Enhancing a Production TM-MT Environment Using a Quotation TM

Hitokazu Matsushita hitokazu.matsushita@ldschurch.org
Steve Richardson stephen.richardson@ldschurch.org
The Church of Jesus Christ of Latter-day Saints
50 East North Temple Street, Salt Lake City, UT, USA

Abstract

In a typical TM-MT environment, translations for segments are provided from TMs based on matching criteria while the remaining segments are translated completely by MT. However, this binary approach does not always produce desirable translations. For example, even though a contiguous portion of a sentence to be translated may exactly match a TM entry or a frequently occurring sub-segment in many TM entries, if the match for the entire sentence does not exceed some arbitrary threshold, the smaller matches will not be used, and the entire sentence will be machine translated, resulting in a less than perfect translation, even for those portions that could have matched perfectly. In this report, we describe our approach to flexibly combine the capability of MT and TMs, applying exact TM matches to sub-segments of sentences and allowing MT to handle the remaining portions of the sentences. We specifically focus on the scenario where the matched phrases, clauses, and/or sentences are quotations in the text to be translated.

1 Introduction

In recent years, individual translators, language service providers, and large enterprises have more actively utilized hybrid systems of translation memories (TMs) and machine translation (MT) to increase translation productivity (Reinke, 2013). In a typical TM-MT environment employed by those translation professionals, translations for segments are provided from TMs based on matching criteria (e.g., high fuzzy-match scores), while the remaining segments are translated completely by MT. However, this binary approach does not always produce desirable translations. For example, even though a contiguous portion of a sentence to be translated may exactly match a TM entry or a frequently occurring sub-segment in many TM entries, if the match for the entire sentence does not exceed some arbitrary threshold (typically around 70%), the smaller matches will not be used, and the entire sentence will be machine translated, resulting in a less than perfect translation, even for those portions that could have matched perfectly. This is especially unfortunate if potentially matching sub-segments consist of frequently quoted or frequently occurring text for which only the exact human translations are acceptable in a production environment.

In this report, we describe our approach to flexibly combine the capability of MT and TMs, applying exact TM matches to sub-segments of sentences and allowing MT to handle the remaining portions of the sentences. We specifically focus on the scenario where the matched phrases, clauses, and/or sentences are quotations in the text to be translated.

2 Background

The Church of Jesus Christ of Latter-day Saints (henceforth, the Church) translates a wide variety of English materials into more than 100 languages to support communication among more than 15 million members around the world (Richardson, 2012). To facilitate its translation processes, the Church provides a TM-MT hybrid system using SDL WorldServer[1] and the Microsoft Translator Hub[2] for their human translators in the various locations of the world. The hybrid system functions based on the binary approach mentioned in the Introduction above, using a 75% fuzzy match threshold to determine whether the TM matches or MT outputs are used as translation candidates.

As a religious organization, the Church's scriptural canon consists of four volumes: the Holy Bible, the Book of Mormon, the Doctrine and Covenants, the Pearl of Great Price.[3] These scriptures are translated in many languages under the strict supervision of Church authorities.[4] An important aspect of the Church's translation effort is that many of the documents to be translated contain verses or phrases quoted from these scriptures. When scripture text is quoted in Church publications, the corresponding phrases or clauses in the current editions of the same scriptures must be strictly used when the publications are translated. In general, scripture quotes appear in Church publications in the following three forms:

1. All, or almost all, of a verse as a single segment

   e.g., Peace be unto thy soul; thine adversity and thine afflictions shall be but a small moment;[5]

2. All, or almost all, of a verse as part of a segment

   e.g., Remember the yearning hope of a father as expressed by John: "I have no greater joy than to hear that my children walk in truth."[6]

3. A smaller part of a verse as part of a segment

   e.g., God has said that His purpose is "to bring to pass the immortality and eternal life of man" (Moses 1:39).

In item 1 above, the corresponding TM entry will be matched with a 100% or high fuzzy match score because the entire verse is stored in the TM. On the other hand, the verse surrounded by the quotation marks in item 2 above cannot be matched by the TM due to the text preceding the quote, which lowers the fuzzy match score to 68%.[7] Furthermore, the quote in item 3 is even more problematic since it is only a part of the entire verse of scripture and it is also only part of the segment. Although the quote in item 3 is part of the most frequently quoted verse in all of the Church's scriptures, the desired TM match cannot be applied due to the low fuzzy match score, and it will be machine-translated along with the rest of the segment.

Our focus in this study is to apply correct quote translations to segments containing quotes like those in items 2 and 3 above. In this study, we investigate a method to collect scripture quotes to form a quotation TM and apply translations for quotes embedded in segments in

[1] http://www.sdl.com/cxc/language/translation-management/worldserver/

[2] https://hub.microsofttranslator.com

[3] http://www.scriptures.lds.org

[4] The Bible is an exceptional case; typically, specific editions translated by authoritative organizations in the various countries or areas are approved for Church use.

[5] Doctrine and Covenants 121:7.

[6] 3 John 1:4.

