

TMX Markup: A Challenge When Adapting SMT to the Localisation Environment

Jinhua Du¹, Johann Roturier² and Andy Way¹

1. CNGL, School of Computing, Dublin City University, Dublin, Ireland
2. Symantec, Ballycoolin Business Park, Blanchardstown, Dublin 15, Ireland
jdu@computing.dcu.ie, johann_roturier@symantec.com
away@computing.dcu.ie

Abstract

Translation memory (TM) plays an important role in localisation workflows and is used as an efficient and fundamental tool to carry out translation. In recent years, statistical machine translation (SMT) techniques have been rapidly developed, and the translation quality and speed have been significantly improved as well. However, when applying SMT technique to facilitate post-editing in the localisation industry, we need to adapt SMT to the TM data which is formatted with special mark-up. In this paper, we explore some issues when adapting SMT to Symantec formatted TM data. Three different methods are proposed to handle the Translation Memory eXchange (TMX) markup and a comparative study is carried out between them. Furthermore, we also compare the TMX-based SMT systems with a customised SYSTRAN system through human evaluation and automatic evaluation metrics. The experimental results conducted on the French and English language pair show that the SMT can perform well using TMX as input format either during training or at runtime.

1 Introduction

Translation memory (TM) plays an important role in the localisation industry (Planas and Furuse, 2000; Garcia, 2005). TM is an effective way to enable the translation of segments (sentences, paragraphs, or phrases) of documents by searching for similar segments in a database and retrieving the suggested matches with a fuzzy match score.

In the localisation process, TM systems are a fundamental tool in automatic translation workflows. We argue that there are three main reasons: 1) a TM is easy to build; 2) it complies with industry standards and so is efficient to store, share and re-use; 3) localisation is generally limited to a specific domain using a TM provides a fast and relatively good translation for specific domains. Even though SMT has been significantly developed and the translation quality has been highly improved in recent years from the academic viewpoint (Brown et al., 1993; Yamada and Knight, 2001; Koehn et al., 2003; Och and Ney, 2003; Chiang, 2005; Koehn et al., 2007), a lot of work must be done to produce high-quality translations when it is applied into industry application. As a result, SMT has a long way to go to establish itself as an acceptable component in computer-assisted translation (CAT) environments such as a commercial post-editing environment, at least when compared to TM systems or rule-based MT systems.

Much research progress has been achieved by combining TM with rule-based or SMT techniques. Roturier (2009) used a hybrid approach combining a customised SYSTRAN system with TM to refine the MT output in Symantec's localisation workflow. DeCamp (2009) and Simard and Isabelle (2009) proposed a way of integrating a phrase-based SMT (PB-SMT) system with TM within a CAT environment. They designed a PB-SMT system which behaves more like the TM component in CAT systems. However, they did not discuss TMX markup issues. In a real world environment, industrial TM data contains TMX tags when exported into this format. During the translation process, especially in a post-editing environment, translations with corresponding markup must be presented to the user or translator for their

reference. Therefore, when we adapt SMT systems to TMX data, the first problem concerns the handling of the markup.

In this paper, we propose three methods to handle the TMX markup when adapting SMT techniques to TMX files. Based on these approaches, we perform a series of comparative experiments on French and English TMX data provided by Symantec. This work is the first step to adapt SMT techniques to a TMX-based localisation environment.

The remainder of this paper is organised as follows. In section 2, we briefly discuss some concepts related to TM and TMX. Section 3 describes our three proposed tag-handling approaches in detail. The experiments conducted on Symantec French and English TMX data are reported in Sections 4 and 5. In addition, we also compare the results from human evaluation on the English-to-French translation direction. In Section 6, we perform an in-depth investigation and analyse three open questions based on our experiments. Section 7 concludes and gives avenues for future work.

2 Translation Memory and TMX

2.1 TM

A TM database contains pairs of source language and target language segments of text. These parallel translations may contain internal tags (such as HTML, XML or RTF markup). Given a segment of source language text as input, the TM system searches its database for an exact or fuzzy match, and then return the target sentence with a match score. The two types of matches are:

- Exact searches: Only segments that match the input text are retrieved from the TM database.
- Fuzzy searches: Set up a match threshold (such as 70%) and retrieve the best matched segments from the TM database.

