

Technical report: OPENMATREX, a free, open-source hybrid data-driven machine translation system*

Pratyush Banerjee Sandipan Dandapat Mikel L. Forcada[†]
Declan Groves Sergio Penkale John Tinsley
Andy Way

Centre for Next Generation Localisation, School of Computing,
Dublin City University, Glasnevin, Dublin 9, Ireland

Version: Wednesday 15th June, 2011

Abstract

This report describes OPENMATREX, a free/open-source hybrid data-driven machine translation system containing core example-based components based on the marker hypothesis. OPENMATREX comprises a marker-driven chunker, a collection of chunk aligners, tools to merge (“hybridise”) marker-based and statistical translation tables, two engines — a simple proof-of-concept monotone “example-based” recombination engine and a statistical decoder based on Moses —, and support for automatic evaluation. It also contains support for “word packing” to improve alignment. OPENMATREX is a free/open-source release of basic components of MATREX, the Dublin City University machine translation system. The components and processes implemented in OPENMATREX are described in both theoretical and functional detail. Additionally, experimental results are shown in which OPENMATREX is compared to plain statistical machine translation on representative tasks.

1 Introduction

This report describes OPENMATREX, a hybrid data-driven (or *corpus-based*) free/open-source machine translation system containing core example-based components based on the marker hypothesis (Green, 1979). It comprises a marker-driven chunker, a collection of chunk aligners, tools to merge

*This report is an extended version of a preliminary presentation of OPENMATREX at IceTAL (Dandapat et al., 2010), containing a more detailed description of hybridization and of the training and translation processes, and results of additional experiments.

[†]Permanent address: Grup Transducens, Dept. Llenguatges i Sistemes Informàtics, Universitat d’Alacant, E-03071 Alacant, Spain

marker-based and statistical translation tables, two engines — a simple proof-of-concept “example-based” monotone recombination engine (previously released as *Marclator*¹) and a statistical decoder based on Moses² (Koehn et al., 2007) —, and support for automatic evaluation. It also contains support for “word packing” to improve alignment (Ma et al., 2007). OPENMATREX is a free/open-source version of basic components (Stroppa and Way, 2006; Stroppa et al., 2006) of MATREX, the Dublin City University machine translation (MT) system. We call OPENMATREX a *hybrid* system because it has the capability to merge “pure” statistically-derived translation units (*phrase*³ pairs) together with translation units obtained in an “example-based” MT fashion using the marker hypothesis (*chunk* pairs) into a single translation table to be used by the decoder. Most of the code in OPENMATREX is written in Java, although there are many important tasks that are performed in a variety of scripting languages (such as Python or Perl). OPENMATREX has been released through <http://www.openmatrex.org> under a free/open-source licence.⁴ The last version, 0.98, was released on 27th May 2011.

The architecture of OPENMATREX is the same as that of a baseline MATREX system (Stroppa and Way, 2006; Stroppa et al., 2006); as MATREX, it can wrap around the free/open-source Moses statistical MT decoder, using a hybrid translation table containing marker-based chunks as well as statistically extracted *phrase* pairs. In particular, once installed, OPENMATREX provides a simple, user-friendly way of running baseline Moses jobs.

OPENMATREX contains MATREX components which have successfully been used in many researches (Groves and Way, 2005; Hassan et al., 2007; Tinsley et al., 2008; Du et al., 2009; Srivastava et al., 2008; Ma et al., 2008, 2009; Okita et al., 2010; Haque et al., 2009). In particular, using components released in OPENMATREX, researchers have previously:

- used alternative recombination modules to generate translations (Gough and Way, 2004; Way and Gough, 2005), sometimes using statistical models to rerank the results of recombination (Groves and Way, 2006);
- used aligned, marker-based chunks in an alternative decoder which uses a memory-based classifier (van den Bosch et al., 2007);
- combined the marker-based chunkers with rule-based components (Sánchez-Martínez et al., 2009), and used the chunker to filter out Moses phrases for linguistic motivations (Sánchez-Martínez and Way, 2009);

¹<http://www.openmatrex.org/marclator/>

²<http://www.statmt.org/moses/>

³In statistical MT, the term *phrase* is stretched to refer to any contiguous sequence of words, in contrast with the mainstream linguistic interpretation of *phrase* as “any syntactic unit which includes more than one word and is not an entire sentence” (Matthews, 1997).

⁴GNU GPL version 3, <http://www.gnu.org/licenses/gpl.html>

- obtained excellent results in machine translation contests; for instance in WMT-2010 (Callison-Burch et al., 2010), a system using MATREX components for the English–Spanish task (Penkale et al., 2010) ranked 4th in the human evaluation;
- improved machine translation quality by using an advanced word-alignment technique called “word packing” (Ma et al., 2007).

