→ and ←) to show how morphs must be attached:

> Po→ ←zvao je političar→ ←e da obave svoj dio i osigura→ ←ju civiliziran→ ←u kampa→ ←nju , te slobodne i pošte→ ←ne izbor→ ←e .

The second form (or an equivalent to it) is to be used as an input for morph-based model training and as an input for translating with the morph-based model from Croatian. When translating into Croatian the output is similar to the second example, but, due to the nature of statistical MT, the arrows attached to some of the morphs may not match and the desegmentation needs to decide what to do with them: remove non matching morphs (which we use for the baseline), try to combine with other morphs or predict missing morphs for example. So for the current version of de-compounding we: *attach adjacent arrows and delete the morphs that are not attached afterwards*.

The segmentations provided by different approaches can vary a lot:

- For rule-based segmentation there are a number of different linguistically-defined boundaries within the word: compound, inflection, derivation, stem.
- The statistical approaches may be either limited to a simply delimiting morphs (Morfessor 2.0 Baseline), or tagging them: prefix, stem, suffix and non-morpheme (Morfessor Flatcat).

For example translating the above-mentioned English sentence with our English to segmented Croatian translation system results in

> On je po→ ←zvao političar→ ←e da u→ ←činiti svoj dio posla u osigura→ ←nju civiliziran→ ←oj kampa→ ←nji i slobodne i prave→ ←dne izbor→ ←e.

which needs to be combined to proper grammatically correct Croatian using some criteria. For baseline, we simply removed (→, ←) combinations with their intervening space character and deleted any leftover morphs, which provides sufficiently fluent results. However, sometimes morphs can appear in places where it is non-trivial to join them to any adjacent units or joining without further processing results in ungrammatical language or even incorrect combinations. During year 3 of the project we are going to go through the rest of the more complicated cases in order to devise an ideal way to combine them into final translation.

3.1 Tools for Morph-Based MT

We have implemented an experimental framework[15] following the standard GNU conventions for open-source projects which is freely available with reference implementations for our Croatian–English corpora (See Deliverable D3.1a, Toral et al. 2013c) and for language pairs based on Europarl, making use of various freely available morph segmentation algorithms and the Moses toolkit for MT. The components of the system are configurable, but the default settings follow mainly the baseline Moses system,[11] that is, Moses 2.1.1, MGIZA++-0.7.3, irstlm-5.80.06, which have been tested to work, as well as Morfessor-2.0.2_alpha and Flatcat-1.0.

The system can be used to build and deploy new MT systems rapidly with a standard setup using GNU development commands (i.e. autoreconf, configure and make). For configuration the setup requires manually giving paths for relevant tools and lexical data.[16] An example :

```
$  git clone https://github.com/flammie/autostuff-moses-smt.git
$  cd autostuff-moses-smt
$  autoreconf -i
$ ./configure --enable-parallel --with-
moses=/home/tpirinen/Koodit/mosesdecoder --with-
irstlm=/home/tpirinen/irstlm --with-
mgizapp=/home/tpirinen/Koodit/mgizapp/bin --with-giza-
tools=/home/tpirinen/Koodit/mgizapp/bin/
$ cd hbs-eng
$ make eng-hbs
```

[15] http://github.com/flammie/autostuff-moses-smt

[16] http://statmt.org/moses/Moses.Baseline

These commands would make all of our English (eng) to Croatian (hbs) models, provided that it has access to our corpus via the Internet or as local files (which are not available from the referred Github repository).

In addition to unsupervised segmentation schemes from Morfessor, the tools we built support rule-based segmentation from a finite-state morphology scheme. An example of this is in the Finnish–English pair where the rule-based morphological segmentation obtained using omorfi[17] is transformed into the format required by SMT toolkits such as Moses.

Large part of the work in this task has been to tune the morphological segmentation to find the optimal segmentation based on the translation task. To this effect we started at the Machine Translation Marathon at Trento[18] a project (project group consisting of myself, Nick Ruiz, Francis Tyers, Liling Tan, detailed list of authors is of course available via github web service[19] as well) for a prototypical system for bilingual segmentation and training using finite-state technology. The results of the Trento project are freely available as a python-based finite-state segmentation alignment tool.[19]

3.2 Translation Models

The models that are available currently in our experiment repository[10] include a linguistically motivated selection of language pairs to draw a reasonable comparison of the effect of different segmentation parameters as part of the SMT task. The Croatian–English pair in the hbs-eng sub-directory is the same as the one used in other tasks (cf. Section 2.1) with morph segmentation added on. For Croatian–English we built a total of six main translation models: two for each direction and one with Morfessor 2.0 Baseline and one with Morfessor Flatcat in addition to the baseline system. In Table 3 one can see the token counts for each pre-processing scheme for the SETimes corpus described in the deliverable D3.1a (Toral et al. 2013c).

