www.kconnect.eu

D1.6 Toolkit and Report for Translator Adaptation to New Languages (Final Version)

Deliverable number	*D1.6*
Dissemination level	*Public*
Delivery date	*16 September 2016*
Status	*Final*
Author(s)	*Aleš Tamchyna, Jindřich Libovický, Pavel Pecina*

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 644753 (KConnect)

Executive Summary

This document presents the final version of the Machine Translation (MT) training toolkit developed for easier adaptation of the Khresmoi/KConnect MT system to new language pairs. The initial version of the toolkit, Eman Lite, mainly focused on simplification of the MT training steps and resulted in a set of scripts implementing the entire training pipeline which produces statistical models and parameter settings to be plugged into the production system based on MTMonkey, a system for deploying MT as a webservice. The new version of the toolkit, Eman Lite Web, builds on Eman Lite and wraps the training pipeline into an easy-to-use webservice which makes the training even more user-friendly and new languages (language pairs) can be simply added by providing parallel and monolingual training data to Eman Lite Web.

The original Khresmoi MT system supported translation of medical search queries from Czech, German, and French into English and translation of full medical sentences from English into the same languages (Czech, German, and French). Within KConnect, the set of supported languages has been extended by four to include Hungarian, Polish, Spanish, and Swedish. The system now allows translation between all seven (non-English) languages and English in both directions (from English and into English) and both modes (search query translation and sentence translation). This is, in total, 28 combinations. Recently, some of the systems have been improved by retraining on new training data. In this report, we also present evaluation results of those systems.

Table of Contents

List of Abbreviations

(S)MT	(Statistical) Machine Translation
TM	Translation Model
LM	Language Model
EN	English
CS	Czech
DE	German
HU	Hungarian
FR	French
PL	Polish
ES	Spanish
SV	Swedish

1 Introduction

The main role of Machine Translation (MT) in Khresmoi and KConnect was to allow cross-lingual search in medical data. The MT system developed within the Khresmoi project supported translation of medical queries in non-English languages (Czech, French, German) into English and translation of full sentences (summary sentences) in the opposite direction. This allowed the use of non-English queries to search English documents and get back results translated into non-English languages.

The MT technology developed within Khresmoi was built on Moses [3], a state-of-the-art Statistical Machine Translation (SMT) toolkit, using domain adaptation methods based on interpolation of in-domain and out-of-domain models (translation models and language models). Training and deployment of such a system for each language pair was relatively difficult and required a skilled professional to do that. In KConnect, we aim at simplification of this process and develop a toolkit for automated training of SMT systems which can be used to for straightforward addition of new language pairs to the existing system.

The first version of the toolkit (Eman Lite) was described in D1.2 [1]. It is a set of scripts implementing the Moses training pipeline which produces trained models and parameter settings that can be plugged into the production system (based on MTMonkey [4], a system for deploying MT as a webservice developed within Khresmoi and further improved within KConnect[1]). The new version of the toolkit, Eman Lite Web, is built on top of Eman Lite and wraps the training procedure into an easy-to-use webservice.

This report has two main parts. First, in Section 2, we provide a description of Eman Lite Web including user/administration documentation and some recommendations regarding availability of training data for new languages. Second, in Section 3, we report on the new version of the SMT systems retrained on additional training data. The paper is concluded in Section 4 and a list of references provided in Section 5.