OPENMATREX has been released as a free/open-source package so that these components may be combined with components from other free/open-source machine translation toolkits such as Cunei⁵ (Phillips and Brown, 2009), or Apertium⁶ (Tyers et al., 2009).⁷

To prepare the free/open-source package OPENMATREX, we had to first identify basic usages in MATREX and provide unified support for them. The process included a complete reorganisation of the common code base for MATREX. It also involved bringing together code that had been used by MATREX researchers but was not part of the centralised repository, and identifying possible licensing issues with some of the components. As a result of this process, some bugs have been identified and fixed, marker word files for new languages have been developed, and a revised translation-table merging procedure has been incorporated.

Freeing/open-sourcing MATREX components as part of OPENMATREX not only guarantees the reproducibility of the work performed with them, but also encourages *collaborative research* around it, as anyone can come aboard and improve OPENMATREX or develop new functionality for it. To encourage this collaborative development, the <http://openmatrex.org> website implements a common code repository managed using Subversion⁸ and access to an Internet relay chat (IRC) channel where developers and users may meet.

The rest of the report is organised as follows. Section 2 describes the principles of training and translation in OPENMATREX and briefly reviews existing work performed with its components as part of MATREX; section 3 describes the specific “example-based” components and the hybridization in OPENMATREX; section 4 describes its software requirements and briefly explains how to install and run the available components. Sample experiments performed on representative tasks with OPENMATREX are described in section 5 and results are compared to those obtained with a standard statistical machine translation (SMT) system. Finally, concluding remarks are made in section 6.

⁵<http://www.cunei.org>

⁶<http://www.apertium.org>

⁷For a longer list of free/open-source systems, visit <http://fosmt.info>

⁸Developers may request access to the repository through the website.

2 OpenMaTrEx: Training and Translation

OPENMATREX can be run in two different modes: MATREX *mode* and *Marclator mode*. Within each of these modes the training and translation processes of the system can be performed in different ways. We outline these below.

2.1 Training

When training with OPENMATREX, the following steps are carried out in MATREX *mode* (see the “training” portion of figure 1):

- a. The source side of each sentence pair in the sentence-aligned training corpus and its counterpart in the target side are divided in subsentential segments by a marker-based *chunker* using a set of specific *marker words* for each language (see section 3.1). Chunks are tagged according to the part of speech of marker words during this process and this tag information can be optionally used to further guide the alignment process (see section 3.2).
- b. A complete Moses–GIZA++⁹ (Och and Ney, 2003) training run is performed up to Moses step 5 (phrase extraction) on the sentence-aligned training corpus. Moses is used to learn a maximum-likelihood lexical translation table (steps 1–4) and to extract phrase-pair tables (step 5).
- c. The subsentential chunks are aligned using one of the chunk *aligners* provided (using, among other information, the lexical translation table generated by Moses from the Giza++ alignments: see section 3.2).
- d. Internal word alignments are assigned to each chunk pair, again based on these lexical translation probabilities, in order to allow for the exploitation of the lexical weighting feature of the Moses decoder.
- e. Aligned chunk pairs from step d are *merged* with the phrase pairs generated by Moses in step b (more details in section 3.3) into a single, merged translation table.
- f. From then on, training proceeds as a regular Moses job after Moses step 6. “Minimum error rate training” (MERT) (Och, 2003) — effectively “Maximum BLEU (Papineni et al., 2002) training” — may be used on a development set for tuning.

In *Marclator mode* (see figure 2), the last three steps (d, e, and f) are not necessary and Moses is only run up to step 4.