Pre-processing	**Unique HR tokens**
None (word-forms)	141,465
Morfessor 2.0 baseline	52,746
Morfessor Flatcat	73,309

Table 3: number of unique tokens for Croatian with different pre-processing schemes

The Finnish–English pair is used to test the system on a language with higher level of morphological complexity than Croatian, when counting e.g., wordforms-per-word or unique token accumulation, as we have existing linguistic models for processing Finnish to work with and compare. With Finnish we have built eight translation models, four per translation direction, using Morfessor baseline and Morfessor Flatcat as before, but also using rule-based approaches to build translation models based on selecting compound and morph boundaries. Table 4 shows segment counts for each pre-processing scheme for the Europarl-v7 corpus.[20]

[17] http://code.google.com/p/omorfi

[18] http://statmt.org/mtm14/

[19] http://github.com/NickRuiz/lattice-align

[20] http://www.statmt.org/europarl/

Pre-processing	Unique FI tokens
None (word-forms)	612,432
Morfessor 2.0 baseline	133,390
Morfessor Flatcat	150,971
Omorfi morphs	187,133
Omorfi compounds	391,231

Table 4: number of unique tokens for Finnish with different pre-processing schemes

For other language pairs, the system may easily be configured for any language pair in Europarl. The same is true for other language pairs as well, but the procedure requires a sentence-aligned training corpus that can be passed through the Moses pipeline (i.e. it contains at least some sentences that can pass `clean-corpus-n.perl`, etc.).

4 Participation in the WMT14 Shared Task

The machine translation community evaluates their systems every year on a MT shared task, the Workshop on Machine Translation (WMT). The members of the Abu-MaTran project participated in the 2014 edition on the French–English language pair and built two SMT systems, one for each translation direction. Rubino et al. (2014) describe these MT systems in further detail.

5 Cleaning of the OpenSubtitles corpus

We propose a procedure to clean parallel corpora so that these corpora can be useful to train MT systems. Our motivation comes from the fact that there are vast amounts of publicly available parallel data that are not clean enough to be usable to be used for training MT systems. Probably the best example of such corpora is OpenSubtitles,[21] a set of corpora made of open-domain subtitles available for several language pairs.

This cleaning procedure consists basically on:

- Converting Cyrillic characters to their Latin counterparts: some subtitles contain Cyrillic mixed with Latin characters and the problem becomes fixed by converting individually each character.
- Detecting and replacing, when possible, switched letters ("I" —uppercase “i”—- for "l" —ell— in a word which is not in capitals).
- Fixing frequent spelling errors detected by a spell-checker.
- Fixing inconsistent punctuation marks, numbers and spacing.
- Removing sentences without alphabetical characters or too different in length or not matching the appropriate source/target language using the LangID language detector[22] in both sides of the corpus for each sentence.

The input to our cleaning procedure is made of the 2012 and 2013 editions of OpenSubtitles (30,035,928 sentence pairs). Applying our procedure leads to a subset of 17,243,328 unique sentence pairs (57.40% are kept).