2 Eman Lite Web: Translator Adaptation to New Languages

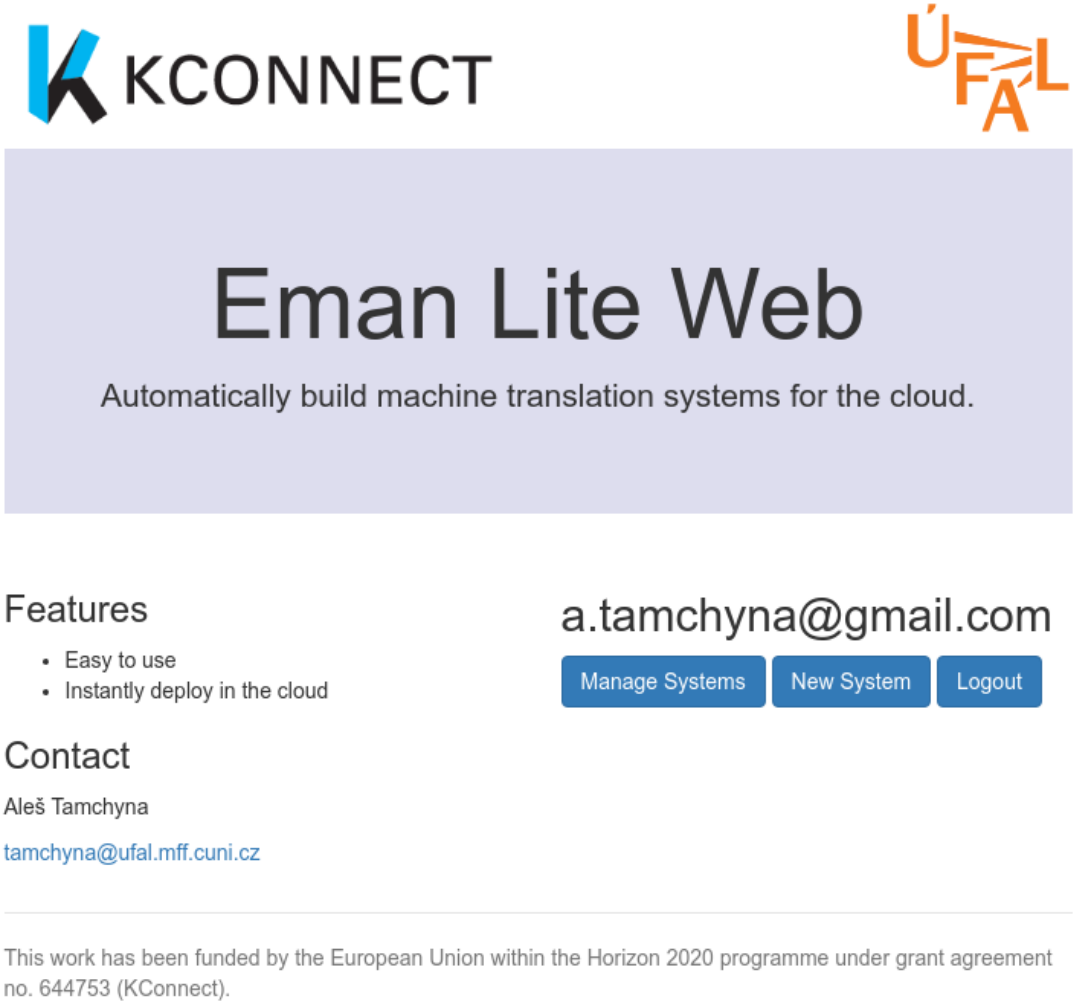
Eman Lite Web is a set of applications which builds on top of Eman Lite (described in Deliverable 1.2 [1]) and provides a web interface and schedule management for automatic training of MT systems. The goal of Eman Lite Web is to allow users to comfortably train an MT system using a web interface. The only input from the user is in a form of training data. The minimum requirement is a set of parallel data (in the source and target language) to train the translation model and language model. Optionally, the user can provide a set of monolingual data in the target language to improve the language model (otherwise the language model is trained on the target side of the parallel data) and a small set of parallel (development) data for optimizing system parameters (also optional, if not provided, the development data is sampled from the parallel training data). When the user provides the data, the new system is scheduled for training. A separate back-end application periodically polls training requests, spawns new instances of Eman Lite and monitors system training progress. Once system training finishes, the system becomes available for download on the webpage. Then, the system models and settings can be plugged into an MTMonkey installation.

