Combining Semantic and Syntactic Generalization in Example-Based Machine Translation

Sarah Ebling[*], Andy Way[**], Martin Volk[*], Sudip Kumar Naskar[**]
[*] Institute of Computational Linguistics, University of Zurich
{ebling,volk}@ifi.uzh.ch
[**] CNGL, School of Computing, Dublin City University
{away,snaskar}@computing.dcu.ie

May 30, 2011

Outline

Example-Based Machine Translation

Generalized Templates in EBMT

EBMT at Dublin City University

EBMT at Carnegie-Mellon University

Experiments
- Our Approach
- Results
- Discussion
- Conclusion

Example-Based Machine Translation I

- Example-Based Machine Translation (EBMT): instance of Corpus-Based Machine Translation (CBMT), like SMT
- main difference between SMT and EBMT: type of knowledge that is consulted at runtime:
  SMT systems consult probabilities of source language–target language (SL–TL) word or phrase pairs which they have learned from the training data offline
  EBMT systems consult the training set (their example base) directly
- EBMT systems have often performed worse than SMT systems in the past

Example-Based Machine Translation II

- biggest shortcoming of EBMT: does not combine translations of phrases well (boundary friction)
  particularly frequent when translating into a morphologically rich language (e.g., into German)

Example

sentence pairs in an example base, Way (2001):
A big dog *eats a lot of meat.* – **Ein großer Hund** *frisst viel Fleisch.*
I have *two ears.* – **Ich habe** *zwei Ohren.*

EBMT system might make use of phrases shown in bold to translate a sentence like *I have a big dog.* into *Ich habe ein großer Hund.* ⟶ would neglect the fact that German uses different inflectional forms to mark grammatical case: German phrase *ein großer Hund* is a nominative noun phrase, but *Ich habe* requires accusative object (*einen großen Hund*)

Generalized Templates in EBMT I

- among the best-performing systems in EBMT: systems that make use of generalized templates
- generalized templates: SL–TL pairs in which certain parts have been replaced by variables
- provide an additional layer of abstraction and can prevent a system from having to revert to word-by-word translation
- our experiments: combining two existing EBMT systems that rely on generalized templates → improvement over the individual performances of these two systems? → no, but improvements over a lexical EBMT system

Generalized Templates in EBMT II

- generalized templates: risk of overgeneralizing (replacing too many parts of an SL–TL pair with variables) → templates are usually restricted to certain categories of words
- semantic generalization: Kitamura and Matsumoto (1995)
- syntactic generalization: Güvenir and Tunc (1996), Kaji et al. (1992)
- generalization over sequences of words: Cicekli and Güvenir (2001)

Systems Used for Experiments

two systems, both started out as purely lexical EBMT systems, i. e., did not make use of generalized templates

extensions:

1. *Marclator*: generalization over function words
2. *CMU-EBMT*: semantic and syntactic generalization

EBMT at Dublin City University: *Marclator*

- developed at Dublin City University (DCU), part of the *MaTrEx* architecture (Stroppa and Way, 2006) (`http://www.openmatrex.org/marclator/marclator.html`)
- modules: chunking, word alignment, chunk alignment, recombination

Chunking in *Marclator* I

- system segments both the training and the test data into chunks
- chunking is based on the Marker Hypothesis (Green, 1979): psycholinguistic hypothesis stating that every language has a closed set of elements that are used to mark certain syntactic constructions
- set of elements includes function words and bound morphemes (*-ing* as an indicator of English progressive-tense verbs)
- *Marclator* chunking module solely considers function words as indicators of chunk boundaries

Chunking in *Marclator* II

- each function word (*Marker word*) triggers the opening of a new chunk, provided that the preceding chunk contains at least one non-Marker word

Example

e. g., *He was* | **on** *the bus*

Chunking in *Marclator* III

Category	Example
determiner	*den*
personal pronoun	*euch*
demonstrative pronoun	*jenem*
possessive pronoun	*seine*
interrogative pronoun	*welch*
indefinite pronoun	*andere*
relative pronoun	*denen*
preposition	*abseits*
coordinative conjunction	*aber*
subordinative conjunction	*falls*
cardinal numeral	*eins*
numeric expression	*neunundneunzig*
auxiliary/modal verb	*darf*
punctuation	*!*

Table: Sample Marker word for each category

- entries are included in their inflected forms

Word and chunk alignment in *Marclator*

- word alignment: *Giza++* (Och and Ney, 2003)
- chunk alignment: edit-distance-style algorithm in which distances are replaced by opposite-log conditional probabilities (Tinsley et al., 2008)