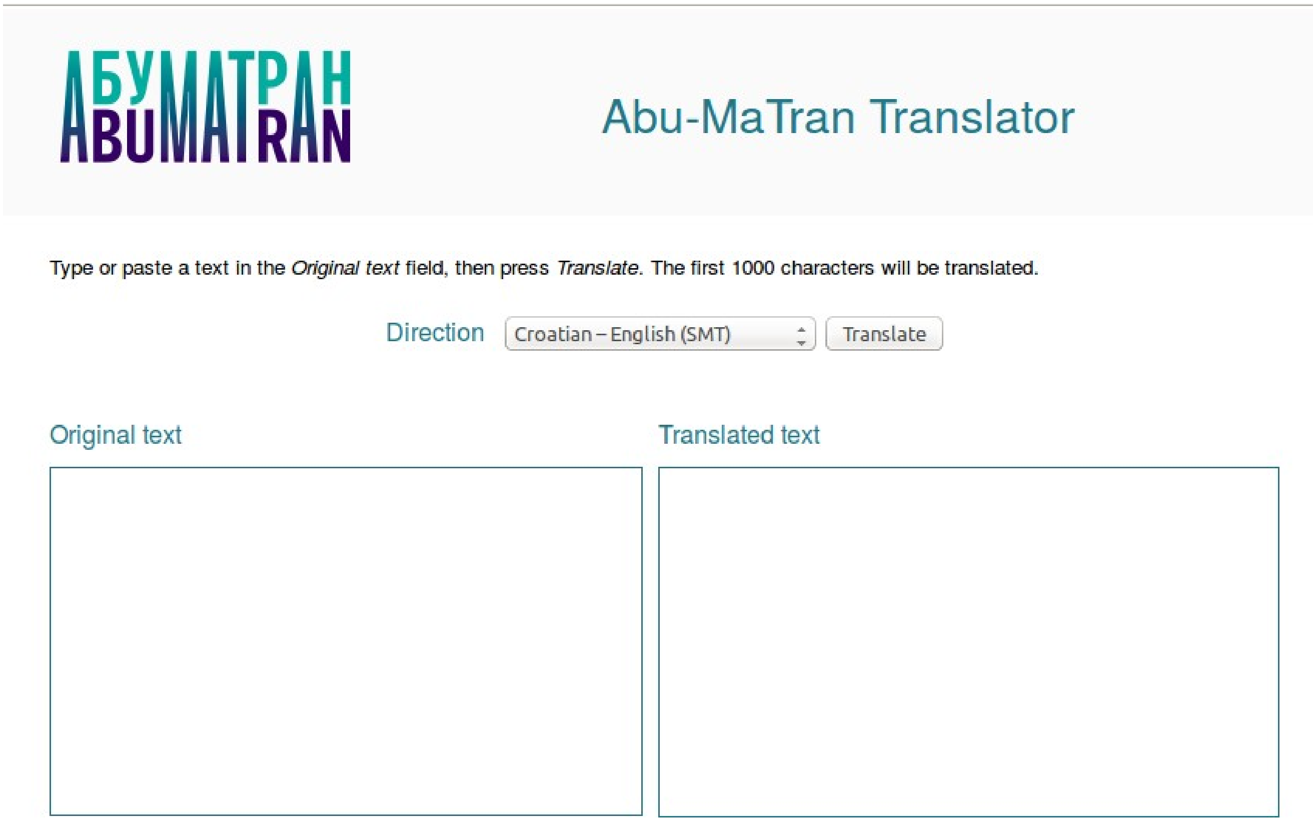
[21] http://opus.lingfil.uu.se/OpenSubtitles.php

[22] https://github.com/saffsd/langid.py

In order to evaluate the impact of the cleaning process on this parallel corpus, we conduct an experiment by training SMT systems on both translation directions, using first a non-cleaned (original) version of the corpus, and second a cleaned version of it. The resulting four systems are trained with the same parameters using MGIZA++ and Moses v2.1.1. The decoding of the WMT13 English–Croatian test set is done using the language models presented in Section 2.1.3 of this deliverable. We evaluate the decoded test set and presents the results according to three automatic metrics in Section 5 of deliverable D5.1b (Forcada et al. 2014).

6 Web Translator

One of the contributions of the Abu-MaTran project is to set up online translators for the languages of newly arrived European countries. The Milestone 1 SMT system was already accessible through the Abu-MaTran web translator available at http://translator.abumatran.eu for the Croatian to English translation direction (see Figure 3 for a screenshot of the online translator).



Type or paste a text in the *Original text* field, then press *Translate*. The first 1000 characters will be translated.

Direction: Croatian – English (SMT)

Figure 3: Screenshot of the Abu-MaTran's online translator

For Milestone 2 of the project, the English to Croatian direction has been added to the online translator. However, due to the way the initial online translator was designed, and due to the large translation and language models used in our SMT systems, it is not possible to host simultaneously the two SMT systems on the same server. Also, each translation query sent to the online translator takes, for the Milestone 1 SMT system, more than 15 seconds on average per sentence.