[1] https://github.com/ufal/mtmonkey

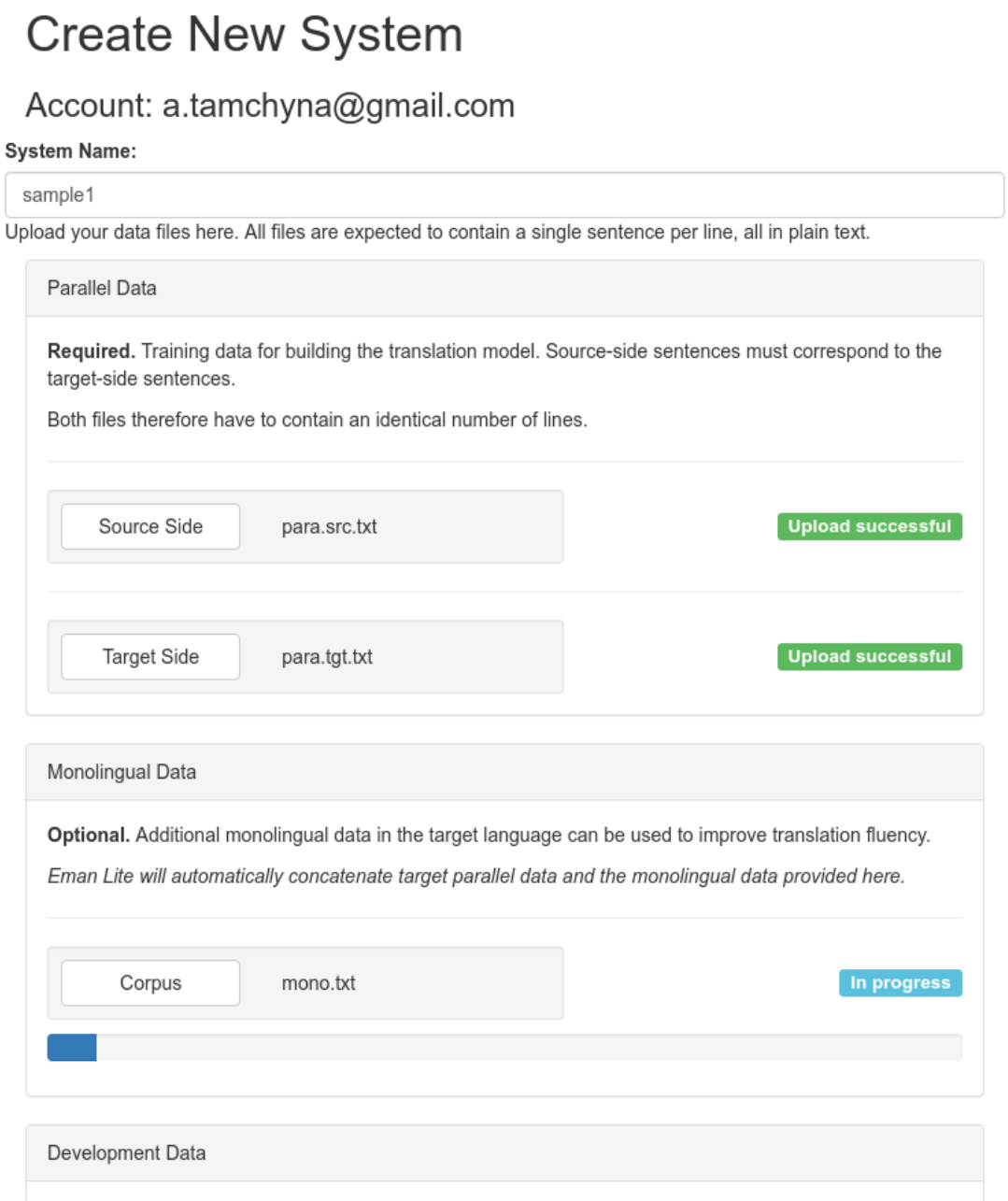
This section provides the documentation of Eman Lite Web. The user part consists of a series of screenshots and description of the individual steps which guide the user through the entire procedure. The administration part helps to install and maintain the system. More details are provided together with the source code.