⁹<http://www.fjoch.com/GIZA++.html>

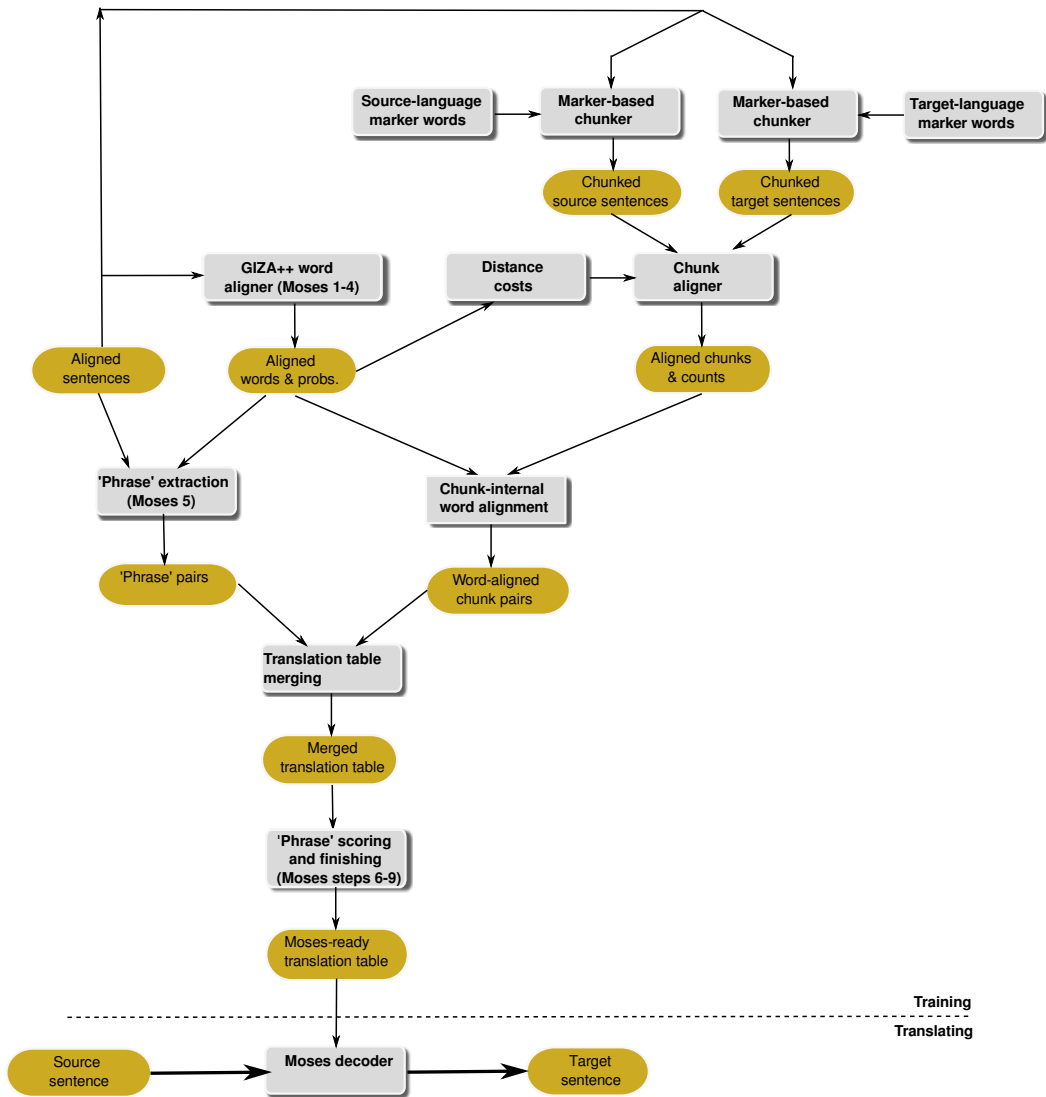


Figure 1: OPENMATREX in "MATREX mode"

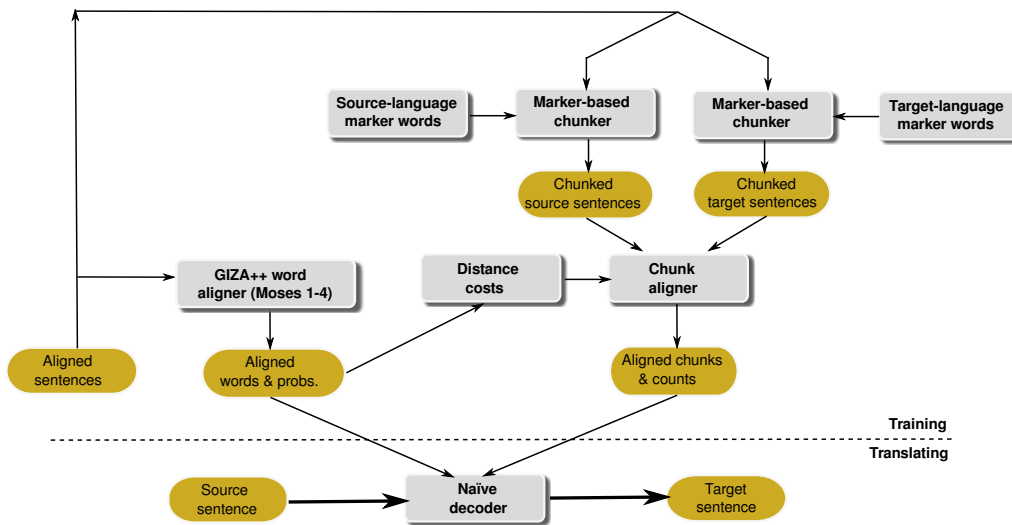


Figure 2: OPENMATREX in “Marclator mode”

2.2 Translation

Translation may be performed, as training, in two ways:

- *Marclator mode*, which uses a monotone (“*naïve*”) “example-based” decoder (previously released as part of Marclator, see the “Translating” part of figure 2), operates as follows:
 - a. Each source sentence is passed through the marker-based chunker.
 - b. The most probable translations for each resulting source-language chunk are retrieved, along with their weights.
 - c. If no chunk translations are found, the decoder backs off to the most likely translations for individual words (as specified in the probabilistic lexical translation tables generated in Moses steps 1 to 4) and concatenates them in the same order as the source input. When no translation is found, source words are left untranslated in the output.

This left-to-right, chunk-by-chunk, greedy decoder has obvious limitations, but it runs orders of magnitude faster than Moses. It may be likely to be of most use in the case of closely related language pairs, but preliminary experiments (see section 5.3) give results which seem too poor to be justifiable in terms of a quality-versus-speed compromise.

- *MATREX mode*, however, is the usual way to use OPENMATREX; that is, the Moses decoder (see the “Translating” part of figure 1) is run on the merged translation table, as is done in MATREX (Stroppa et al., 2006; Stroppa and Way, 2006).

