To be presented at IceTAL (Reykjavík, 16-18 Aug. 2010)

OpenMaTrEx: A Free/Open-Source Marker-Driven Example-Based Machine Translation System

Sandipan Dandapat, Mikel L. Forcada*, Declan Groves**, Sergio Penkale, John Tinsley, and Andy Way

Centre for Next Generation Localisation, School of Computing,
Dublin City University, Glasnevin, Dublin 9, Ireland
{sdandapat,mforcada,dgroves,spenkale,jtinsley,away}@computing.dcu.ie

Abstract. We describe OpenMaTrEx, a free/open-source example-based machine translation (EBMT) system based on the marker hypothesis, comprising a marker-driven chunker, a collection of chunk aligners, and two engines: one based on a simple proof-of-concept monotone EBMT recombinator and a Moses-based statistical decoder. OpenMaTrEx is a free/open-source release of the basic components of MaTrEx, the Dublin City University machine translation system.

Keywords: example-based machine translation, corpus-based machine translation, free/open-source software

1 Introduction

We describe OpenMaTrEx, a free/open-source (FOS) example-based machine translation (EBMT) system based on the marker hypothesis [1]. It comprises a marker-driven chunker, a collection of chunk aligners, and two engines: one based on the simple proof-of-concept monotone recombinator (previously released as *Marclator*[1]) and a Moses-based decoder [2]. OpenMaTrEx is a FOS version of the basic components of MaTrEx, the Dublin City University machine translation (MT) system [3, 4]. 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. A preliminary version, 0.71, has been released for download from `http://www.openmatrex.org` on 2nd June 2010, under a FOS licence.[2]

The architecture of OpenMaTrEx is the same as that of a baseline MaTrEx system [3, 4]; as MaTrEx, it can wrap around the Moses statistical MT

* Permanent address: Departament de Llenguatges i Sistemes Informàtics, Universitat d'Alacant, E-03071 Alacant, Spain
** Also: Traslán Teoranta, 31 Mill Grove, Killincarrig, Greystones, Co. Wicklow, Ireland
[1] `http://www.openmatrex.org/marclator/`
[2] GNU GPL version 3, `http://www.gnu.org/licenses/gpl.html`

decoder, using a hybrid translation table containing marker-based chunks as well as statistically extracted *phrase*[3] pairs.

OpenMaTrEx has been released as a FOS package so that MaTrEx components which have successfully been used [5–7] may be combined with components from other FOS machine translation (FOSMT) toolkits such as Cunei[4] [8], Apertium[5] [9], etc.[6] Indeed, using components released in OpenMaTrEx, researchers have previously: used statistical models to rerank the results of recombination [10]; used aligned, marker-based chunks in an alternative decoder which uses a memory-based classifier [11]; combined the marker-based chunkers with rule-based components [12], and used the chunker to filter out Moses phrases for linguistic motivations [13].

The rest of the paper is organized as follows. Section 2 describes the principles of training and translation in OpenMaTrEx; section 3 describes the EBMT-specific components in OpenMaTrEx; section 4 describes its software requirements and briefly explains how to install and run the available components. A sample experiment performed on a standard task with OpenMaTrEx is described in section 5 and results are compared to those obtained with a a standard statistical machine translation (SMT) system. Concluding remarks are made in section 6.

2 OpenMaTrEx: Training and Translation

Training with OpenMaTrEx may be performed in two different modes. In MaTrEx *mode:*

1. Each example sentence in the sentence-aligned source text and its counterpart in the target training text are divided in subsentential segments using a marker-based *chunker*. Chunks may optionally be tagged according to their initial marker word (to further guide the alignment process).
2. A complete Moses–GIZA++[7] training run is performed up to Moses step 5 (phrase extraction). Moses is used to learn a maximum-likelihood lexical translation table and to extract phrase-pair tables.
3. The subsentential chunks are aligned using one of the *aligners* provided (using, among other information, probabilities generated by GIZA++).
4. Aligned chunk pairs from step 3 are *merged* with the phrase pairs generated by Moses in step 2 (more details in section 3).
5. From then on, training proceeds as a regular Moses job after Moses step 6. MERT [14] may be used on a development set for tuning.