2.1 Eman Lite Web User Documentation

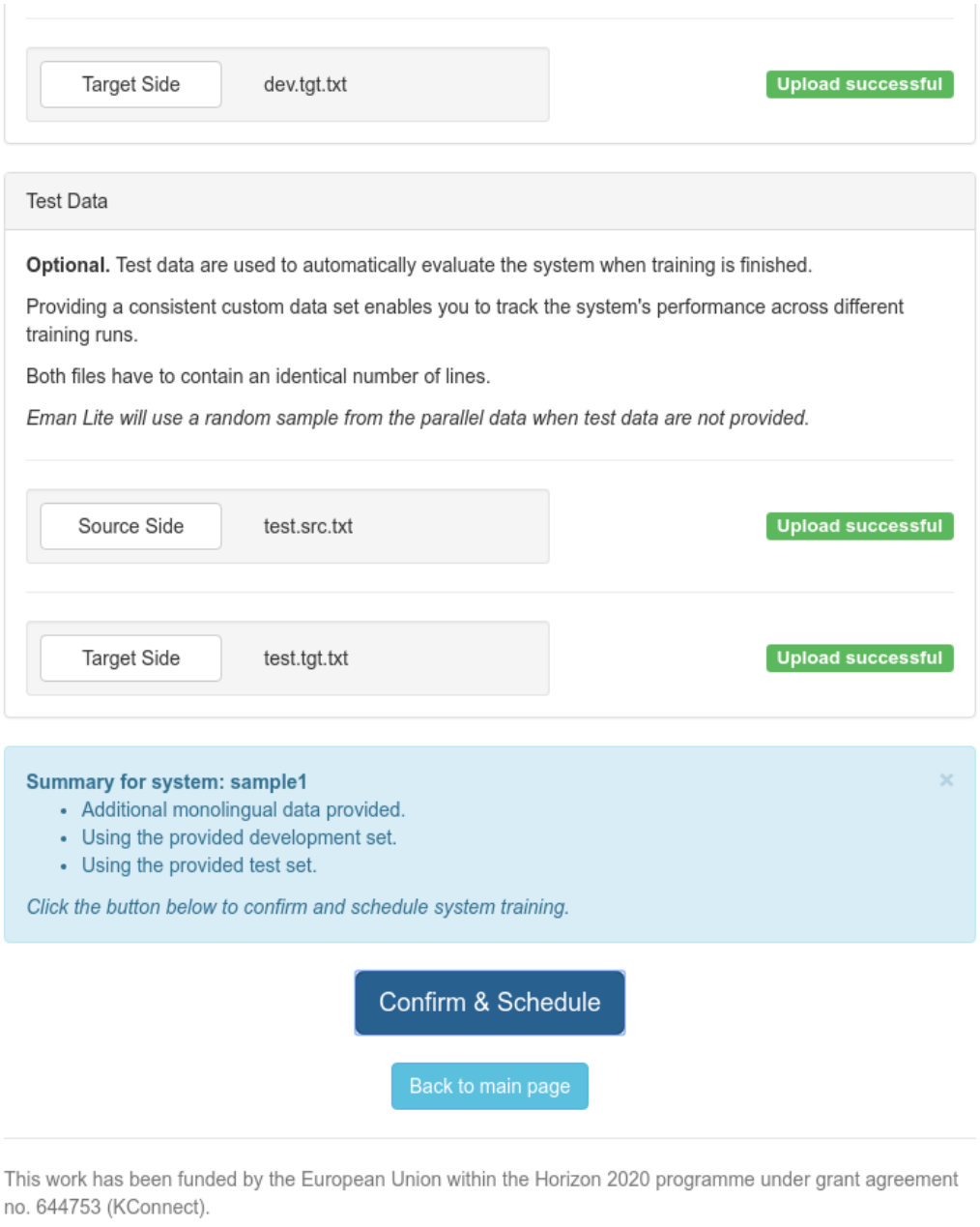
In principle, Eman Lite Web is a dynamic web page with an intuitive user interface. After signing in, the main page displays a menu of actions. You can list your existing systems, create a new submission or sign out again:



To train a system, go to New System. The page displays a list of upload forms for your data files, along with descriptions. Please refer to Deliverable 1.2 for details about file formats (Eman Lite documentation).