These two aspects motivate us to develop an improved online translator for the Abu-MaTran project, allowing us to set up multiple language pairs and directions, to speed up the translation process, and ease the deployment and maintenance of new MT systems. Because the power of servers limit the number of simultaneously running SMT systems, we decide to separate the entry point of the translator (http://translator.abumatran.eu) and the MT systems themselves and deploy them on different servers. With this technique, we can multiply the number of MT systems available through the Abu-MaTran translator without increasing the computation load on this server. Figure 4 describes the general architecture of our new online translator.

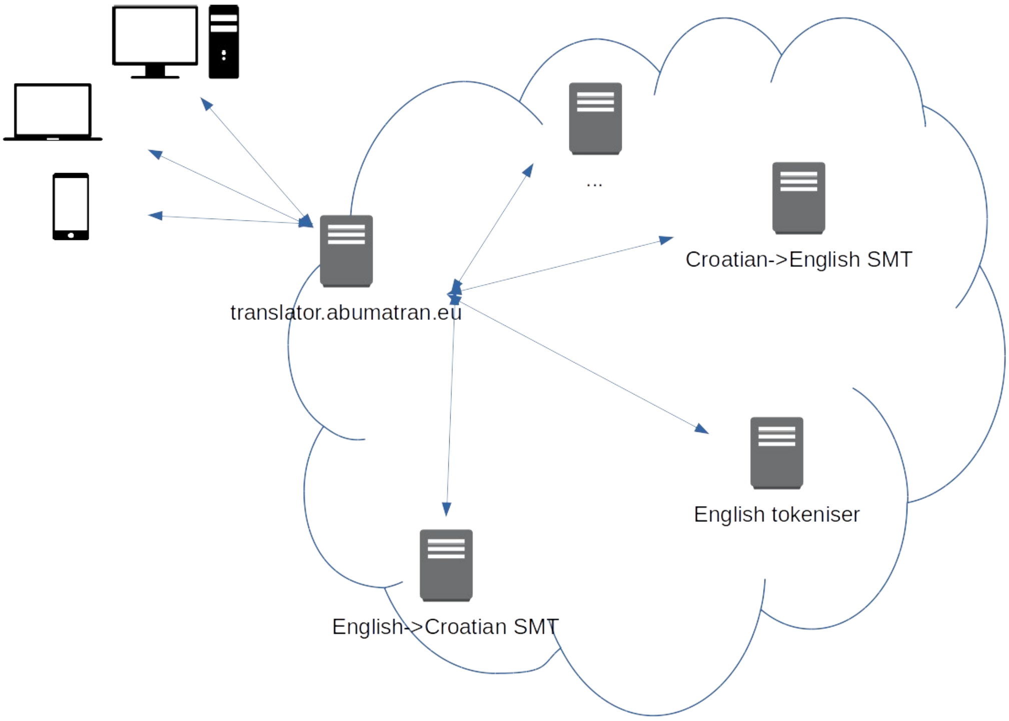


Figure 4: Online translator architecture

The new Abu-MaTran online translation service allows users to benefit from the multiple translation systems developed for the project, but also from the existing tools adapted to the languages of the project. A list of all the translation systems available on the Abu-MaTran online translation service is presented in Table 5, along with the average translation time per sentence (for sentences with an average length of 20 words).

		Target					**Legend**
		Bosnian	Croatian	English	Serbian	Slovene	SMT
Source	Bosnian	-	-	-	-	-	RBMT
	Croatian	-	-	1.2	0.2	-	
	English	0.25	1.4	-	0.2	-	
	Serbian	-	0.2	-	-	-	
	Slovene	0.2	0.2	-	0.2	-	

Table 5: Average decoding time per sentence (average length of 20 words) in seconds, colors indicate which approach is used by the MT system. When both approaches are available for a language pair, the best performing one according to automatic metrics is presented.

7 Conclusions

This deliverable has covered the work done in the area of MT development and deployment (Work Package 4) during the period of the second milestone of the project (M7–M24). We have built a new generic MT system by building upon the system we prepared for milestone 1. We have then broadened our scope by (i) building a MT system for a specific domain (tourism) and (ii) experimenting with the application of linguistic knowledge, through morph segmentation, to avoid data sparsity. We have developed a comprehensive architecture to provide our MT systems through a web interface. Other related activities in this workpackage have regarded (i) our participation in WMT14 and (ii) experimenting with a cleaning procedure for noisy parallel corpora.