In *Marclator* mode (see below), the last two steps are not necessary and Moses is only run up to step 4.

[3] In statistical MT, the term *phrase* is stretched to refer to any contiguous sequence of words.
[4] `http://www.cunei.org`
[5] `http://www.apertium.org`
[6] For a longer list of FOSMT systems, visit `http://fosmt.info`
[7] `http://www.fjoch.com/GIZA++.html`

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

- *Marclator mode* uses a monotone ("*naïve*") decoder (released as part of Marclator): each source sentence is run through the marker-based chunker; the most probable translations for each chunk are retrieved, along with their weights; if no chunk translations are found, the decoder backs off to the most likely translations for words (as aligned by GIZA++) and concatenates them in the same order, and when no translation is found, leaves any unfound source words untranslated. This decoder has obvious limitations, but it is fast and likely to be of most use in the case of closely related language pairs.
- MaTrEx *mode*, however, is the usual way to use OpenMaTrEx; that is, the Moses decoder is run on a merged phrase table, as in MaTrEx [3, 4].

3 EBMT-Specific Components

Chunker: The main chunker in OpenMaTrEx is based on the *marker hypothesis* [1] 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 chunker in OpenMaTrEx deals with left-marking languages: a chunk starts at a marker word, and must contain at least one non-marker word. Punctuation is also used to delimit chunks. Version 0.71 provides marker files for Catalan, Czech, English, Portuguese, Spanish, Irish, French and Italian. 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.

Chunk aligners: There are a number of different chunk aligners available in OpenMaTrEx. The default aligner aligns chunks using a regular Levenshtein edit distance with a combination of costs specified in a configuration file, optionally allowing *jumps* or block movements [3]. The default combination uses two costs: a *probability cost* based on word translation probabilities as calculated by using GIZA++ and Moses (see training step 2 in section 2), and a *cognate cost* based on a combination of the Levenshtein distance, the longest common subsequence ratio and the Dice coefficient. As in [3], equal weights are used as a default for all component costs specified.

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). Firstly, each chunk pair is assigned a word alignment based on the refined GIZA++ alignments, for example "`please show me ||| por favor muéstreme ||| 0-0 0-1 1-2 2-2`". In cases where there is no word alignment for a particular chunk pair according to GIZA++, the chunk pair is discarded. Using these word alignments, we additionally extract a phrase orientation-based lexicalised reordering model *à la* Moses [15]. Finally, we may also limit the maximum length of chunks pairs that will be used. The resulting chunk pairs are in the same format as those phrase pairs extracted by Moses. The next step is to combine the chunk pairs with Moses phrase pairs. In order to do this, the

two sets of chunk/phrase pairs are merged into a single file. Moses training is then carried out from step 6 (*scoring*) which calculates the required scores for all feature functions, including the reordering model, based on the combined counts. A binary feature distinguishing EBMT chunks from SMT chunks may be added for subsequent MERT optimization as was done in [16].

4 Technical Details

Required software: OpenMaTrEx requires the installation of the following software: GIZA++, Moses, IRSTLM [17], and a set of auxiliary scripts for corpus preprocessing[8] and evaluation (mteval).[9] Refer to the `INSTALL` file that comes with the distribution for details.

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, `OpenMaTrEx` (see below).

Running: A shell (`OpenMaTrEx`) has options to initialise 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 optimization jobs, and to execute the decoders. 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. Test files will be provided in the `examples` directory of the OpenMaTrEx package.