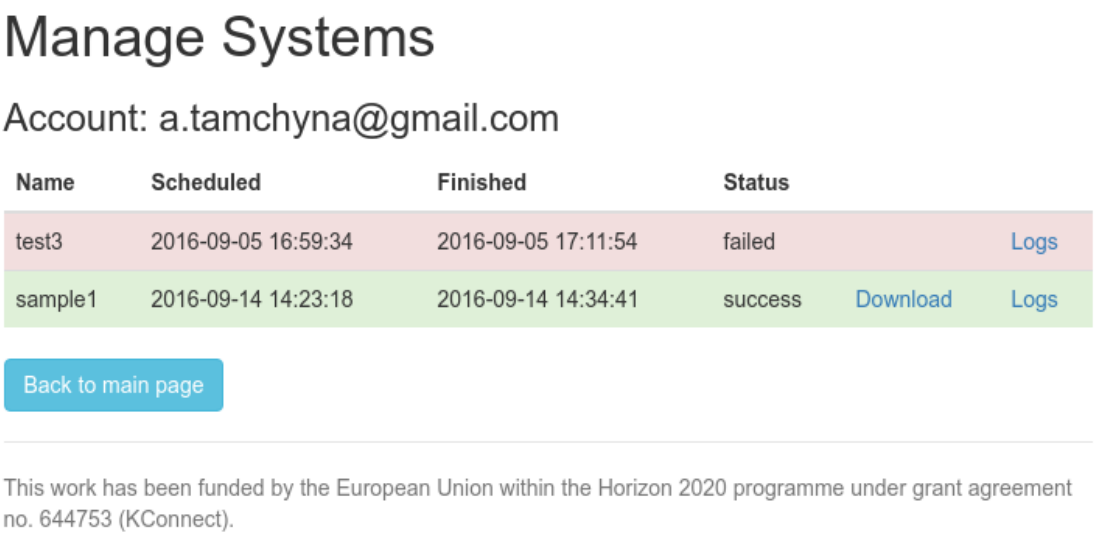


When you have uploaded all data files, click Validate at the bottom of the page. Eman Lite Web will display a confirmation box which summarizes all the information before confirming and scheduling the system training:



This work has been funded by the European Union within the Horizon 2020 programme under grant agreement no. 644753 (KConnect).

After scheduling your system for training, you can go back to the main page and check the training progress in the Manage Systems section:



Note that when a system training run finishes (with either success or failure), training log files become available for inspection. The log files can help you track down errors in your data for cases of training failure. When training succeeds, log files can provide expert users with additional information about settings and system performance.

2.2 Eman Lite Web Administration

The web application is written in Python on top of the Flask framework. It can run on most Linux distributions. It requires the following Python packages:

- flask
- sqlite3
- Flask-Login

As a first step, initialize the sqlite3 database file:

```
sqlite3 database.db < db.schema
```

Next, configure the application. Edit the file `app.cfg` in the application root directory. You can set the following options:

- `MAX_CONTENT_LENGTH` -- maximum file size in bytes
- `UPLOAD_FOLDER` -- where to upload user data
- `SECRET_KEY` -- secret key for the Flask login module
- `DATABASE_PATH` -- path to the sqlite3 database file
- `PORT` -- when used directly with Flask server, on which port to run

Before running the application, note that Flask temporarily stores uploaded files in the system temporary directory. You can change this path by overriding the `TMPDIR` environment variable:

```
export TMPDIR="/path/to/temp/"
```

Once configured, you can run the application using the Flask server. For production use, consider using a standard web server and a WSGI module instead.

```
python app.py
```

2.3 Eman Lite Web Back-End Administration

The web application stores information about training requests in the sqlite3 database file, along with the locations of the uploaded data files. The back-end process, which we describe in this section, periodically polls this database and schedules new systems for training. The back-end keeps track of how many instances of Eman Lite training are currently running and does not allow more than a configured number to run at the same time. When training starts or finished (either with failure or success), the back-end exports the final system tarball and log files from the Eman Lite working directory and updates the database, allowing users of the web application to access the logs and the final system.

The back-end process is a single Python program `backend.py`. It is located in the application root directory.