3 “Example-Based” Components and Hybridization

This section defines the components in OPENMATREX that implement an “example-based” approach to the extraction of subsentential translation units (chunk pairs, sections 3.1 and 3.2) and gives details about the hybridization of these “example-based” translation units with statistical “phrase pairs” into a single, merged translation table that will be used by the decoder (section 3.3).

3.1 Chunker

The main chunker in OPENMATREX is based on the *marker hypothesis* (Green, 1979) which states that the syntax of a language is marked at the surface level by a set of marker (closed-category) words or morphemes: the *markers*. In OPENMATREX, marker *words* are used to *chunk* the text:

[He came] [from the office] [to witness] [the chemical process.] (1)
[Vino] [del despacho] [para presenciar] [el proceso químico.]

In English and Spanish, as in example (1) above, markers are predominantly right-facing; they are therefore left-marking languages where, for instance, determiners or prepositions mark the start of noun phrases or prepositional phrases, respectively; there are also right-marking languages such as Japanese, with left-facing markers.¹⁰ The chunker in OPENMATREX currently deals with left-marking languages where markers appear as independent words: a chunk starts at a marker word, and must contain at least one non-marker word. Punctuation is also used to delimit chunks.¹¹

3.1.1 Marker files

Version 0.98 provides marker files for Catalan, Czech, English, Portuguese, Spanish, Irish, French, German, and Italian, as well as preliminary marker files for Breton and Welsh. In the near future, we plan to release marker files for new languages. Marker files specify one marker word or punctuation in each line: its surface form, its category and (optionally) its subcategory. A typical marker word file contains a few hundred entries. Figure 3 shows excerpts of a marker word file for English, which defines chunking (:PUNC:Chunking) and non-chunking (:PUNC:Non_Chunking) punctuation,

¹⁰If marker words in a natural language are taken to be analogous to operators in a programming language, the right-facing markers of left-marking languages would be *prefix* operators, whereas the left-facing markers of right-marking languages would be analogous to *postfix* operators.

¹¹Researchers have used alternative chunkers when using MATREX to deal with languages which are not left-marking (Basque: Stroppa et al. (2006)), or where markers are not independent words (Arabic: Stroppa and Way (2006)).

!:PUNC:Chunking	billion:Q:Card_NUM
”:PUNC:Non_Chunking	both:O:Det_PRON
’:PUNC:Non_Chunking	fifteen:Q:Card_NUM
(:PUNC:Non_Chunking	fifty:Q:Card_NUM
:PUNC:Non_Chunking	five:Q:Card_NUM
[...]	following:O:Det_PRON
I:O:Pers_PRON	[...]
a:D	her:O:Det_PRON
aboard:P	[...]
about:P	that:C:Sub_C
above:P	that:O:Dem_PRON
according to:P	that:O:Det_PRON
[...]	that:O:Wh_PRON
all:O:Det_PRON	[...]
along:P	have:AUX
[...]	’ve:AUX
and:C:Cor_C	having:AUX
another:O:Det_PRON	has:AUX
any:O:Det_PRON	had:AUX
around:P	am:AUX
[...]	’m:AUX
beyond:P	is:AUX

Figure 3: Excerpts of a marker file for English

personal pronouns (:O:Pers_PRON), prepositions (:P), determiners that can act as pronouns (:O:Det_PRON), coordinating (:C:Cor_C) and subordinating conjunctions (:C:Sub_C), cardinal numbers (:Q:Card_NUM), auxiliary verbs (:AUX), etc. Marker files for a particular task may automatically be obtained from existing dictionaries (see section 5.2 for an example).¹²

3.2 Chunk aligners

Chunk aligners obtain chunk pairs, that is, *subsential translation units* from aligned, chunked sentences. In example (1) above:

$$\begin{array}{c}
 \dots \\
 [\mathbf{from\ the\ office}] \leftrightarrow [\mathbf{del\ despacho}] \\
 [\mathbf{to\ witness}] \leftrightarrow [\mathbf{para\ presenciar}] \\
 \dots
 \end{array} \tag{2}$$

There are a number of different chunk aligners available in OPENMA-

¹²In particular, markers may be made to be dependent on the particular language pair: for instance, Czech does not have definite articles such as English has *the*; removing articles from the English marker word file might improve Czech–English alignment.

TREX. The default aligner aligns chunks using a regular Levenshtein edit distance (Wagner and Fischer, 1974) with a combination of costs specified in a configuration file, optionally allowing *jumps* or block movements (Stroppa and Way, 2006), similar to those allowed in evaluation metrics such as TER (Snover et al., 2006) or CDER (Leusch et al., 2006). The default combination uses two costs: a *probability cost* based on word translation probabilities as calculated by using GIZA++ and Moses (see training step b in section 2), and a *cognate cost* based on a combination of the Levenshtein distance (Wagner and Fischer, 1974), the longest common subsequence ratio Hirschberg (1975) and the Dice coefficient (Dice, 1945). An additional cost based on marker tags can also be used during the alignment process. During alignment, if the source and target chunks in question have the same marker category they are given a *tag-based cost* of 0, otherwise they are assigned a *tag-based cost* of 1. As in (Stroppa and Way, 2006), equal weights are used as a default for all component costs specified.