[7] Calculated by the character-based Levenshtein distance.

a manner similar to TM matches, while the remaining parts of those segments are machine-translated. This approach can be effective in reducing the amount of post-editing by human translators if the quotes recurring in Church documents are correctly identified and the proper translations are applied. While this approach is particularly relevant to our context, we feel that it could be applied in any context where correct human translations of quotes from canonical sources must be included in the publications of organizations attempting to use MT. In the following section, we discuss the previous work related to the focus of this study.

3 Related Work

Many methods have been proposed for combining TM and MT technology. A popular approach is to integrate TM matches directly into the MT decoder. Biçici and Dymetman (2008) extract phrases commonly found in a sentence to be translated along with their fuzzy-matched TM entries and put weights on those phrases in the MT phrase table to favor them during decoding. Dandapat et al. (2012) investigate enhancement of MT outputs using a hybrid example-based and statistical MT system in a approach similar to that of Biçici and Dymetman (2008). Wang et al. (2013) propose a phrase-based translation model which includes TM information as parameters of the model in order to dynamically choose the best phrase matches during decoding. Li et al. (2016) extend this idea and apply it to syntax-based MT systems to address translations of non-contiguous phrases.

Koehn and Senellart (2010a,b) discuss an approach to extract matching portions in a sentence to be translated and a TM entry, and then constrain the MT decoder with an XML frame to translate only the unmatched portions using hierarchical translation models combined with suffix arrays. A similar approach is also reported by Zhechev and Van Genabith (2010). Furthermore, Ma et al. (2011) and He et al. (2011) extend the approach by Koehn and Senellart (2010a) and investigate a method to identify the most promising translation among all the fuzzy-matched TM entries using support vector machines (SVMs) trained on various linguistic features extracted from TM data.

Other studies focus on an MT-system-agnostic approach, where fuzzy-matched TM entries are identified and applied to sentences to be translated before they are sent to MT systems. Espla-Gomis et al. (2011) and Ortega et al. (2014) investigate a method to patch sentences to be translated with elements in fuzzy-matched TM entries to improve the output from a rule-based MT system in a computer-aided translation (CAT) environment. He et al. (2010) discuss automatic quality estimation of statistical MT otuputs, which determines whether the MT outputs are suitable for post-editing based on an SVM approach with features similar to those described in He et al. (2011).

In this study, we employ an MT-system-agnostic approach using XML frames. We apply TM entries to input segments before submitting them to an MT system, marking the entries that match quotes in the segments with XML frames to constrain the MT system to processes only the text outside of the framed portions of the segments, similar to the approach of Koehn and Senellart (2010a). Unlike the previous studies summarized above, however, which consider cases where various elements of TM entries are used to repair parts of sentences to be translated wherever they may be applied, we confine our focus solely to the application of TM entries to quoted text, as we described in Section 2 above. This is because of the nature of scripture quotes, where the approved translations must absolutely be used. In this sense, the limitation we impose is essential in our production environment, where it must be highly likely that we impact the quality of MT output in only a positive way. In the following sections, we describe our method to prepare scripture TM data in the form of a quotation TM to be used in the patching process.

4 Quotation TM Creation

As shown in the examples in Section 2, the quotes used in sentences are often small portions of the original scripture verses, such as syntactic constituent phrases (e.g., NP and VP) and dependent clauses. In such cases, we cannot apply the scripture translation units (TUs) in the TM directly because most of them are aligned at the verse or sentence level. To properly apply sub-sentential quotes to sentences to be translated, the TUs must be re-aligned at a much finer level of granularity. Several previous studies used aligned phrases in the phrase tables generated during the MT training process (e.g., Biçici and Dymetman 2008; Dandapat et al. 2012). However, these phrases are non-syntactic sequences of words and can be very noisy due to errors made in the word alignment process.

To obtain more finely aligned and linguistically well-formed scripture quotes, we process the original scripture TUs with a bilingual segment alignment algorithm proposed by Deng et al. (2007). The reasons we chose this algorithm are:

1. The algorithm performs alignment processes using small syntactic entities generated by segmentation based on non-terminal punctuation marks.

2. The algorithm considers non-monotonic alignment cases, which frequently occur in the alignment process of linguistically divergent language pairs.

With this algorithm, we obtain scripture TUs aligned at a sub-sentential level. We create a quotation TM with these scripture TUs and use that TM to process sentences in the subsequent quote application process. In the following subsections, we describe the alignment algorithm in detail.

4.1 Two-Step Segment Alignment

The traditional method for bilingual segment alignment is based on dynamic programming (DP) with the assumption that the segments can be aligned monotonically (Gale and Church 1993; Moore 2002, *inter alia*). This assumption is reasonably effective if one aligns sequences of full sentences in a bitext, but it is not so applicable if the alignment process is at the sub-sentence level, especially for language pairs with a significant linguistic distance from one another such as English and Japanese. To overcome this issue, we use the two-step alignment approach proposed by Deng et al. (2007), which aligns segments with DP and divisive clustering (DC) algorithms in a sequential manner. With this approach, the sub-sentential segments are aligned both monotonically and non-monotonically, and desirable quote TUs with finer granularity are collected.