The threshold in fuzzy searches is called “match quality”. The higher the threshold is set, the better the quality of the proposed translations, but the higher the risk of no translation being retrieved. The retrieved translation would be proposed to the user or translator, usually in a post-editing environment. The user can then reuse, edit or discard the suggested match. Generally, the human-corrected translation is fed back to the TM database.

2.2 TMX

TMX is the vendor-neutral open XML standard for the exchange of TM data created by CAT and localization tools.¹ The purpose of TMX is to allow easier exchange of TM data between tools and/or translation vendors with little or no loss of critical data during the process.

We illustrate some TMX related tags used in this paper as follows:

- `<ph>...</ph>`: placeholder which is used to delimit a sequence of native standalone codes in the segment.
- `<ut>...</ut>`: unknown tag which is used to delimit a sequence of native unknown codes in the segment.

3 TMX Markup Processing When Adapting SMT to TM Data

3.1 Motivation

In the past ten years, SMT technology has continued to improve to the extent that it now outperforms rule-based MT systems for many language pairs. However, much SMT research work in the academic field mainly focuses on news or spoken language domains, and uses plain text and largely clean data for translation. Facing the huge demand of the translation market, commercialising further SMT technology should benefit the MT community and society as a whole.

When an SMT system is confronted with TMX data, the main challenge is to determine how to handle the markup. Processing the TMX markup is a challenging issue because the markup information should be kept in the translation result in order to provide a meaningful translation to the user.

The intuitive way to handle the markup is to record the positions of the markup in the source sentence and then restore it in the corresponding places in the translated sentence. This method is also used in some TM or rule-based systems.² In this section, we propose three methods to process the TMX markup, and carry out a series of comparative experiments to verify the three approaches.

3.2 Method 1: Markup Transformation

“Markup transformation” represents the intuitive way mentioned in section 3.1, i.e., transforming

¹<http://www.lisa.org/tmx/tmx.htm>.

²<http://pierre.senellart.com/publications/senellart2005sts.html>

the markup from the source side to the target side using substituting and restoring steps. The functionalities of Moses (Koehn et al., 2007) are used in this approach. Moses has the capability to take an XML-tagged sentence as input in order to make use of the external knowledge for decoding. Furthermore, it is able to specify reordering constraints³ to the decoder so that some source phrases or fragments can be translated as a “block” which may keep the coherence inside such phrases. Such a “block” is called a **zone**, and Moses allows the specification of such constraints using XML markup.⁴ We call these two functionalities “**XML Interface**”.

We define the “Method 1” of using Moses XML markup and reordering constraints functionalities as “markup transformation”. However, this is not a simple way to solve the TMX markup issue. Different from the XML markup usage and reordering constraints usage in Moses, the biggest challenge of handling TMX markup is to recognise and specify all the TMX markup boundaries in the input sentences and then restore the markup in the corresponding places in the translated sentences. In order to solve these problems, we combine the functionality of “-report-segmentation (-t)”⁵ of Moses, with the XML interface functionalities to provide a solution for markup restoration. Figure 1 shows the workflow of this method.

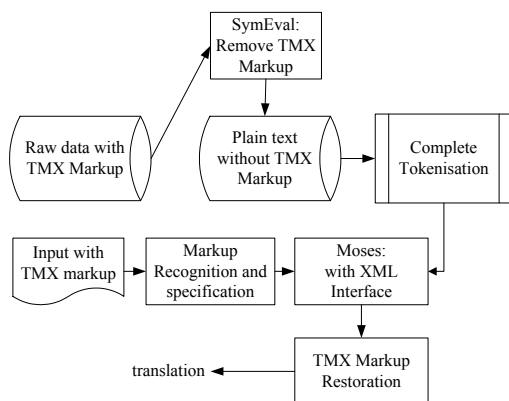


Figure 1: Using Moses XML Markup and Reordering Constraints Functions

In Figure 1, the TMX markup in the training data has been removed automatically using the Symantec tool “SymEval” which parses a TMX

³<http://www.statmt.org/moses/manual/manual.pdf>.