The program accepts a number of configuration options:

- `--work-dir` -- the working directory
- `--final-dir` -- the directory where trained systems are stored
- `--upload-dir` -- the directory containing uploaded files
- `--eman-lite` -- path to Eman Lite installation
- `--database-file` -- path to the database file shared with the web application
- `--polling-interval` -- polling interval in seconds
- `--max-eman-instances` -- maximum number of Eman Lite instances

The program requires very little administration. Simply configure it by setting the listed options to the correct values on the command line. The recommended use is to run the back-end in the background, e.g. using `nohup`:

```
nohup python backend.py [options] 2>backend.log &
```

2.4 Training Data Recommendation

Eman Lite Web is a further step in simplifying MT system training and makes adding new languages into the KConnect MT system much simpler than before. The training data can be obtained from various sources. An extensive list of potential resources was provided in D1.2 [1] together with a set of scripts for easy extraction/preparation of the training data. Many of the resources are available in multiple languages including those not covered by Khresmoi and KConnect.

In principle, the most important resource is in-domain parallel training data which should include the essential knowledge for mapping medical terminology and phrases from the source language to the target language. Here, especially the EU languages can benefit from existence of public domain data sets such as EMEA[2] and patent collections (COPPA, Corpus Of Parallel Patent Applications provided by World Intellectual Property Organization[3] or the PaTTR collection[4]). A very valuable source of medical domain terminology in multiple languages is the UMLS metathesaurus[5]. Its coverage varies depending on language but has an increasing tendency over the years and new terminologies and their translations are added in each version. If the size of the available in-domain parallel training data is insufficient (i.e. smaller than tens or hundreds of millions of words, which is usually the case), it can be enlarged by exploiting general-domain data, e.g. resources from the European Union (Europarl[6], JRC Acquis[7], and EU Bookshop[8]) or United Nations (e.g., the Multi UN corpus[9]). Community-developed resources, such as Wikipedia[10] or OpenSubtitles[11], should also be considered. Obtaining data from the Web by large-scale web-crawling is another possibility but it should be noted that it requires significant resources (CPU time, network bandwidth, etc.) and specific software (e.g. Bitextor)[12].

Translation quality can be improved by using additional monolingual data in the target language to improve the language model. In general, monolingual data is usually easier to obtain, but if in-domain data is not available in sufficient amounts (hundreds of millions or billions of words), general-domain data can be useful too. A list of such resources for some languages was provided in D1.2, other sources of in-domain data can be probably identified in local/national databases of medical publications.

The third resource needed to train an MT system is a smaller set of parallel data for optimization of system parameters. Within Khresmoi and KConnect, we created such a resource for the seven languages (Czech, French, German, Spanish, Hungarian, Polish, Swedish) and English by manual translation of several hundreds of sentences/queries from English to the other languages. If a similar data set is not available, it can be sampled from the parallel training data which is usually a good alternative. This functionality is built in Eman Lite and Eman Lite Web.

3 New versions of the MT systems

3.1 System description

We adapted the system training pipeline developed within the Khresmoi project to four new languages: Swedish, Spanish, Polish, and Hungarian. For each language, we developed systems which translate in two directions: from English and into English. Additionally, there are two distinct types of documents which require different system optimization: translation of short search queries and translation of summaries. In total, we therefore developed 16 new MT systems. The initial versions of the new systems, with the exception of Polish, are described in D1.4 [2].

[2] http://opus.lingfil.uu.se/
[3] http://www.wipo.int/patentscope/en/data/#coppa
[4] http://www.cl.uni-heidelberg.de/statnlpgroup/pattr/
[5] http://www.nlm.nih.gov/research/umls/
[6] http://www.statmt.org/europarl/
[7] http://opus.lingfil.uu.se/JRC-Acquis.php
[8] http://opus.lingfil.uu.se/EUbookshop.php
[9] http://www.euromatrixplus.net/multi-un/
[10] https://www.wikipedia.org/
[11] http://www.opensubtitles.org/
[12] http://bitextor.sourceforge.net/

We now added additional out-of-domain training data and slightly changed our domain adaptation pipeline. Statistics of all available parallel data for each language pair are given in Table 1.

language pair	En→Spanish	En→Hungarian	En→Polish	En→Swedish
available data (k sentences)	131,651	61,100	56,041	29,116

Table 1: Available parallel training data for the new language pairs.