4.1.1 Monotonic DP Alignment

The segment alignment process using DP is typically based on probabilistic models that employ features such as segment lengths and word alignment probabilities (Braune and Fraser, 2010; Mújdricza-Maydt et al., 2013). Deng et al. (2007) use length and word alignment features based on the Bayesian hierarchical model in Figure 1. In this figure $\boldsymbol{s}$ represents source-language (SL)

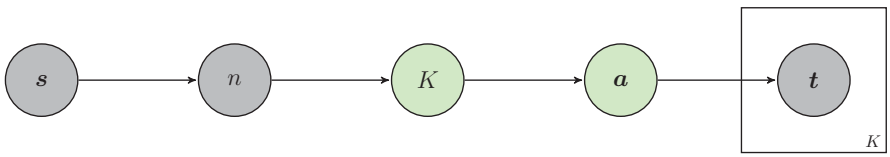
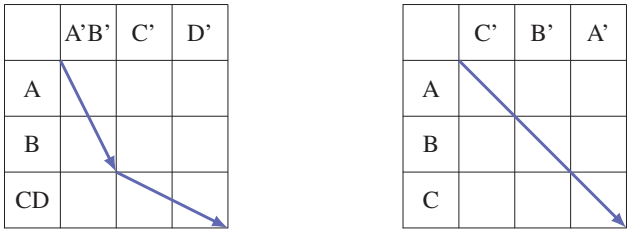


Figure 1: Graphical Model of DP Alignment (Deng et al., 2007)



(a) Monotonic Case (b) Non-Monotonic Case

Figure 2: Examples of DP Alignment with Monotonic and Non-Monotonic Pairs

segments of a certain length, measured by the segment count; n represents the number of target-language (TL) segments; K represents the number of aligned segment pairs; $\boldsymbol{a}$ represents the sequence of K aligned segment pairs (i.e., "beads" in Brown et al. 1991); and $\boldsymbol{t}$ represents the target segments, which can be split into K chunks to form aligned TUs with $\boldsymbol{s}$. The problem is to estimate K and $\boldsymbol{a}$ using the observed values $\boldsymbol{s}$, n, and $\boldsymbol{t}$. Based on this assumption, the segment alignment model is formulated as follows:

$$P(\boldsymbol{t}|\boldsymbol{s}, n, K, \boldsymbol{a}) = \prod_{k=1}^{K} P(\boldsymbol{t}_{a_k}|\boldsymbol{s}_{a_k}), \tag{1}$$

where

$$P(\boldsymbol{t}_{a_k}|\boldsymbol{s}_{a_k}) = \frac{P(u|v)}{(v+1)^u} \prod_{j=1}^{u} \sum_{i=0}^{v} t(f_j|e_i). \tag{2}$$

$P(\boldsymbol{t}_{a_k}|\boldsymbol{s}_{a_k})$ is an extension of the IBM1 model for an individually aligned TUs (See Brown et al. 1993). $P(u|v)$ in Equation 2 is the segment length model where u is the word count of SL segments in the bead a_k, v is the word count of the TL segments in a_k, and $t(f_j|e_i)$ is the word alignment model where f and e indicate the SL and TL words in the bead, respectively. The DP algorithm searches for the optimal values of K and $\boldsymbol{a}$ which determine the best aligned TU sequence by maximizing $P(\boldsymbol{t}|\boldsymbol{S}, n, K, \boldsymbol{a})$ in Equation 1. This algorithm works effectively if the alignment process is monotonic, i.e., where there are no TL segments in a transposed order. Figure 2a shows a monotonic alignment example in DP. In this example, A, B, C, and D represent SL segments, and A', B', C', and D' are the corresponding translations. C and D are combined into one source segment, and A' and B' are combined into one target segment to artificially create differences in these hypothetical segment sequences. The optimal aligned TUs are correctly discovered with the alignment model in Equation 1 by pursuing the best scores yielded by the local model in Equation 2, as indicated by the arrows shown in Figure 2a. However, the non-monotonic case depicted in Figure 2b is problematic because the segment order in the TL is the complete opposite of that in the SL. In this case, the only valid alignment is A-B-C and C'-B'-A', which is the same as the entire original TU, unless the alignment algorithm allows for deletions and insertions (i.e., 1-0 and 0-1 mappings) in the bead types. Such transposed cases are highly likely to occur when the segments to be aligned are at the sub-sentence level. To address such problematic cases, we use the DC alignment method. In the following section, we describe this method in detail.