3.3 Translation table merging

To run the system in MATREX *mode*, marker-based chunk pairs are merged with phrase pairs from alternative resources (here, Moses phrases). In order to do this, the chunk pairs must be converted into the same format as the Moses phrases. This is done to ensure compatibility with the decoder in question (following chunk alignment, the marker-based chunk pairs are in the format of Marclator’s monotone decoder, and not Moses): by formatting the chunk pairs in the same way as the Moses phrases (which involves the introduction of supplementary information, described further below), they are able to compete for translations on a level playing field during decoding as they can exploit the same set of features available via the decoder, e.g. lexicalised reordering and lexical weighting, as the Moses phrase pairs.

4 Technical Details

4.1 Required software

Installation of OPENMATREX requires the pre-installation of the following software: GIZA++ for word alignment, Moses for phrase extraction and translation decoding, IRSTLM (Federico and Cettolo, 2007) as a target-language modelling tool kit, and a set of auxiliary scripts for corpus pre-processing.¹³ The current version also requires the presence of the Java-based `args4j`¹⁴ package required for command-line argument parsing within OPENMATREX. Optionally, one can install the machine translation evaluation software Meteor (Lavie and Agarwal, 2007): OPENMATREX will

¹³<http://homepages.inf.ed.ac.uk/jschroe1/how-to/scripts.tgz>

¹⁴<https://args4j.dev.java.net>

compute this measure if it finds the software. For specific details on the installation process refer to the `INSTALL` file that comes with the distribution; an experimental installer for the latest version may be found at <http://www.openmatrex.org>.

4.2 Installing OpenMaTrEx itself

OPENMATREX may easily be built simply by invoking `ant` or an equivalent tool on the `build.xml` provided. The resulting `OpenMaTrEx.jar` contains all the relevant classes, some of which will be invoked using a shell called `OpenMaTrEx` (see below).

4.3 Running

A shell (`OpenMaTrEx`) has options to initialise and filter the training, development, and testing sets, to call the chunker and the aligner, to train a target language model with IRSTLM, to run GIZA++ and Moses training jobs, to merge marker-based chunk pairs with Moses phrase pairs, to run MERT optimisation jobs, to execute the decoders, and to compute MT quality evaluation measures on a test set. Future versions will contain higher-level ready-made options for the most common training and translation jobs. For detailed instructions on how to perform complete training and translation jobs in both MATREX and *Marclator* mode, see the `README` file.¹⁵ Test files are provided in the `examples` directory of the OpenMaTrEx package.

5 Sample experiments

To show how OPENMATREX results compare with baseline SMT results, we report on three experiments, two in *MaTrEx* mode and one in *Marclator* mode.

5.1 A standard task

We have performed a simple experiment in *MaTrEx* mode using 200,000 randomly selected sentences from the Spanish–English Europarl corpus provided for the Third Workshop on SMT (WMT08):¹⁶ testing was performed on the 2,000-sentence test set provided also by WMT08. The experimental conditions replicate those reported by Srivastava et al. (2009). Table 1 shows results for (i) a baseline Moses job, (ii) a job in which marker-based chunk pairs were transformed into Moses translation table pairs as described in section 3 and simply appended to the Moses phrase pairs, and (iii) a third

¹⁵A `sample-run.sh` shell script is provided which performs a plain Moses run and two MaTrEx runs as described in sections 5.1 and 5.2.

¹⁶<http://www.statmt.org/wmt08/>

System	BLEU	Marker-based chunk pairs
Baseline Moses	30.59%	27.60%
Simple merging	30.42%	29.53%
Feature-based merging	30.75%	33.55%

Table 1: A sample experiment using 200,000 randomly-selected sentences from the Spanish–English fraction of Europarl, as provided for the Third Workshop on SMT (WMT08). Testing was performed on the 2,000-sentence test set provided by WMT08.

job in which the extra feature (having the value 1 for marker-based chunk pairs and 0 for Moses-extracted phrase pairs) described in section 3.3 is added to the usual five features in all phrase pairs before MERT tuning. The table shows BLEU scores as well as the fraction of phrase pairs used during translation that were extracted by the marker-based chunker and aligner. Clearly, using the feature-informed phrase table merging improves the BLEU (with 93% statistical significance¹⁷ (Koehn, 2004)), while simple merging does not seem to help. These improvements correlate nicely with the number of marker-based chunks actually used during translation. It would be interesting to pursue a more detailed study of the actual differences in the translations produced when using more linguistically-motivated chunk pairs.