⁴<zone> is a keyword in Moses to identify the reordering constraints.

⁵This option reports phrase segmentation in the output.

file and extracts the text contained in the source and target segments. The plain text data is then used to build a regular SMT system. The key issue in this method is how the TMX markup is processed in the input document. We present three steps to handle it:

- recognise the blocks or boundaries of the TMX markup as well as the content of the input sentence, then replace the markup blocks with the reordering constraint tag “<zone>”, which can keep the content to be translated as a whole to guarantee that the markup is placed in the correct corresponding positions in the translated sentence;
- configure the decoding of Moses with options “-xml-input”⁶ and “-t” to activate the XML interface functionality and the source phrase segmentation functionality;
- given the relationship (generated by option “-t”) between the translated phrase and the source phrase segmentation, the TMX markup is restored in the corresponding positions in the translated text.

See Figure 2 (b) as an illustration. In Figure 2 (a), the pair of sentences are the original data with TMX markup without any tokenisation. In Figure 2 (b), the boundary of the markup “<ph> ... </ph>” as well as the content “Requirements for using IDR” were recognised and then tagged with “<zone>”. </zone> indicates the end of the boundary.

“Method 1” is very complicated and has the highest risk of quality loss in the three approaches. We will explore this issue in Section 6.2.

3.3 Method 2: Complete Tokenisation

What would happen if we threw all the TMX markup and the content together into a SMT system? This is an interesting question because intuitively we might be concerned that the markup might be some kind of noise which may accordingly have a negative impact on the translation quality. Thus, we propose “Method 2” to tackle this issue. This approach keeps all the TMX tags in the training data and the test data, and

⁶This option allows markup of input with desired translations and probabilities. Values can be ‘pass-through’ (default), ‘inclusive’, ‘exclusive’, ‘ignore’.

English: `<ph><:imk 3></ph><ph><:crmk 2></ph>`Requirements for using IDR

French: `<ph><:imk 3></ph><ph><:crmk 2></ph>`Configuration requise pour l'utilisation de l'option IDR
 (a) original sentence pair

English: `<zone>` Requirements for using IDR `</zone>`

French: `<zone>` Configuration requise pour l ' utilisation de l ' option IDR `</zone>`
 (b) Method 1: markup transformation

English: `< ph > & lt ; : imk 3 & gt ; < / ph > < ph > & lt ; : crmk 2 & gt ; < / ph >` Requirements for using IDR

French: `< ph > & lt ; : imk 3 & gt ; < / ph > < ph > & lt ; : crmk 2 & gt ; < / ph >` Configuration requise pour l &apos ; utilisation de l &apos ; option IDR
 (c) Method 2: complete tokenisation

English: `<ph> < :imk 3 > </ph> <ph> < :crmk 2 > </ph>` Requirements for using IDR

French: `<ph> < :imk 3 > </ph> <ph> < :crmk 2 > </ph>` Configuration requise pour l &apos ; utilisation de l &apos ; option IDR `</seg>`
 (d) Method 3: partial tokenisation

Figure 2: Comparison of three TMX mark-up processing methods

tokenises them just like ‘normal’ data. We define this method as “Complete Tokenisation”. Figure 3 demonstrates the workflow of Method 2 and Method 3. The dashed lines indicate that the SMT system is fed by different markup processing methods independently.

In Method 2, all the words, symbols and punctuations in the TMX markup are treated as normal tokens. In Figure 2 (c), we can see that each symbol or punctuation mark is separated as a single token. By examining the tokenised data, we found there are three potential problems for this method which we define as “over-tokenisation” problems:

- this method causes the length of a sentence to be significantly increased;
- the long sentences increase the risk of poor word alignments;
- the long sentences increase the risk of poor word reordering;

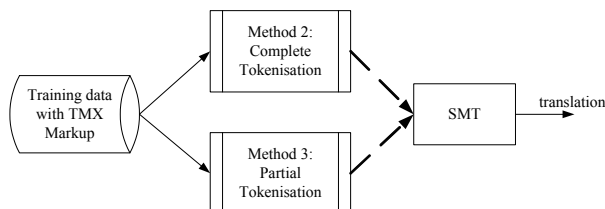


Figure 3: Diagrams of Method 2 and Method 3

Considering the problems of Method 2, we present a new tokenisation method which we define as “Partial Tokenisation” in section 3.4.