4.1.2 Divisive Clustering

The divisive clustering (DC) method described by Deng et al. (2007) is an effective approach to overcome the non-monotonic alignment problem. Figure 3 shows how the alignment is

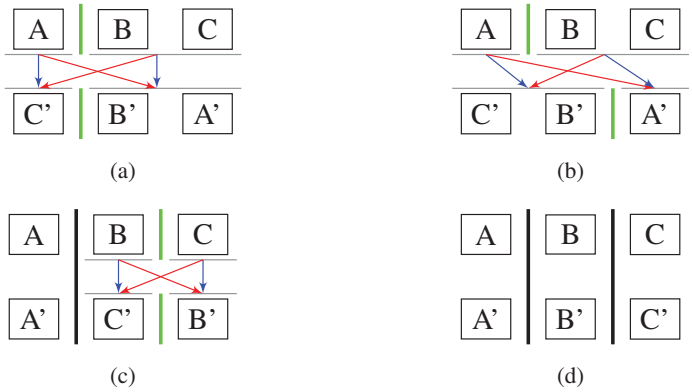


Figure 3: Example of Divisive Clustering Alignment

accomplished with the example case depicted in Figure 2b. First, the algorithm divides SL and TL segments, as shown by the green vertical lines in Figure 3a. Then it compares the divided segments in both monotonic and non-monotonic orders as indicated by the blue and red arrows in Figure 3a, and it records the respective alignment probabilities computed by the local model in Equation 2 as the costs. Next, it moves the split point of the TL segments, as shown in Figure 3b, and performs the same probability computation. The algorithm determines that the non-monotonic case in Figure 3b is the best among all four cost values, and reorders the TL segments so that A and A' and BC and C'B' are aligned, as shown in Figure 3c. Since A and A' consist of single segments, no further alignment process is applied. For BC and C'B', the same alignment process is followed, and the non-monotonic alignment case in Figure 3c is chosen. Since B-B' and C-C' consist of single segment pairs, the TL segments are simply reordered and the entire alignment process is completed.

In the overall alignment process, we use both DP and DC methods in a sequential manner, as described in Deng et al. (2007). We apply DP alignment to the original scripture TUs segmented only by terminal punctuation marks and then apply DC alignment to each of the resulting DP-aligned TUs, which are re-segmented by both terminal and non-terminal punctuation marks. We use this two-step alignment process in order to first obtain "large" aligned TUs with the DP alignment, and then process those TUs with the DC alignment to generate "small" aligned TUs. This approach is based on the assumption that the use of terminal punctuation marks between SL and TL is generally consistent; thus allowing the use of the time-efficient DP algorithm. Also, the resulting DP-aligned TUs narrow down the search space explored by the DC alignment, since the latter process is confined within each DP-aligned TU rather than the much larger original TU.

4.1.3 Iterative Segment Alignment

In typical scenarios, alignment algorithms collect TUs with high alignment probabilities computed by a simple model, such as a segment-length model, to obtain alignment features or train a word alignment model. With the features or model, the algorithm re-aligns the original TUs to identify aligned TUs with better accuracies as the final output (e.g., Moore 2002; Braune and Fraser 2010). This approach is effective only if the aligned TUs with high alignment probabilities are identified (i.e., precision-conscious alignment). However, this approach is not desirable in our scenario because quotes in sentences to be translated cannot be matched if only aligned TUs with high precision exist in the quotation TM. We need to collect aligned TUs with alignment scores that are as high as possible without sacrificing recall so that the quotation TMs can

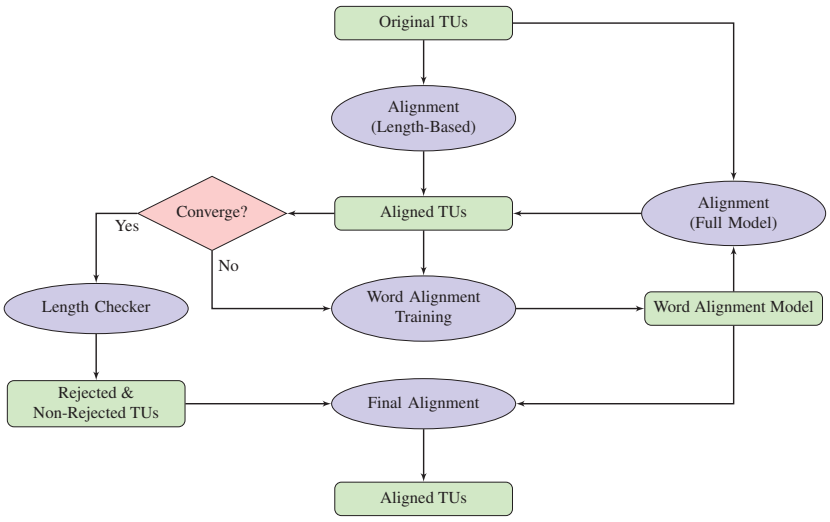


Figure 4: Iterative Segment Alignment Process

be used to match the largest number of potential quotes in the sentences.

To create a quotation TM, we use the alignment process illustrated in Figure 4, which begins with the original verse or sentence aligned TUs and iteratively identifies high probability sub-sentential TUs to be used in the creation of the quote TUs the TM will contain. First, the DP+DC algorithm aligns original TUs with a pre-computed length model to create the first collection of aligned sub-sentential TUs. This set of aligned TUs is then used for word alignment training. After the training process, the DP+DC process re-aligns the original TUs using the full model in Equation 2. If the alignment result is better than the previous alignment result, the new set of aligned TUs is used to train a new word alignment model and conduct the DP+DC alignment again. Once the alignment results converge and do not exhibit any improvement, the original TUs are aligned with the final version of the word alignment model in the final alignment process. Before this process, we evaluate the aligned TUs using a length checker based on the Poisson model used by Moore (2002) to avoid abnormal length discrepancies between the SL and TL segments of the aligned TUs. In this final alignment process, we aggressively re-align the TUs rejected by the length checker, segmenting them with all punctuation marks in both DP and DC processes. If the newly aligned TUs are accepted by the length checker, then we add them to the aligned TU set and export it as the output.