5.2 An experiment with a minor language

We have performed a second experiment in *MaTrEx* mode to explore the benefits of having marker-based chunk pairs when translating from a less-resourced language. To this effect, we have studied translation from Breton (**br**, a less resourced language) to French (**fr**). Markers for **fr** are available in OpenMaTrEx since version 0.9 and markers for **br** have been automatically derived from the **br** morphological dictionaries in revision 23,790 (August 26, 2010) of the **apertium-br-fr** language pair data (Tyers, 2010) in the Apertium free/open-source machine translation platform (Forcada et al., 2009) by generating all surface forms,¹⁸ then extracting those starting with prepositions, coordinative conjunctions, subordinative conjunctions, numer-

¹⁷The free/open-source program FastMTEval by Nicolas Stroppa, http://www.computing.dcu.ie/~nstroppa/softs/fast_mt_eval.tgz was used to compute the statistical significance.

¹⁸The **1t-expand** tool in Apertium was used.

Section	Sentences	br words	fr words
Training	27,989	298,233	308,998
Development	1,000	10,497	10,854
Testing	1,000	10,463	10,910

Table 2: The Ofis ar Brezhoneg fr-br corpus as used for the experiments in section 5.2

als, forms of the verb “to be” and modal verbs,¹⁹ and subsequently removing 131 multi-word units where prepositions were followed by placenames (such as *en Alloz* or *er Waien*) to avoid having whole chunks as marker words; the resulting preliminary marker file is available since version 0.97.

The br-fr corpus used was prepared by Tyers (2009) from materials at the the *Ofis ar Brezhoneg*²⁰ and is available²¹ under the GNU General Public License. The corpus already comes divided in training, development and testing sets as described in Table 2; we have used this configuration for our experiments. The French side of this corpus was also used to train the target language model. Special care had to be taken when tokenizing the br corpus because the sequence *c’h*, representing the [x] sound, must be treated as a single letter.

The results are in table 3. The slight increase in BLEU score observed when adding marker-based chunks seems to be insensitive to whether the additional feature is used or not when merging the translation tables; in fact, the system assigns almost a zero weight to this feature. The statistical significance of those improvements in BLEU with respect to the Moses baseline are 98.8% for simple merging, 99.2% for feature-based merging and 99.8% for Apertium. On the other hand, in these experiments, there does not seem to be apparent correlation between the increase in the BLEU score and the number of marker-based chunks used. The small increase in BLEU may seem disappointing but shows that marker-based chunks may help when a minor language does not have a large parallel corpus available.

5.3 A *Marclator*-mode experiment on a pair of related languages

We have also performed an experiment to check the usefulness of the *Marclator* mode of OPENMATREX for a pair of related languages. A bilingual

¹⁹This method may be used generally to extract a preliminary set of OPENMATREX marker words from an Apertium dictionary. Automatic extraction of markers from dictionaries has also been used in MATREX: Owczarzak et al. (2006) extracted them from CELEX (Baayen et al., 1996) dictionaries.

²⁰Office of the Breton Language, <http://www.ofis-bzh.org/>

²¹http://elx.dlsi.ua.es/~fran/brfr_OAB_corpus.tgz

System	BLEU	Marker-based chunk pairs
Baseline Moses	38.84%	34.09%
Simple merging	39.78%	37.96%
Feature-based merging	39.78%	26.88%
Apertium-br-fr	17.10%	Not applicable

Table 3: A sample experiment with Breton (a less-resourced language) and French using 27,989 phrase pairs from the *Ofis ar Brezhoneg* corpus (Tyers et al., 2009), a small, free bilingual corpus.

Section	Sentences	es words	ca words
Training	819,205	14,428,738	15,003,960
Development	999	17,211	17,948
Testing	999	17,186	17,865

Table 4: The Spanish–Catalan corpus from *El Periódico de Catalunya* used to test the *Marclator* mode of OPENMATREX (see section 5.3)

corpus of Spanish–Catalan newspaper text harvested from the website of *El Periódico de Catalunya* (see table 4) was used to train OPENMATREX in *Marclator* mode. The crude nature of the left-to-right, greedy decoding process in *Marclator* mode prevents it from getting acceptable results, as shown in figure 5: the BLEU score obtained by *Marclator* is clearly very low when compared to the BLEU score of a baseline Moses run on the same corpus. Note also the result obtained with Apertium, a rule-based system, with a BLEU score which is clearly lower than the Moses result, perhaps due to the general-purpose nature of Apertium vocabularies as compared to the specialization attained by Moses on the newspaper texts used both for training and testing.

System	BLEU
<i>Marclator</i> mode	56.57%
Apertium-es-ca	77.66%
Moses baseline	85.71%

Table 5: Results of *Marclator* experiments on the Spanish–Catalan corpus

6 Concluding Remarks and Future Work

We have presented OPENMATREX, a hybrid data-driven free/open-source machine translation system including a marker-driven chunker, a collection of chunk aligners, tools to merge (“hybridise”) marker-based and statistical translation tables, two engines —a simple proof-of-concept monotone “example-based” recombination engine and a Moses-based statistical decoder, so that it can be used as a decoder for a merged translation table containing Moses phrases and marker-based chunk pairs—, and support for automatic evaluation. OPENMATREX releases basic components of MATREX, the Dublin City University machine translation system, under a free/open-source license, and makes them available to researchers and developers of MT systems.