3.4 Method 3: Partial Tokenisation

In this method, the basic idea is also to keep all the TMX tags in the training data and test data. We first classify the TMX tags into different categories, and then keep the categories as a whole word or token not to be split. See Figure 2 (d) as an illustration. For the sake of not significantly increasing the length of a sentence, the “<ph>, </ph>” etc. tags can be kept as a whole symbol during tokenisation. This keeps the length of a sentence to a reasonable size as shown in Figure 2 (d). In Figure 2, we can see that the lengths of the example using Method 2 are 36 and 47 in the English and French sides respectively, while they are respectively 16 and 23 using Method 3 which are approximately 50% shorter.

Examining this method, we also find that there may be some potential problems such as the generalisation issue, which indicates the capability to generalise other TMX tag classifications that never occur in the training data (normally we call this “unknown words”). In section 6.1, we will further discuss this problem.

4 Experimental Settings

In our experiments, all the data including training data and test data come from Symantec TM databases that have been exported to TMX files. The language pair is English and French. We performed the experiments on both directions.

4.1 Data and SMT System

The whole training data includes 108,967 sentence pairs. We extracted 2,500 pairs as our development set (devset) and the rest of the data is used

as training data. The test data comes from another document which contains 4,977 pairs. All the data are tagged with TMX 1.4b format. In order to reliably verify the capability of SMT methodology, we should avoid a serious “overlap” problem between the training data, devset and test data. In our experiments, there is no overlap between the training data and devset, the training data and test data are only 4% overlapped which is acceptable for such a big test set.

We employ our MaTrEx system which provides a wrapper around the Moses decoder to carry out the experiments (Du et al., 2009). The Moses decoder represents the-state-of-the-art phrase-based SMT engine. Based on the three markup-handling methods, we built three translation models for each translation direction. The language models are trained with 5-grams.

4.2 Evaluation

In order to give an overall review of SMT performance using TMX data, we use the mainstream automatic evaluation metrics such as BLEU (Papineni et al., 2002), NIST (Doddington, 2002), METEOR (Banerjee and Lavie, 2005), TER (Snover et al., 2006) as well as some human evaluation scores.

5 Experimental Results

In this section, we report the experimental results on French-to-English and English-to-French translation directions respectively in terms of four automatic evaluation metrics. All the scores are case-sensitive and evaluated on TMX-formatted texts.

5.1 French-to-English Translation

Table 1 shows the results of the three TMX markup processing approaches. We can see that the “Method 3” method achieved the best performance in terms of the four metrics. However, it is just slightly better than “Method 2”. It is surprising that there is a huge drop of 13.34%, 1.44%, 5.07%, 10.98% absolute points for “Method 1” in terms of BLEU, NIST, MTR and TER compared to the “Method 3”. We will give a detailed discussion on these results in section 6.

5.2 English-to-French Translation

Table 2 shows the comparable scores of the three methods on English-to-French translation direction. The results are different from those in Table 1. The “Method 3” only performed best in

| Method | BLEU | NIST | MTR | TER |
|----------|--------------|-------------|--------------|--------------|
| Method 1 | 45.51 | 7.84 | 71.15 | 50.16 |
| Method 2 | 58.66 | 9.28 | 74.30 | 39.61 |
| Method 3 | 58.85 | 9.28 | 76.22 | 39.18 |

Table 1: Results on French-to-English Translation

terms of MTR score while “Method 2” performed best on BLEU, NIST and TER metrics. “Method 1” still performs worst in this direction. The bottom line in Table 2 is the results from the custom SYSTRAN system. We can see that our “Method 2” system achieved 11.38%, 1.47%, 3.34%, 8.99% absolute points higher on BLEU, NIST, METEOR and TER metrics than the custom SYSTRAN system. From the automatic evaluation viewpoint, this is a significant improvement. It is also noticed that

| Method | BLEU | NIST | MTR | TER |
|----------|--------------|-------------|--------------|--------------|
| Method 1 | 48.05 | 7.77 | 31.39 | 47.45 |
| Method 2 | 60.05 | 9.32 | 33.33 | 40.55 |
| Method 3 | 59.42 | 9.22 | 33.82 | 40.94 |
| SYSTRAN | 48.67 | 7.85 | 29.99 | 49.54 |

Table 2: Results on English-to-French Translation

the MTR scores for French are obviously lower than those of English, which is because MTR is a language dependent metric and it uses the different chunk penalty for English and French.