Recombination in *Marclator*

- left-to-right monotone recombinator
- translating:
  1. matching sentence
  2. if none is found: sentence is chunked
  3. each chunk that is not found in the example base is then split into single words

  if several TL correspondences for an SL chunk or word are found in the example base, the one with the highest probability is chosen

output: single hypothesis for each input sentence

Generalized Templates in *Marclator*

- problem inherent in the approach described is that the chunks of an input sentence often cannot be found in the example base
- goal: increase the chunk coverage of a system
- Gough and Way (2003) extended a precursor to *Marclator* by including an additional layer of abstraction: produced generalized chunks by replacing the Marker word at the beginning of a chunk with the name of its category

Example

of a marathon → *<PREP> a marathon*

EBMT at Carnegie-Mellon University: *CMU-EBMT*

- *CMU-EBMT*: part of *PanLite* (Frederking and Brown, 1996), an MT architecture developed at Carnegie-Mellon University (CMU) `http://sourceforge.net/projects/cmu-ebmt/`
- modules: matching, alignment, recombination

Matching and Alignment in *CMU-EBMT* I

- every substring of the input sentence with a minimum length of two tokens that appears in the SL half of the example base is extracted
- for each of these fragments, system identifies the smallest and the largest possible segment in the TL sentence that correspond to it (on the basis of bilingual dictionary and, optionally, a TL synonym list)
- every possible substring of the largest segment that contains at least the minimal segment receives a score
- alignment score is the weighted sum of the values of eight features, which include:
  - number of SL words with no correspondences in the TL segment
  - number of TL words with no correspondences in the SL fragment
  - number of SL words with a correspondence in the TL sentence but not in the relevant TL segment
  - difference in length between the SL and the TL segment
- translations are passed on to the recombination step as long as their scores do not exceed five times the length of the SL fragment

Generalized Templates in *CMU-EBMT* I

- Brown (1999): generate semantic and syntactic generalized templates (equivalence classes)
- class members can in turn contain classes

Class	Sample member
<religion>	*Christianity – Christentum*
<month>	*December – Dezember*
<fullname-m>	*<firstname-m> <lastname> – <firstname-m> <lastname>*
<fullname-m>	*George Washington – George Washington*
<adj-s>	*affordable – accesible*
<noun-m-p>	*painters – pintores*
<np-m>	*<poss> <noun-m> – <poss> <noun-m>*
<np-f>	*the <noun-f> – la <noun-f>*
<np-f>	**a <color> <noun-f> – une <noun-f> <color>**

Table: Semantic and syntactic equivalence classes

Generalized Templates in *CMU-EBMT* II

- system generalizes both the training and the test set: recursively replaces words and phrases that are part of an equivalence class with the corresponding class tag
- syntactic classes are applied before semantic classes
- training data: generalization is performed only if a member of a particular equivalence class is found in both the SL and the corresponding TL sentence

Generalized Templates in *CMU-EBMT* III

- test set: all members of an equivalence class are replaced recursively
- matching process is equivalent to that of the purely lexical *CMU-EBMT* system, with the apparent difference that here, two matching levels – a lexical and a generalized one – exist
- alignment: proceeds in the same way as in *CMU-EBMT*
- following this, the rules that were stored during the generalization of the input sentence are applied in reverse so as to transform the generalized TL fragments into word form TL fragments

Our Approach I

- combining the generalized template extensions of *Marclator* and *CMU-EBMT* → build new system that applies both the DCU and the CMU generalization scheme
- goal: see whether our combined system could outperform the two individual systems → for this, we ran an experiment with the combined system as well as one with each individual system
- we (re-)implemented the three approaches on top of *Marclator*
- three systems:
  1. *Marclator* with DCU generalized templates (System 1)
  2. *Marclator* with CMU generalized templates (System 2)
  3. *Marclator* with DCU & CMU generalized templates (System 3)

Our Approach II

- experimental data set: English–German subtitles provided by a commercial subtitling company
- corpus contained 1,133,063 subtitles which consisted of on average 8.9 tokens for English and 7.9 for German
- training set: 1,130,717 subtitles
- test set, development set: 1173 subtitles each

System 1: DCU Generalized Templates I

- includes the generalized template extension to *Marclator*
- if a chunk is not found in example base: replace the Marker word at the beginning of a chunk by its corresponding Marker tag and search for the resulting generalized chunk in the example base (if this attempt fails, the system reverts to word-by-word translation)

System 1: DCU Generalized Templates II

Example

an SL chunk *i 've finally got* cannot be found in the example base
→ System 1 generalizes it to <*PERS_PRON*> *'ve finally got*
→ extracts the corresponding TL generalized chunk, e.g., <*PERS_PRON*> *haben*
→ searches for a German translation for the SL Marker word *i* (underlying the SL Marker tag <*PERS_PRON*>) in the word alignments
→ finds, e.g., *ich*
→ produces the TL chunk *ich haben*
translation is deficient → discussion of problems inherent in the approach coming up