Experimental results in MATREX mode, one with a pair of *major* languages and another one with a pair including a *minor* language, show that the hybridisation in the *MaTrEx* mode of OPENMATREX, that is, merging the translation table obtained by aligning marker-based chunks (the “example-based” table) with the translation table obtained by Moses using statistical phrase extraction may lead to an improved translation quality (as measured with BLEU) with respect to that obtained by using only the statistical translation table. Results for the *Marclator* mode show that the reduction in quality may not be justified by the speed increase obtained with its monotone, greedy decoder.

As for future work, version 1.0 of OPENMATREX will contain, among other improvements, a better set of marker files, improved installing and running procedures with extensive training and testing options, and improved documentation; further versions are expected to free/open-source additional MATREX components. We should also continue to promote the creation of a real community of developers and users around OPENMATREX, such as the one around other free/open-source machine translation projects such as Moses (Koehn et al., 2007), Apertium (Tyers et al., 2009), etc.

Acknowledgements: The original MATREX code on which OPENMATREX is based was developed among others by S. Armstrong, Y. Graham, N. Gough, D. Groves, H. Hassan, Y. Ma, B. Mellebeek, N. Stroppa, J. Tinsley, and A. Way. We specially thank Y. Graham and Y. Ma for their advice. S. Ebling built the German marker file. P. Pecina helped with Czech markers and Jim O’Regan with Irish markers. We thank Felipe Sánchez-Martínez for useful comments on the manuscript. M.L. Forcada’s sabbatical stay at Dublin City University was supported by Science Foundation Ireland (SFI) through ETS Walton Award 07/W.1/I1802 and by the Universitat d’Alacant (Spain). Support from SFI through grants 05/IN/1732, 06/RF/CMS064 and 07/CE/I1142 is also gratefully acknowledged.

References

- Baayen, R., Piepenbrock, R., and Gulikers, L. (1996). Celex2. Linguistic Data Consortium, catalog number #LDC96L14.
- Callison-Burch, C., Koehn, P., Monz, C., Peterson, K., Przybocki, M., and Zaidan, O. (2010). Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and Metrics-MATR*, pages 17–53.
- Dandapat, S., Forcada, M., Groves, D., Penkale, S., Tinsley, J., and Way, A. (2010). OpenMaTrEx: A Free/Open-Source Marker-Driven Example-Based Machine Translation System. *Advances in Natural Language Processing*, pages 121–126.
- Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, 26(3).
- Du, J., He, Y., Penkale, S., and Way, A. (2009). MaTrEx: the DCU MT System for WMT 2009. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 95–99, Athens, Greece.
- Federico, M. and Cettolo, M. (2007). Efficient handling of n-gram language models for statistical machine translation. In *Proc. of the 2nd Workshop on Statistical Machine Translation*, pages 88–95, Prague, Czech Rep.
- Forcada, M., Tyers, F., and Ramírez-Sánchez, G. (2009). The Apertium machine translation platform: five years on. In *Proceedings of the First International Workshop on Free/Open-Source Rule-Based Machine Translation*, pages 3–10.
- Gough, N. and Way, A. (2004). Robust large-scale EBMT with marker-based segmentation. In *Proc. of the 10th Conf. on Theoretical and Methodological Issues in Machine Translation (TMI-04)*, pages 95–104, Baltimore, MD.
- Green, T. (1979). The necessity of syntax markers. Two experiments with artificial languages. *Journal of Verbal Learning and Behavior*, 18:481–496.
- Groves, D. and Way, A. (2005). Hybrid example-based SMT: the best of both worlds? *ACL-2005 Workshop on Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond*, 100:183–190.
- Groves, D. and Way, A. (2006). Hybridity in MT: Experiments on the Europarl corpus. In *Proc. of the 11th Ann. Conf. of the European Association for Machine Translation (EAMT-2006)*, pages 115–124, Oslo, Norway.

- Haque, R., Dandapat, S., Srivastava, A. K., Naskar, S. K., and Way, A. (2009). English-Hindi Transliteration Using Context-Informed PB-SMT: the DCU System for NEWS 2009. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*, pages 104–107, Singapore.
- Hassan, H., Ma, Y., Way, A., and Dublin, I. (2007). MaTrEx: the DCU machine translation system for IWSLT 2007. In *Proc. of IWSLT 2007*, pages 69–75, Trento, Italy.
- Hirschberg, D. (1975). A linear space algorithm for computing maximal common subsequences. *Communications of the ACM*, 18(6):343.
- Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*, volume 4, pages 388–395.
- Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. *Ann. Meeting of the Association for Computational Linguistics (ACL), demonstration session, Prague, Czech Republic, June*, pages 177–180.
- Lavie, A. and Agarwal, A. (2007). METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231. Association for Computational Linguistics.
- Leusch, G., Ueffing, N., and Ney, H. (2006). CDER: Efficient MT evaluation using block movements. In *Proceedings of the Thirteenth Conference of the European Chapter of the Association for Computational Linguistics*, pages 241–248.
- Ma, Y., Okita, T., Özlem Çetinoğlu, Du, J., and Way, A. (2009). Low-Resource Machine Translation Using MaTrEx: The DCU Machine Translation System for IWSLT 2009. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 29–36, Tokyo, Japan.
- Ma, Y., Stroppa, N., and Way, A. (2007). Bootstrapping word alignment via word packing. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 304–311.
- Ma, Y., Tinsley, J., Hassana, H., Du, J., and Way, A. (2008). Exploiting Alignment Techniques in MaTrEx: the DCU Machine Translation System for IWSLT08. In *Proceedings of International Workshop on Spoken Language Translation (IWSLT)*, pages 26–33, Honolulu, HI, USA.