5.3 Human Evaluation

We also carried out a human evaluation for the English-to-French task to compare the performance of our three methods with the custom SYSTRAN system from the translator viewpoint. The criteria for human evaluation use a unique scale of 4 scores to measure the acceptability of the output on segment level (Roturier, 2009), which are:

- Excellent: the output is syntactically correct and there is no post-editing required;
- Good: only minor post-editing is required in terms of actual changes or time spent post-editing;
- Medium: significant post-editing is required after spending some time trying to understand the intended meaning and where the errors are;
- Poor: it would be better to manually retranslate from scratch (post-editing is not worthwhile).

Based on these criteria, we extracted 100 segments with TMX markup from each system and carried out the human evaluation. The comparative results are shown in Figure 4.

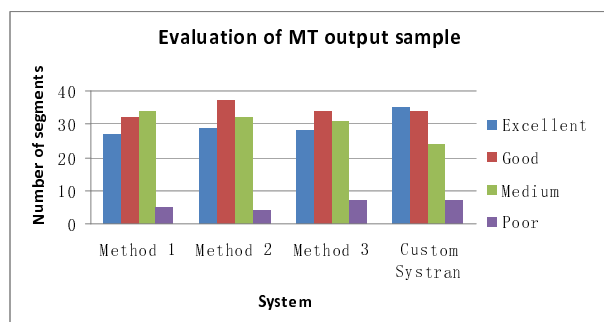


Figure 4: Human evaluation on the proposed three methods and the customised SYSTRAN system

In Figure 4, there are 59, 66, 62 and 69 segments which are at least “Good” respectively from “Method 1”, “Method 2”, “Method 3” and the customised SYSTRAN, and 5, 4, 7 and 7 segments respectively from the four systems are declared to be “Poor”. Based only on these numbers, it is difficult to claim which is the best method. However, the interesting finding here is that the differences between these systems evaluated by humans are small while the distinctions evaluated by automatic metrics are huge, especially in terms of BLEU scores (Callison-Burch et al., 2006). From a user’s perspective, multiple segments could be presented to users in a post-editing environment so that users can select the best segment based on source characteristics.

6 Findings and Analysis

Based on these experimental results, we have some interesting findings as well as many open questions about the results, for example,

- Q1: how much is the data processed by Method 2 different from that of Method 3? What would happen if we used a different test set with a different TMX version format?
- Q2: why is there a huge drop in terms of performance with Method 1 compared to Method 2 and Method 3?
- Q3: what kinds of errors occurred in these three approaches from a translator’s viewpoint?

In order to investigate the three questions, we also select 100 translated sentences with TMX format to statistically calculate the format errors. Three types of format errors are defined:

- Error 1: the block of markup unit is well-formed,⁷ but it is in wrong position;
- Error 2: the block of markup unit is badly-formed;⁸
- Error 3: there are no markup units in source sentence, but there are in the reference sentence. This is an error of the corpus itself rather than a translation error.