System 2: CMU Generalized Templates I

- 81 classes for language pair English–German provided to us by the developer of the *CMU-EBMT* generalized extension, majority are semantic classes
- classes contain a total of 5545 replacement rules (equivalence class tag and an SL–TL pair whose two halves may be replaced by the tag)

System 3: DCU & CMU Generalized Templates

- combines Systems 1 and 2 → generalizes over DCU Marker words as well as CMU semantic and syntactic equivalence classes
- DCU and the CMU generalization schemes are not mutually exclusive → overlaps, i. e., the CMU classes contain 50 words that are also Marker words for English (e. g., *after*, *and*, *before*), and 19 for German (e. g., *aber*, *allen*, *er*) → we prompted the system to generalize over the Marker words first, giving preference to the DCU scheme in case of overlaps

Baseline Systems

three baselines:

1. *Marclator* (non-generalized)
2. *OpenMaTrEx* (Dandapat et al., 2010): uses EBMT chunk pairs from *Marclator* and SMT phrase pairs from *Moses*
   default configuration (5-gram language model with modified Kneser-Ney smoothing, tuning via MERT, optional binary feature that records whether a phrase pair is an EBMT chunk pair or not)
3. *Moses* (Koehn et al., 2007): default system included in *OpenMaTrEx* (5-gram language model, modified Kneser-Ney smoothing, tuning via MERT (Och, 2003), lexicalized reordering model)

language models for *Moses* and *OpenMaTrEx*: TL side of training data

Results I

System	BLEU	NIST	METEOR
1	0.1274	**4.3948**	**0.4052**
2	0.1269	4.3815	0.4047
3	**0.1277**	4.3937	0.4051
Marclator	0.0995	4.2411	0.3990
OpenMaTrEx	0.2763	5.7880	0.4914
Moses	0.2709	5.7472	0.4854

Table: Evaluation scores

- no agreement among all three metrics as to which system performed best: System 3 performed best according to BLEU, while System 1 performed best according to NIST and METEOR

Discussion I

- our generalized EBMT systems achieved higher scores than the lexical EBMT system *Marclator* → supports earlier findings according to which EBMT systems benefit from generalized templates
- investigated the generalized chunk coverage of Systems 1 and 2 (number of successful generalized chunk matches with respect to the total number of attempts made at matching a generalized chunk): 8.26 % for System 1, and 2.14 % for System 2 (very low) → the higher generalized chunk coverage of System 1 was the reason why this system performed better than System 2

Discussion II

- low generalized chunk coverage of System 2: demonstrates the problem inherent in the use of semantic word classes, which form the majority of the CMU equivalence classes → classes are very specific, many of them (e. g., *city, company, country*) have proper name members → on average, each class contains 69 members → to improve the generalized chunk coverage, number would have to be increased
- combining Systems 1 and 2 into System 3 did not yield a clear improvement over the individual performances of these two systems → due to overlaps in the generalization schemes
- System 1: one major source of errors: chunk-internal boundary friction
  boundary friction is normally caused by combining two separate translation units that do not agree in grammatical case
  with the introduction of Marker-based templates, it can also take place *within* a single chunk, i. e., when a Marker word is inserted that does not agree with the grammatical properties of the rest of the chunk

Discussion III

in the case of translating from English to German, inserting TL Marker words context-insensitively (as is done in System 1) is error-prone: due to the morphological richness of German, an English Marker word can correspond to multiple word forms of the same lemma on the German side
e.g., English Marker word *are* can correspond to German Marker words *bist*, *sind* and *seid*

Example

are you sure . . . – **sind du** *sicher . . .*

chunk-internal boundary friction: combination of *sind* and *du* is grammatically incorrect

Discussion IV

- our EBMT systems performed much worse than the baseline systems *Moses* and *OpenMaTrEx* → performance gap is largely due to the recombination module of *Marclator* (monotone recombinator, outputs only the one-best hypothesis, no language model is applied) → both *OpenMaTrEx* and *Moses* apply a language model → it is essential for an EBMT system to make use of a language model for hypothesis recombination

Conclusion I

- experimented with combining two existing EBMT systems that rely on generalized templates → combined system did not yield a significant improvement in translation quality compared to the individual performances of the two systems
- however, the generalized EBMT systems consistently outperformed the lexical EBMT baseline → shows that generalized templates are beneficial to an EBMT system's performance
- more difficult to achieve a high generalized chunk coverage with semantic generalized templates than with generalized templates based on function words

  semantic generalized templates have the advantage that they do not interfere with the grammar of a sentence

Conclusion II

- generalized templates based on function words are relatively easy to compile

  system which relies on such templates can suffer from chunk-internal boundary friction

  to reduce the chunk-internal boundary friction problem, we plan to develop an algorithm that context-sensitively instantiates TL Marker tags by using a language model → incorporate it into our generalized template extension of *Marclator*

Bibliography I

Stephen Armstrong, Declan Groves, Marian Flanagan, Yvette Graham, Bart Mellebeek, Sara Morrissey, Nicolas Stroppa, and Andy Way. The MaTreX System: Machine Translation Using Examples. In *TC-STAR OpenLab Workshop on Speech Translation*, page pages not numbered, Trento, Italy, 2006.