- Matthews, P. H. (1997). *Oxford Concise Dictionary of Linguistics*. Oxford Univ. Press.
- Och, F. (2003). Minimum error rate training in statistical machine translation. In *Proc. 41st Ann. Meeting of the Association for Computational Linguistics-Volume 1*, pages 160–167, Sapporo, Japan.
- Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.
- Okita, T., Jiang, J., Haque, R., Al-Maghout, H., Du, J., Naskar, S., and Way, A. (2010). MaTrEx: the DCU MT System for NTCIR-8. In *Proceedings of NTCIR-8 Workshop Meeting*, pages 377–383, Tokyo, Japan.
- Owczarzak, K., Mellebeek, B., Groves, D., van Genabith, J., and Way, A. (2006). Wrapper syntax for example-based machine translation. In *AMTA 2006 - 7th Conference of the Association for Machine Translation of the Americas*, pages 148–155, Cambridge, Massachusetts, USA.
- Papineni, K., Roukos, S., Ward, T., and Zhu, W. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.
- Penkale, S., Haque, R., Dandapat, S., Banerjee, P., Srivastava, A. K., Du, J., Pecina, P., Naskar, S. K., Forcada, M. L., and Way, A. (2010). MATREX: The DCU MT System for WMT 2010. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 149–154, Uppsala, Sweden.
- Phillips, A. and Brown, R. (2009). Cunei machine translation platform: System description. In *Proc. of the 3rd Workshop on Example-Based Machine Translation*, pages 29–36, Dublin, Ireland.
- Sánchez-Martínez, F., Forcada, M., and Way, A. (2009). Hybrid rule-based – example-based MT: Feeding Apertium with sub-sentential translation units. In *Proc. of the 3rd Workshop on Example-Based Machine Translation*, pages 11–18, Dublin, Ireland.
- Sánchez-Martínez, F. and Way, A. (2009). Marker-based filtering of bilingual phrase pairs for SMT. In *Proc. of EAMT-09, the 13th Ann. Meeting of the European Association for Machine Translation*, pages 144–151, Barcelona, Spain.
- Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231. Citeseer.

- Srivastava, A., Penkale, S., Groves, D., and Tinsley, J. (2009). Evaluating syntax-driven approaches to phrase extraction for MT. In *Proc. of the 3rd Workshop on Example-Based Machine Translation*, pages 19–28, Dublin, Ireland.
- Srivastava, A. K., Haque, R., Naskar, S. K., and Way, A. (2008). MaTrEx: The DCU MT System for ICON 2008. In *Proceedings of the NLP Tools Contest: Statistical Machine Translation (English to Hindi), 6th International Conference on Natural Language Processing (ICON-2008)*, Pune, India.
- Stroppa, N., Groves, D., Way, A., and Sarasola, K. (2006). Example-based machine translation of the Basque language. In *Proc. of AMTA 2006*, pages 232–241, Cambridge, MA, USA.
- Stroppa, N. and Way, A. (2006). MaTrEx: DCU machine translation system for IWSLT 2006. In *Proceedings of IWSLT 2006*, pages 31–36.
- Tinsley, J., Ma, Y., Ozdowska, S., and Way, A. (2008). MaTrEx: the DCU MT system for WMT 2008. In *Proc. of the Third Workshop on Statistical Machine Translation*, pages 171–174, Waikiki, HI.
- Tyers, F., Forcada, M., and Ramírez-Sánchez, G. (2009). The Apertium machine translation platform: Five years on. In *Proc. of the First Intl. Workshop on Free/Open-Source Rule-Based Machine Translation*, pages 3–10, Alacant, Spain.
- Tyers, F. M. (2009). Rule-based augmentation of training data in Breton–French statistical machine translation. In *Proc. of the 13th Annual Conf. of the European Association of MT, EAMT09*, pages 213–218.
- Tyers, F. M. (2010). Rule-based Breton to French machine translation. In Hansen, V. and Yvon, F., editors, *Proceedings of the 14th Annual Conference of the European Association of Machine Translation, EAMT10*, pages 174–181.
- van den Bosch, A., Stroppa, N., and Way, A. (2007). A memory-based classification approach to marker-based EBMT. In *Proc. of the METIS-II Workshop on New Approaches to Machine Translation*, pages 63–72, Leuven, Belgium.
- Wagner, R. and Fischer, M. (1974). The string-to-string correction problem. *Journal of the ACM (JACM)*, 21(1):168–173.
- Way, A. and Gough, N. (2005). Comparing example-based and statistical machine translation. *Natural Language Engineering*, 11(03):295–309.