6.1 Analysis on Question 1

In order to answer this question, we firstly carried out the statistical characteristic of the data processed by Method 2 and Method 3 respectively, and then we used another test set with TMX 1.1 format to test the generalisation issue. Table 3 shows the characteristics of the two differently tokenised data and Table 4 demonstrates the comparative scores of Method 2 and Method 3 on another test set.

| Method | English | | | French | | |
|----------|---------|-----------|--------------|--------|-----------|--------------|
| | Word | Ave. Len. | %Sen. (>100) | Word | Ave. Len. | %Sen. (>100) |
| Method 2 | 3.43m | 31.45 | 6.72 | 3.78m | 34.71 | 7.5 |
| Method 3 | 2.1m | 19.31 | 0.52 | 2.36m | 21.66 | 0.63 |

Table 3: Characteristics of data processed by Method 2 and Method 3

In Table 3, we can see that the word counts in “Method 2” are significantly increased because of the fine-grained tokenisation. Consequently, the ratio of long sentences that are more than 100 words in “Method 2” are more 10-fold than that of “Method 3”, which will be abandoned by GIZA++ (Och and Ney, 2003) default configuration. Otherwise, it will make the word alignment more difficult and complicated. Therefore, from a data complexity viewpoint, “Method 3” is more efficient and effective.

In Table 4, we can see that “Method 2” generally beats “Method 3” in most automatic evaluation metrics in both directions. We argue that it

⁷“well-formed” indicates the markup is paired, i.e., it is closed.

⁸“badly-formed” indicates the markup unit is opened without the closed part or the markup unit is lost in translated text.

| Metric | French-to-English | | English-to-French | |
|--------|-------------------|--------------|-------------------|--------------|
| | Method 2 | Method 3 | Method 2 | Method 3 |
| BLEU | 65.70 | 60.11 | 64.66 | 62.25 |
| NIST | 8.85 | 8.63 | 8.87 | 8.75 |
| MTR | 76.35 | 76.40 | 33.43 | 30.80 |
| TER | 38.13 | 36.62 | 39.86 | 39.20 |

Table 4: Results on TMX 1.1 format test data

is possibly because of the “generalisation” problem which is caused by the supervised learning method. The segments with TMX 1.1 format is mainly with tag `<ut>` which never happens in TMX 1.4b format data. Since “Method 3” takes the tag as a whole word, it may regard `<ut>` as an “unknown” word. Consequently, during the decoding process, it would cause some serious re-ordering problems. “Method 2” will split `<ut>` into three tokens of “`< ut >`” which decreases the number of unknown words. Therefore, we can say that although the “Method 2” has a serious problem of long sentences, it has a better generalisation capability.

6.2 Analysis on Question 2

As we mentioned in section 3.2, the potential problems of “Method 1” are the “mismatch” and “limited reordering”. “Mismatch” indicates that 1) in some cases, we cannot exactly record the markup positions on the source side because of some reasons such as complicated, different word order etc. 2) the functionality of specifying reordering constraints limits the word/phrase reordering to happen either inside each block or between blocks which in some sense becomes partially “monotone decoding” so as to have an significant influence on the word order on the target side. Examining the 100 formatted sentences, we found that there is a lot of Error type 2 in the results of Method 1, in particular that some markup units were missed. We need to improve the accuracy of recognising the corresponding relationship of markup pairs between the source side and the target side.

6.3 Analysis on Question 3

Based on the human evaluation data, we investigated and analysed the types of errors that occurred in our three proposed methods. Taking the selected 100 sentences as a set of samples, there are 17% errors of Type 3. In Method 1, there are about 11% errors of Type 1, 20% errors of Type 2; in Method 2, there are about 9% errors of Type 1 and 0% errors of Type 2; in Method 3, the Error 1 is 10%

and Error 2 is 0%. From the statistics of the error types, we can see 1) the format errors are an important factor in lowering the translation performance; 2) Method 2 and Method 3 did not mess up the the markup units.

Here we list some examples of specific types of errors from a human evaluation viewpoint.

Error type 1:

REF: uncheck the `<ph> < guimenuitem moreinfo="none" > </ph>` include weekends `<it pos="end"> </guimenuitem > </it>` **check box**.

Method 3: uncheck the **check box** `<ph> < guimenuitem moreinfo="none" > </ph>` include weekends `<it pos="end"> </guimenuitem > </it>`.

In this type of error, the markup units are well-formed, but in the translated text using Method 3, the markup is incorrectly placed behind the phrase “check box”.

Error type 2:

REF: click `<ph> < guimenuitem moreinfo="none" > </ph>` all media servers `<it pos="end"> </guimenuitem > </it>`.

Method 1: click `<ph> < guimenuitem moreinfo="none" > </ph>` all media servers.