Figure 5 shows the increase in the aligned TU count for our English-Japanese (EN-JA) scripture TM data, which results from the iterative alignment process. The first alignment process using only the length model increases the aligned TU count slightly (44456 $\rightarrow$ 48899). Then there is a very substantial increase in the next iteration, where both length and word alignment models are used in the alignment process. In the next iterations, the aligned TU counts end up fluctuating somewhere between 117800 and 121550. We arbitrarily stopped the alignment process at iteration 15, assuming that the convergence point has been reached at or before this iteration. With the word alignment model created at this iteration, we aligned TUs using the aforementioned final alignment method, and obtained 143100 aligned TUs as the final output.

4.2 Quote Generation

Once the aligned TUs are generated, they are used to create quote TUs to be applied to sentences to be translated. To accomodate various types of quotes, we generate collections of segment *n*-

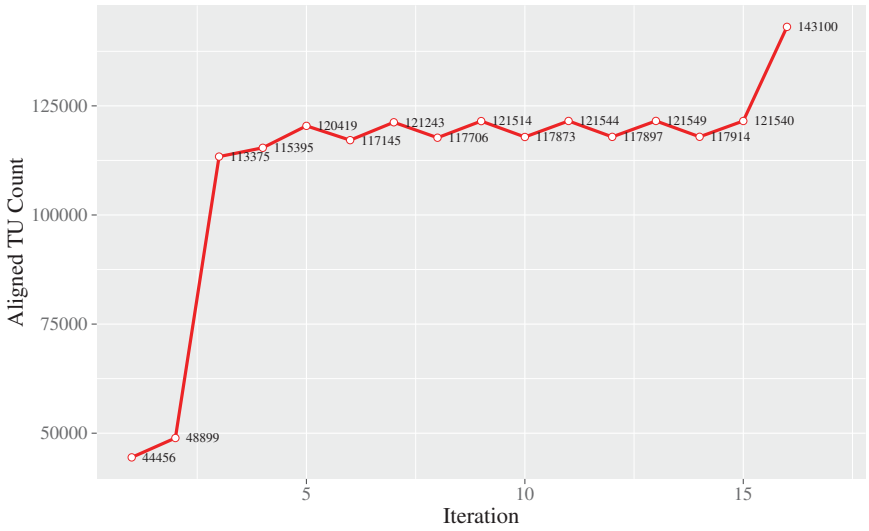


Figure 5: Aligned TU Increase over Alignment Iterations with EN-JA Scripture Dataset

grams with the aligned TUs. Figure 6 shows a simple example. Based on the identified aligned

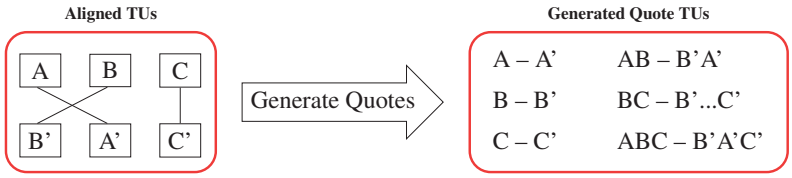


Figure 6: Example of Quote TU Generation using Aligned TUs

TUs, potential quote TUs are generated. In this case, six different quote TUs are generated based on the possible combinations of SL segments in the aligned TUs. Because the order of SL segments is different from that of the TL segments, appropriate treatments, such as swapping (e.g., A'B' $\rightarrow$ B'A' for AB) and ellipsis (e.g., B'...C' for BC) in Figure 6, need to be applied to the corresponding TL segments when segment *n*-grams are formed based on the SL segment order. To accomplish this, we keep track of the order of TL segments as they are aligned with SL segments using a generic tree data structure (i.e., a parent node with an arbitrary number of child nodes). The tree is then referred to in a depth-first order during the quote generation process as multiple SL segments are used to form a quote. If swapping or ellipsis cases are identified while traversing the tree, the corresponding treatments are applied. After quote TUs are generated and TL ordering is modified as needed, the TUs are stored in a quotation TM and applied to sentences to be translated as described in the following section.