In the current situation, the parallel training data is very large (tens of millions up to over a hundred million sentence pairs). Given the data size and, crucially, the fact that most of this data is not relevant for the domain at hand, we used the technique described in D1.4 [2] to sort all the parallel sentences based on their similarity to the in-domain data. We then selected 10 million sentence pairs that were the most similar to the medical domain. We carried out this procedure for each language pair and built the MT systems on this data.

With the exception of Spanish, this size of training data is a significant improvement over the initial systems that we described in D1.4. Our training pipeline is otherwise identical to the approach described in D1.4.

3.2 Evaluation

The evaluation was conducted using the same test data sets as described in D1.4 [2] (medical search queries and sentences from summaries of medical articles). The results are presented in Table 2 in terms of BLEU [5] scores and their 5% confidence intervals. The bold figures in blue denote the scores which were improved with respect to the previous versions of the system (the scores in parentheses, presented in D1.4). The Hungarian→English and Swedish→English query translation systems and the English→Swedish summary were improved significantly, the other improvements are only minor. The results of the English→Polish and Polish→English systems have not been presented in D1.4 [2].

BLEU is a standard metric for translation quality evaluation but its scores cannot be meaningfully compared across different language pairs and different test datasets (even though the semantics of the sentences in our test sets is the same). This makes analysis of the results difficult, but we can clearly see that the systems for Polish and Hungarian achieve much lower scores, especially when these languages act as target languages. This can be explained by two facts: 1) Polish and Hungarian are more complex languages (linguistically) and the in-domain training resources are smaller. Increasing the (in-domain) training data size would likely lead to improvement of translation quality and the scores.

direction	query	summary
English→Spanish	35.25±4.89	34.81±1.12
English→Hungarian	**17.36±5.24** (15.64)	**10.14±0.74** (9.92)
English→Polish	**21.78±4.35**	**11.61±0.72**
English→Swedish	33.65±5.83	**39.08±1.21** (37.05)
Spanish→English	43.02±5.29	51.80±1.34
Hungarian→English	**44.40±5.72** (38.16)	21.06±1.08
Polish→English	**29.66±4.96**	**24.62±1.09**
Swedish→English	**51.24±5.41** (44.79)	50.87±1.32

Table 2: Evaluation results of the current systems (BLEU scores with 5% confidence intervals). The bold figures in blue denote improvements compared to the previous results (in parentheses)

4 Conclusions

In this report, we presented the final version of the MT training toolkit Eman Lite Web, which now allows training of a complete MT system via a user-friendly web interface and can be used to add new languages into the KConnect MT system. We also discussed potential sources of training data for such languages.

We also reported evaluation results of the newly trained MT systems. Currently, we provide medical translation service between 7 non-English languages (Czech, French, German, Spanish, Hungarian, Polish, and Swedish) and English. We allow translation of search queries and full text (summaries) in both directions (from English and to English). This is, in total, 28 MT systems. For some of the new languages we were able to further improve translation quality. However, the differences in translation quality among the language pairs still vary a lot. This primarily depends on linguistic properties of the languages and availability of (in-domain) training data. It is expected that with increasing training data size the translation quality will improve.

5 References

[1] Aleš Tamchyna, Jindřich Libovický, Pavel Pecina: D1.2 Toolkit and Report for Translator Adaptation to New Languages (First Version). KConnect deliverable. www.kconnect.eu. 2015.

[2] Aleš Tamchyna, Jindřich Libovický, Pavel Pecina: D1.4: Adaptation to Hungarian, Swedish, and Spanish. KConnect deliverable. www.kconnect.eu. 2016.

[3] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar,

Alexandra Constantin, and Evan Herbst (2007). Moses: open source toolkit for Statistical Machine Translation. In Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, pages 177–180, Prague, Czech Republic.

[4] Aleš Tamchyna, Ondřej Dušek, Rudolf Rosa, and Pavel Pecina. MTMonkey: A Scalable Infrastructure for a Machine Translation Web Service. In The Prague Bulletin of Mathematical Linguistics, No. 100, pp. 31–40, 2013.

[5] Kishore Papineni, Salim Roukos, Todd Ward, and WeiJing Zhu. BLEU: a method for automatic evaluation of Machine Translation. In 40th Annual Meeting on Association for Computational Linguistics, pp. 311–318. 2002