It can be seen that the markup in bold in REF was missed in translated text of Method 1. This is probably because of the mismatch of the “`</zone>`” boundary.

Error type 3:

REF: `<bpt i="1" type="font">\f2 </bpt>`in addition, enable debug logs also creates debug log files that are stored on a media server `<ept i="1"></ept><bpt i="2" type="font">\f2 </bpt>`s hard drive.`<ept i="2"></ept>`
SOURCE: en outre, des fichiers journaux de dbogage sont cres et stocks sur un dis dur du serveur de supports.

Method 2: in addition, in the debug log files are cres and stored on a hard drive of the media server.

We can see that this error is impossible for any MT system to automatically correct. Therefore, it needs to be filtered as a kind of noise in the pre-processing stage. From the analysis above, we can conclude that most of the errors occurring in our proposed methods are related to the tag processing. Therefore, in future work, we have to find a more effective and efficient way to facilitate the TMX-based SMT system.

7 Conclusions and Future Work

In this paper, we proposed three ways to handle the TMX markup when adapting SMT techniques

into TMX files, namely “Markup Transformation”, “Complete Tokenisation” and “Partial Tokenisation”. Based on these approaches, we performed a series of comparative experiments on the French–English language pair marked up with TMX data provided. Comparisons are carried out between the proposed methods and the customised SYSTRAN system from the aspect of both human and automatic evaluation. The evaluation results show that the SMT systems can handle the TMX markup well and produce good translations, which is encouraging. Finally we give some detailed error analysis on these three methods.

As for future work, firstly we need to refine these methods and further investigate the problems occurred in experiments. Secondly, we need to employ these approaches on more different language pairs such as Chinese and English to verify their consistency and effectiveness.

Acknowledgment

This work is supported by Science Foundation Ireland (Grant No. 07/CE/I1 142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Dublin City University. Thanks also to the reviewers for their insightful comments.

References

- Satanjeev Banerjee and Alon Lavie. 2005. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, ACL-2005*, pages 65–72.
- Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics* 19 (2), pages 263–311.
- Chris Callison-Burch, Miles Osborne and Philipp Koehn. 2006. Re-evaluating the role of BLEU in machine translation research. In *Proceedings of EACL'06*, Trento, Italy, pages 249–256.
- David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL'05*, Ann Arbor, MI, pages 263–270.
- Jennifer DeCamp. 2009. What is missing in user-centric MT? In *Proceedings of MT Summit XII*, Ottawa, Canada, pages 489–495.
- George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram cooccurrence statistics. In *Proceedings of the Second international conference on Human Language Technology Research*, pages 138–145.
- Jinhua Du, Yifan He, Sergio Penkale and Andy Way. 2009. MaTrEx: The DCU MT System for WMT2009. In *Proceedings of the Third Workshop on Statistical Machine Translation, EACL 2009*, pages 95–99.
2005. Long term memories: Trados and TM turn 20. *Journal of Specialised Translation* 4, pages 18–31.
- Philipp Koehn, Franz Josef Och and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT-NAACL'03*, Edmonton, Canada, pages 48–54.
- 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, Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of ACL 2007 of demo and poster sessions*, Prague, Czech Republic, pages 177–180.
- Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of ACL'02*, Philadelphia, United States, pages 295–302.
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics* 29 (1) pages 19–51.
- Kishore Papineni, Salim Roukos, Todd Ward and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of ACL-02*, pages 311–318.
- Emmanuel Planas and Osamu Furuse. 2000. Multi-level similar segment matching algorithm for translation memories and example-based machine translation. In *Proceedings of Coling 2000*, pages 621–627.
- Johann Roturier. 2009. Deploying novel MT technology to raise the bar for quality: a review of key advantages and challenges. In *Proceedings of MT Summit XII*, Ottawa, Canada, pages 1–8.
- Michel Simard and Pierre Isabelle. 2009. Phrase-based machine translation in a computer-assisted translation environment. In *Proceedings of MT Summit XII*, Ottawa, Canada, pages 120–127.
- Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pages 223–231.
- Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of ACL-EACL'01*, Toulouse, France, pages 523–530.