5 Quote-Applied MT Inputs

Using the quotation TM, quote TUs are applied to sentences before they are machine-translated. Figure 7 shows an example of the quote application process. Once a sentence with an embedded quote is identified, the double- or single-quoted portion of the sentence is matched against the contents of the quotation TM. In our experiment, we used 98% as the fuzzy match threshold to

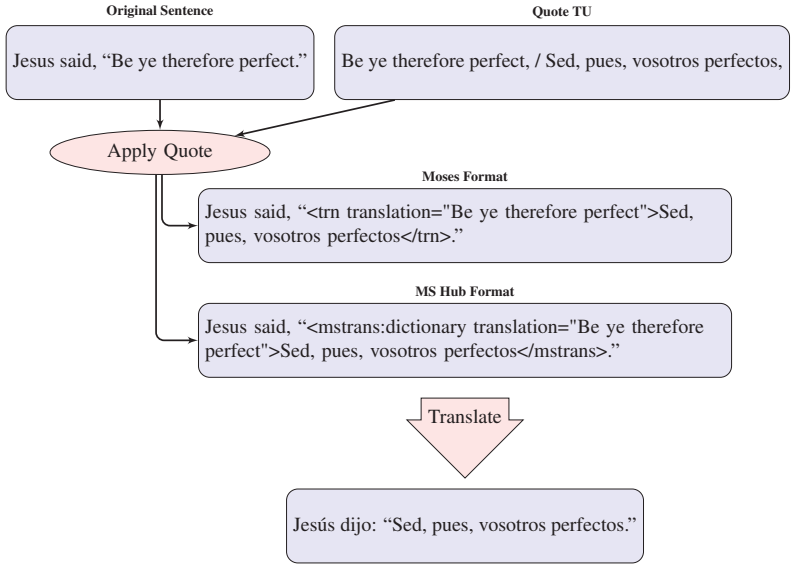


Figure 7: Example of Applying a Quote TU to a Sentence to be Machine-Translated

accommodate potential minor diffences in punctuation. If a match is found, the identified quote TU is applied to the sentence using a system-defined XML frame. Figure 7 shows example XML-framed sentences for the Moses toolkit[8](Koehn et al., 2007) and the Microsoft Translator Hub.[9] The modifed sentence is then sent to the MT system and translated.

6 Experiments

With the quote generation and application approach described above, we created scripture quotation TMs in eight language pairs and applied the quotes to test sets selected for experimentation. In the following section, we describe the experimental configuration.

6.1 Experiment Configuration

For the experiments, we used our in-house TM data for eight language pairs (with English as SL) to build trained MT systems and conduct the evaluation. The dataset sizes (based on the number of English segments) are shown in Table 1. The MT engine used for the experiments was the Microsoft Translator Hub.[10] For system tuning, we provided separate tuning sets (2500 TUs) extracted from the same data source as the training sets. The test sets consisted of TUs collected from the speeches given at the two most recent semi-annual general conferences of the Church held in October 2015 and April 2016,[11] which are not contained in the training sets. We cleaned the training and test datasets with Okapi[12] and segmented them using in-house

[8]Any arbitrary tag names other than “trn” can be used. See `http://www.statmt.org/moses/?n=Advanced.Hybrid` for more detail.

[9]Usage is described at `https://social.msdn.microsoft.com/Forums/en-US/de55e04f-7bc8-4a03-8a6d-13d43b2f739b/wordaround-about-donottranslate-list?forum=translatorhub`

[10]`https://hub.microsofttranslator.com`

[11]The texts of these speeches are available in over 90 languages at `https://www.lds.org/general-conference/2015/10` and `https://www.lds.org/general-conference/2016/04`, respectively.

[12]`http://okapiframework.org/`

Language	Training Set Size		Test Set Size	
	#TUs	#Tokens	#Sentences	#Tokens
Spanish (ES)	2,103,203	28,500,765	7,228	170,220
Portuguese (PT)	1,605,441	23,315,397	8,096	161,120
German (DE)	1,178,506	17,447,421	8,305	161,604
French (FR)	1,544,140	21,762,962	8,358	171,837
Italian (IT)	1,538,076	22,717,683	8,851	183,928
Russian (RU)	1,247,934	17,265,925	9,093	184,289
Japanese (JA)	1,026,201	15,801,673	6,385	150,647
Chinese (ZH)	1,233,531	18,809,448	6,208	136,248

Table 1: MT Training and Test Datasets

SRX[13] rules. The data were also sentence-aligned with an aligner similar to that of Moore (2002). Unaligned segments were excluded to reduce noise. The scripture quotation TMs for the eight languages were prepared using the quote generation process described in Section 4. They included all the quote TUs derived from verse-aligned TMs containing the four volumes of scripture described in Section 2.

7 Results

We examined the effectiveness of the quote application method with an automatic evaluation using case-sensitive BLEU (Papineni et al., 2002) and a human evaluation using Dynamic Quality Framework[14] (DQF, TAUS 2016). The details are described in the following sections.