Ralf D. Brown. Example-Based Machine Translation in the Pangloss System. In *COLING-96: The 16th International Conference on Computational Linguistics, Proceedings*, pages 169–174, Center for Sprogteknologi, Copenhagen, Denmark, 1996.

Ralf D. Brown. Adding Linguistic Knowledge to a Lexical Example-based Translation System. In *8th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI 99)*, pages 22–32, University College, Chester, England, 1999.

Ilyas Cicekli and Halil Altay Güvenir. Learning Translation Templates from Bilingual Translation Examples. *Applied Intelligence*, 15(1):57–76, 2001.

Sandipan Dandapat, Mikel L. Forcada, Declan Groves, Sergio Penkale, John Tinsley, and Andy Way. OpenMaTrEx: A Free/Open-Source Marker-Driven Example-Based Machine Translation System. In *Proceedings of IceTAL*, pages 121–126, Reykjavík, Iceland, 2010.

Robert E. Frederking and Ralf D. Brown. The Pangloss-Lite Machine Translation System. In *Expanding MT Horizons, Proceedings of the Second Conference of the Association for Machine Translation in the Americas*, pages 268–272, Montreal, Quebec, Canada, 1996.

Bibliography II

Nano Gough and Andy Way. Controlled Generation in Example-Based Machine Translation. In *MT SUMMIT IX: Proceedings of the Ninth Machine Translation Summit*, pages 133–140, New Orleans, USA, 2003.

Thomas Green. The Necessity of Syntax Markers. Two Experiments with Artificial Languages. *Journal of Verbal Learning and Behavior*, 18:481–496, 1979.

Declan Groves and Andy Way. Hybrid Example-Based SMT: the Best of Both Worlds? In *ACL-05: Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond, Proceedings of the Workshop*, pages 183–190, University of Michigan, Ann Arbor, Michigan, USA, 2005.

Halil Altay Güvenir and Aysegül Tunc. Corpus-Based Learning of Generalized Parse Tree Rules for Translation. In *Proceedings of the 11th Biennial Conference of the Canadian Society for Computational Studies of Intelligence on Advances in Artificial Intelligence (AI'96)*, pages 121–132, London, UK, 1996.

Hiroyuki Kaji, Yuuko Kida, and Yasutsugu Morimoto. Learning Translation Templates from Bilingual Text. In *Proceedings of the fifteenth [sic] International Conference on Computational Linguistics, Actes du quinziéme colloque international en linguistique informatique, COLING-92*, pages 672–678, Nantes, 1992.

Mihoko Kitamura and Yuji Matsumoto. A Machine Translation System Based on Translation Rules Acquired from Parallel Corpora. In *International Conference: Recent Advances in Natural Language Processing, Proceedings*, pages 27–36, Tzigov Chark, Bulgaria, 1995.

Bibliography III

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 177–180, Prague, Czech Republic, 2007.

Franz Josef Och. Minimum Error Rate Training in Statistical Machine Translation. In *41st Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pages 160–167, Sapporo Convention Center, Japan, 2003.

Franz Josef Och and Hermann Ney. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51, 2003.

Nicolas Stroppa and Andy Way. MaTrEx: DCU Machine Translation System for IWSLT 2006. In *IWSLT 2006: Proceedings of the 3rd International Workshop on Spoken Language Translation*, pages 31–36, Palulu Plaza, Kyoto, Japan, 2006.

John Tinsley, Yanjun Ma, Sylwia Ozdowska, and Andy Way. MATREX: the DCU MT System for WMT 2008. In *ACL-08: HLT: Third Workshop on Statistical Machine Translation, Proceedings of the Workshop*, pages 171–174, The Ohio State University Columbus, Ohio, USA, 2008.

Francis M. Tyers, Felipe Sánchez-Martínez, Sergio Ortiz-Rojas, and Mikel L. Forcada. Free/open-source resources in the Apertium platform for machine translation research and development. *The Prague Bulletin of Mathematical Linguistics*, 93: 67–76, 2010.

Bibliography IV

Andy Way. Translating with Examples. In *MT Summit VII: Workshop on Example-Based Machine Translation, Proceedings of the Workshop*, pages 66–80, Santiago de Compostela, Spain, 2001.