5 A Sample Experiment

To show how OpenMaTrEx can be used to improve baseline SMT results, we report on a simple experiment using 200,000 randomly selected sentences from the Spanish–English Europarl corpus provided for the Third Workshop on SMT (WMT08): testing was performed on the 2,000-sentence test set provided by WMT08. The experimental conditions are the same as those reported in [16]. 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 job in which an extra feature (having the value 1 for marker-based chunk pairs and 0 for Moses-extracted phrase pairs) is added to the usual five features in all phrase pairs before MERT tuning. The table shows BLEU and NIST scores as well as the fraction of phrase pairs used during translation that were extracted

[8] `http://homepages.inf.ed.ac.uk/jschroe1/how-to/scripts.tgz`
[9] We currently use version 11b from `ftp://jaguar.ncsl.nist.gov/mt/resources/` .

by the marker-based chunker and aligner. Clearly, using the feature-informed phrase table merging improves the BLEU (with 93% statistical significance [18]) and NIST scores (76% confidence), 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.

System	BLEU	NIST	EBMT pairs
Baseline Moses	30.59%	7.5171	27.60%
Simple merging	30.42%	7.5156	29.53%
Feature-based merging	30.75%	7.5269	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.

6 Concluding Remarks and Future Work

We have presented OpenMaTrEx, a FOS EBMT system including a marker-driven chunker (with marker word files for a few languages), chunk aligners, a simple monotone recombinator, and a wrapper around Moses so that it can be used as a decoder for a merged translation table containing Moses phrases and marker-based chunk pairs. OpenMaTrEx releases the basic components of MaTrEx, the Dublin City University machine translation system under a FOS license, to make them available to researchers and developers of MT systems.

As for future work, version 1.0 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.

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. P. Pecina helped with Czech markers and Jim O'Regan with Irish markers. M.L. Forcada's sabbatical stay at Dublin City University is 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 grant 07/CE/I1142 is acknowledged.

References

1. T. Green. The necessity of syntax markers. two experiments with artificial languages. *Journal of Verbal Learning and Behavior*, 18:481–496, 1979.

2. P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 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, 2007.
3. N. Stroppa and A. Way. MaTrEx: DCU machine translation system for IWSLT 2006. In *Proceedings of IWSLT 2006*, pages 31–36, 2006.
4. N. Stroppa, D. Groves, A. Way, and K. Sarasola. Example-based machine translation of the Basque language. In *Proc. of AMTA 2006*, pages 232–241, Cambridge, MA, USA, 2006.
5. D. Groves and A. Way. 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, 2005.
6. H. Hassan, Y. Ma, A. Way, and I. Dublin. MaTrEx: the DCU machine translation system for IWSLT 2007. In *Proc. of IWSLT 2007*, pages 69–75, Trento, Italy, 2007.
7. J. Tinsley, Y. Ma, S. Ozdowska, and A. Way. MaTrEx: the DCU MT system for WMT 2008. In *Proc. of the Third Workshop on Statistical Machine Translation*, pages 171–174, Waikiki, HI, 2008.
8. A.B. Phillips and R.D. Brown. Cunei machine translation platform: System description. In *Proc. of the 3rd Workshop on Example-Based Machine Translation*, pages 29–36, Dublin, Ireland, Nov. 2009.
9. F.M. Tyers, M.L. Forcada, and G. Ramírez-Sánchez. 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, Nov. 2009.
10. D. Groves and A. Way. 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, 2006.
11. A. van den Bosch, N. Stroppa, and A. Way. 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, 2007.
12. F. Sánchez-Martínez, M.L. Forcada, and A. Way. 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, Nov. 2009.
13. Felipe Sánchez-Martínez and Andy Way. 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, 2009.
14. F.J. Och. 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, 2003.
15. P. Koehn, A. Axelrod, A.B. Mayne, C. Callison-Burch, M. Osborne, and D. Talbot. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proc. of IWSLT 2005*, Pittsburgh, PA, 2005.
16. A. Srivastava, S. Penkale, D. Groves, and J. Tinsley. 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, November 2009.
17. M. Federico and M. Cettolo. 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., 2007.
18. P. Koehn. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*, volume 4, pages 388–395, 2004.