7.1 Automatic Evaluation

Language	#Sentences	No Quotes Applied	Quotes Applied	Difference
ES	7,228	38.18	39.58	1.40
PT	8,096	40.15	41.63	1.48
DE	8,305	27.11	29.41	2.30
FR	8,358	39.82	41.02	1.20
IT	8,851	37.36	38.73	1.37
RU	9,093	28.15	29.59	1.44
JA	6,385	19.58	22.41	2.83
ZH	6,208	18.68	21.72	3.04

Table 2: Translation Quality Results (BLEU) of All Test Sentences

Table 2 shows the evaluation results for the test set of each language. The column "No Quotes Applied" indicates the BLEU scores for the entire test sets without applying any quotes (i.e., regular MT output). The column "Quotes Applied" shows the scores for the test sets with scripture quotes applied to the source sentences before being machine-translated. The column "Difference" indicates the difference in BLEU score between No Quotes Applied and Quotes Applied. As expected, applying quotes produces better results across all eight languages by

[13] http://www.ttt.org/oscarStandards/srx/

[14] http://dqf.taus.net

roughly 1.2 to 3 BLEU points, since we are substituting accurate reference translations in portions of the MT output. Of course, the improvement could also be offset somewhat by agreement problems and other grammatical anomalies caused by inserting the quoted text. But this modest improvement in BLEU score across the entire test set does not really reflect the true impact of quote application, since it is also dependent on the number of sentences in the test sets that actually contain quotes. To focus particularly on those test sentences containing quotes, we computed their BLEU scores separately. These scores and their relative impact are shown in Table 3. In this case, the score differences are much more significant, extending roughly from

Language	#Sentences	No Quotes Applied	Quotes Applied	Difference
ES	426	47.18	59.52	12.34
PT	408	50.55	61.70	11.15
DE	401	41.78	66.42	24.64
FR	415	52.80	64.47	11.67
IT	428	45.47	59.19	13.72
RU	280	41.93	61.31	19.38
JA	369	22.44	44.28	21.84
ZH	357	25.88	48.72	22.84

Table 3: Translation Quality Results (BLEU) of Quote-Applied Sentences

11 to 24 BLEU points, and convincingly demonstrating the positive effect of quote application. In comparing the differences across the eight languages, we observe that quote application has a somewhat lesser positive effect on those languages that are closer to English in grammar and syntax (e.g., Spanish, Portuguese, French, and Italian, with differences between 11 and 13 BLEU points), and a greater positive effect on the languages that are more divergent from English (e.g., German, Russian, Japanese, and Chinese, with differences between 19 and 24 BLEU points). This only makes sense, given that the BLEU scores of these latter systems are generally lower than the former ones.

7.2 Human Evaluation

Table 4 shows the DQF MT system comparison results of three languages (Portuguese, German, and Japanese). For this evaluation, we randomly chose 100 translations of the quote-applied sentences and the corresponding regular MT translations. The evaluation of each language was conducted by two bilingual raters: Rater 1 was a non-translator, and Rater 2 was a professional translator. In the evaluation process, the raters were asked to rank the translations of each of the 100 English sentence (ties were allowed). The percentages in the third and fourth columns of Table 4 are weighted rank scores reported by the DQF system, and the numbers in the parentheses indicate the number of times that the raters chose the translations generated by a particular method (Quotes Applied or No Quotes Applied) as being better than the other. For these numbers, we excluded the ties. The fifth column shows the number of rankings in agreement between the two raters.

Overall, the Quotes Applied translations were chosen by the raters more often than the No Quotes Applied translations across all three languages. This correlates strongly with the BLEU score results in Table 3. In particular, the Quotes Applied translations were most often preferred by German raters. For the other two languages, although the professional translators were more conservative about choosing Quotes Applied translations than non-translators, they still preferred Quote Applied translations significantly more often than No Quote Applied ones. We speculate that accurate machine translations of the quoted text, together with poten-

Language	Rater	Weighted Ranking Score		#Agreement
		No Quotes Applied	Quotes Applied	
PT	1	38.80% (6)	61.20% (77)	82
	2	42.12% (20)	57.88% (69)	
DE	1	36.93% (7)	63.07% (87)	96
	2	37.54% (7)	62.46% (84)	
JA	1	37.30% (5)	62.70% (84)	87
	2	44.05% (12)	55.95% (52)	

Table 4: Human Evaluation Results of Three Languages

tial agreement and other grammatical issues arising from the insertion of the quotes, may have influenced some of the choices made by the raters. Nevertheless, the results of these three language evaluations confirm the potential benefit of the quote application method in the machine translation of documents containing frequent scripture quotes.

8 Conclusions

In this paper we discussed our approach to enhance MT output using quotation TMs. We demonstrated the positive effect of the quote application process with real-world data for eight major languages used in a production translation environment, as measured by both automatic and human evaluations. A key factor in the success of creating and using a quotation TM, which is also a unique contribution of this work, is the generation of quote TUs, which are linguistically well-formed sub-sentential quote segments, aligned with their corresponding translations with a high degree of accuracy.

Although our focus was particularly on the translation of quotes in documents from the religion domain, our approach may be utilized in other translation scenarios and domains as well. For example, a document to be translated may contain quotes from canonical or standard texts in a heterogeneous domain. The texts containing those quotes can be processed to create a quotation TM, and quote TUs from this TM can be applied to sentences to be translated if the quoted portions in the sentences are correctly identified. Such simple domain adaptation approaches can be effective in translating mixed-domain documents more accurately in a production environment.