Regarding future work for milestone 3, we consider mainly the following two lines of action:

1. Improving upon our current generic MT systems. To do so we consider (i) acquiring and adding more data sets, (ii) exploring optimal ways to combine different data sets and (iii) exploring ways to add linguistic knowledge.
2. Building MT systems not only for Croatian but also for other related South Slavic languages.

Bibliography

Abu-MaTran (2014) “Abu-MaTran: Automatic building of Machine Translation”, In Proceedings of the 17th Annual Conference of the European Association for Machine Translation (EAMT), p. 132.

Beesley, K.R., Karttunen, L. (2003) Finite State Morphology, CSLI Publications, Stanford, Calif.

Forcada, M.L., Ginestí-Rosell, M., Nordfalk, J., O'Regan, J., Ortiz-Rojas, S., Pérez-Ortiz, J.A., Sánchez-Martínez, F., Ramírez-Sánchez, G., Tyers, F. (2011) “Apertium: a free/open-source platform for rule-based machine translation”, *Machine Translation* 25:2, 127-144.

Forcada, M.L., Pirinen, T., Rubino. R., Toral, A. (2014) “Deliverable D5.1b: Evaluation of the MT systems deployed in the second development cycle”, version 1.0, available from http://www.abumatran.eu/?page_id=59

Gao, Q., Vogel, S. (2008) “Parallel Implementations of Word Alignment Tool”, Software Engineering, Testing, and Quality Assurance for Natural Language Processing, pp. 49-57.

Grönroos S.-A., Virpioja, S., Smit, P., Kurimo, M (2014). Morfessor FlatCat: An HMM-based method for unsupervised and semi-supervised learning of morphology. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics (Dublin, Ireland, August 2014), Technical Papers, p. 1177–1185, Association for Computational Linguistics.

Heafield, K. (2011) “KenLM: Faster and smaller language model queries”, Proceedings of the Sixth Workshop on Statistical Machine Translation, pp. 187--197.

Koehn, P. (2005) “Europarl: A parallel corpus for statistical machine translation”, Proceedings of Machine Translation Summit X, pp. 79–86.

Koehn, P. (2010) *Statistical Machine Translation* (New York, Cambridge University Press), chapter 5: “Phrase-based models”.

Rubino, R., Toral, A., Sánchez-Cartagena, V.M., Ferrández-Tordera, J., Ortiz-Rojas, S. Ramírez-Sánchez, G., Sánchez-Martínez, F., Way, A. (2014), “Abu-MaTran at WMT 2014 Translation Task:

Two-step Data Selection and RBMT-Style Synthetic Rules". In Proceedings of the 9th Workshop on Statistical Machine Translation (WMT), pp. 171–177.

Toral, A., Cortés-Vaíllo, S., Ortiz-Rojas, S., Ramírez-Sánchez, G., Forcada, M.L. (2013a) "Deliverable D4.1a: MT systems for the first development cycle", version 1.0, available from http://www.abumatran.eu/?page_id=59

Toral, A., Cortés-Vaíllo, S., Ramírez-Sánchez, G., Klubička, F., Ljubešić, N. (2013b) "Deliverable D5.1a: Evaluation of the MT systems deployed in the first development cycle", version 1.0, available from http://www.abumatran.eu/?page_id=59

Toral, A., Cortés-Vaíllo, S., Ramírez-Sánchez, G., Forcada, M.L., Ljubešić, N. (2013c) "Deliverable D3.1a: Acquisition for the first development cycle", version 1.0, available from http://www.abumatran.eu/?page_id=59

Toral, A. Rubino, R., Esplà-Gomis, M., Pirinen, T., Way, A., Ramírez-Sánchez, G. (2014), "Extrinsic Evaluation of Web-Crawlers in Machine Translation: a Case Study on Croatian–English for the Tourism Domain". In Proceedings of the 17th Annual Conference of the European Association for Machine Translation (EAMT), pp. 221–224.

Virpioja S., Smit, P., Grönroos, S.-A., Kurimo, M. (2013). Morfessor 2.0: Python Implementation and Extensions for Morfessor Baseline. Aalto University publication series SCIENCE + TECHNOLOGY, 25/2013. Aalto University, Helsinki, 2013. ISBN 978-952-60-5501-5.