For future work, we will investigate approaches to dynamically generate quote TUs based on finding quotable text in a sentence without relying on segments generated by punctuation marks. A challenge in this case will be to capture the corresponding translation for the quotable text in order to create the sub-sentential quote TU. To address this problem, we will explore the idea of forced alignment in speech recognition, or a similar concept, as a good starting point (e.g., Alkhouli et al. 2016). If accurate forced alignment of words in each quote TU is possible, and if it can be obtained automatically with existing or new techniques, then it will be possible to extract the corresponding TL segment and form a quote TU dynamically.

Acknowledgments

We thank William Byrne at SDL and Cambridge University for making the original sentence alignment code available. We also thank Ryan Lee at the LDS Church for providing support for the experiments and for his insightful comments on this study.

References

Alkhouli, T., Bretschner, G., Peter, J.-T., Hethnawi, M., Guta, A., and Ney, H. (2016). Alignment-based neural machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 54–65, Berlin, Germany. Association for Computational Linguistics.

Biçici, E. and Dymetman, M. (2008). Dynamic translation memory: using statistical machine translation to improve translation memory fuzzy matches. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 454–465. Springer.

Braune, F. and Fraser, A. (2010). Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 81–89. Association for Computational Linguistics.

Brown, P. F., Lai, J. C., and Mercer, R. L. (1991). Aligning sentences in parallel corpora. In *Proceedings of the 29th annual meeting on Association for Computational Linguistics*, pages 169–176. Association for Computational Linguistics.

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Dandapat, S., Morrissey, S., Way, A., and Van Genabith, J. (2012). Combining EBMT, SMT, TM and IR technologies for quality and scale. In *Proceedings of the Joint Workshop on Exploiting Synergies between Information Retrieval and Machine Translation (ESIRMT) and Hybrid Approaches to Machine Translation (HyTra)*, pages 48–58. Association for Computational Linguistics.

Deng, Y., Kumar, S., and Byrne, W. (2007). Segmentation and alignment of parallel text for statistical machine translation. *Natural Language Engineering*, 13(3):235–260.

Espla-Gomis, M., Sánchez-Martínez, F., Forcada, M. L., et al. (2011). Using machine translation in computer-aided translation to suggest the target-side words to change. Machine Translation Summit.

Gale, W. A. and Church, K. W. (1993). A program for aligning sentences in bilingual corpora. *Computational linguistics*, 19(1):75–102.

He, Y., Ma, Y., van Genabith, J., and Way, A. (2010). Bridging SMT and TM with translation recommendation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 622–630. Association for Computational Linguistics.

He, Y., Ma, Y., Way, A., and van Genabith, J. (2011). Rich linguistic features for translation memory-inspired consistent translation. In *Proceedings of the Thirteenth Machine Translation Summit*, pages 456–463.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Koehn, P. and Senellart, J. (2010a). Convergence of translation memory and statistical machine translation. In *Proceedings of AMTA Workshop on MT Research and the Translation Industry*, pages 21–31.

Koehn, P. and Senellart, J. (2010b). Fast approximate string matching with suffix arrays and A* parsing. In *Meeting of the Association for Machine Translation of the Americas (AMTA)*.

Li, L., Escartín, C. P., and Liu, Q. (2016). Combining translation memories and syntax-based smt. *Baltic Journal of Modern Computing*, 4(2):165–177.

Ma, Y., He, Y., Way, A., and van Genabith, J. (2011). Consistent translation using discriminative learning: a translation memory-inspired approach. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1239–1248. Association for Computational Linguistics.

Moore, R. C. (2002). Fast and accurate sentence alignment of bilingual corpora. In *Conference of the Association for Machine Translation in the Americas*, pages 135–144. Springer.

Mújdricza-Maydt, É., Körkel-Qu, H., Riezler, S., and Padó, S. (2013). High-precision sentence alignment by bootstrapping from wood standard annotations. *The Prague Bulletin of Mathematical Linguistics*, 99:5–16.

Ortega, J. E., Sánchez-Martınez, F., and Forcada, M. L. (2014). Using any machine translation source for fuzzy-match repair in a computer-aided translation setting. In *Proceedings of the 11th Biennial Conference of the Association for Machine Translation in the Americas*, volume 1, pages 42–53.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Reinke, U. (2013). State of the art in translation memory technology. *Translation: Computation, Corpora, Cognition*, 3(1):27–48.

Richardson, S. (2012). Using the Microsoft Translator Hub at the Church of Jesus Christ of Latter-day Saints. In *AMTA 2012, Proceedings of the Tenth Conference of the Association for Machine Translation in the Americas*.

TAUS (2016). TAUS Quality Dashboard: From quality evaluation to business intelligence. `https://www.taus.net/component/rsfiles/download-file/files?path=Reports%252FFree%2BReports%252FQualityDashboardDocument-%2BMarch2016.pdf`. [Online; accessed September 1, 2016].

Wang, K., Zong, C., Su, K.-Y., et al. (2013). Integrating translation memory into phrase-based machine translation during decoding. In *ACL (1)*, pages 11–21.

Zhechev, V. and Van Genabith, J. (2010). Seeding statistical machine translation with translation memory output through tree-based structural alignment. Association for Computational Linguistics.