Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 4820–4827
Marseille, 11–16 May 2020

Improving NMT Quality Using Terminology Injection

Duane K. Dougal, Deryle W. Lonsdale
Brigham Young University
Provo, Utah, USA 84602
dkdougal@outlook.com, lonz@byu.edu

Abstract

Many organizations use domain- or organization-specific words and phrases. This paper explores the use of vetted terminology as an input to neural machine translation (NMT) for improved results: ensuring that the translation of individual terms is consistent with an approved multilingual terminology collection. We discuss, implement, and evaluate a method for injecting terminology and for evaluating terminology injection. Our use of the long short-term memory (LSTM) attention mechanism prevalent in state-of-the-art NMT systems involves attention vectors for correctly identifying semantic entities and aligning the tokens that represent them, both in the source and the target languages. Appropriate terminology is then injected into matching alignments during decoding. We also introduce a new translation metric more sensitive to approved terminological content in MT output.

Keywords: neural machine translation, NMT, terminology, term injection, attention vector

1. Introduction

Machine translation (MT) is a growing research area within natural language processing, a high-demand area of computer science. For many years, statistical MT (SMT) has dominated this research area and much progress has been made in MT using this approach and its variants, such as phrase-based MT. However, many believe, as quoted by a researcher in the field, that "the current approach of statistical, phrase-based MT has kind of reached the end of its natural life."[1] (Marking, 2016) With the recent resurgence of neural networks, advances in MT over the last two to three years are due to neural machine translation (NMT).

Most organizations use an increasing number of domain- or organization-specific words and phrases. Any associated translation process, whether human or automated, must accurately and efficiently use these specific multilingual terminology collections. Hasler et al. (2018) make the point that enforcing "terminology. . . is a requisite, . . . for companies wanting to ensure that brand-related information is rendered correctly and consistently when translating. . . and is often more important than translation quality alone." Cost savings are significant for organizations that manage and apply terminology (SDL, 2017), including in multilingual processes like translation.

However, comparatively little has been done to explore the integration of vetted terminology and MT processing, with the goal of improving overall results. In particular, manipulating the translation output of NMT systems to adhere to user-provided terminology specifications, despite the impressive quality improvements of NMT, remains an open problem (Hasler et al., 2018).

The now-common (and perhaps past its prime) SMT and the newer NMT both rely on training data and computed probabilities and weights to achieve their results. However, this does not always guarantee that the output results will contain the approved terminology, as is the case with Dinu et al. (2019). A sentence translated using MT may be grammatically or linguistically correct, but if it does not reflect terminological content mandated by the organization, the translation is suboptimal.

This paper presents a method for introducing approved terminology into the core of NMT processing. A type of terminology injection, it involves substituting or replacing an approved term where it doesn't emerge during translation. We show how it achieves improved terminology selection over most current state-of-the-art translation methods. We also introduce a new translation metric more sensitive to approved terminological content in evaluating MT output.

2. Background

In addressing the terminology challenge in MT, differing solutions have been implemented in the last few years. A common technique is to use rote dictionary lookup: substituting exact matches, typically in the output translation, from a table that specifies source/target terminology items (Luong et al., 2015b). This is usually possible since, if a source item is unknown to the MT system, it is often passed through untranslated to the output sentence, allowing it to be replaced automatically by the target entry from the dictionary. Direct string replacement in MT output may also be triggered by more active means, such as pre-annotating in the input (manually or otherwise) source terms with their translation equivalent for replacement in post-processing.

Another recent technique, constrained decoding, manipulates the data structure (a beam) that represents a specified number of output hypotheses, all scored statistically. The standard beam search algorithm that determines and returns the best-scored output is modified to allow for the satisfaction of constraints. In this case the constraints are essentially pre-specified sub-sequences of terminological content that can be injected into the beam search process, thus improving its chances of appearing in the output (Hokamp and Liu, 2017; Hu et al., 2019). However, probability still plays an important role in the final output. Although the output is constrained, probabilities in play at this point mean that the final output may or may not contain the constrained sequences. With terminology injection, our system precludes adjustment of output probabilities in the beam search, which could result in errors.

[1] Alan Packer, Engineering Director and head of the Language Technology team at Facebook

Dinu et al. (2019) propose a technique that involves inline-annotation of the input during training that specifies preferred translation units. The system learns to bias these input specifications, and fuzzy matching can abstract away from minor morphological variations. No constrained decoding or direct injection occurs.

Terminology content expresses semantic relationships—similarities and differences—between concepts, and the lexical content that encodes them in a context of domain knowledge. Effective translation relies upon the ability to faithfully communicate accurate concept equivalences as reflected in terminological units, beyond merely splicing together word sequences. To the extent that semantic principles find expression in a terminology-grounded MT task, the result will be more terminologically correct, naturalistic, and understandable.

It is often the case that currently approved terminology either has not been used historically and is not contained at all in any legacy texts, or it has been used only rarely. This poses another problem for an MT system: how to produce quality translations when there is no—or only minimal—approved terminology in the training data. This diverges from methods that intentionally include raw terminology in the training data (Dinu et al., 2019). On the other hand, new translations over time that take advantage of approved and integrated terminology will become part of the standard training data for the MT system. This will ensure that proper terminology becomes an integral part of future translations.

Related to the question of domain adaptation in MT, we evoke the issue in this paper but leave it for further future exploration.

2.1. NMT and terminology

Within the last few years, NMT has found traction in MT research and commercial use because of its ability to produce higher-quality and more-fluent translations than SMT (Denkowski and Neubig, 2017; Koehn and Knowles, 2017). We now sketch considerations for proper processing of terminological content in an NMT system. We do not provide a discussion of NMT architectures in general, which can be found elsewhere.

The training stage in NMT prepares a translation model from aligned bitexts in a given domain. An expensive and time-consuming process, it may not be practical to frequently retrain an NMT model to include the most current terminology in an active and growing or changing terminology collection. In this paper our proposed solution to this problem allows for an MT model of a particular domain to be augmented with both legacy translations and more up-to-date terminology to produce desirable results. This can mean lower cost and shorter time to delivery than is necessary with retraining a model to include evolving terminology.

When a word token is encountered in the training data, a vector representation is created for that word. An *n*-dimensional continuous vector space model (i.e., word embedding) encapsulates the vocabulary's characteristics as reflected by words' usages in the training data; one result is that semantically similar words are mapped to nearby points. Particular embedding methods vary, but they all reflect to some degree the Distributional Hypothesis, which states that "words that occur in the same contexts tend to have similar meanings" and are therefore semantically related (Pantel, 2005). This observation also extends to terminological units, hereafter TU's,[2] where similar terms will group proximally since TU's combine conceptual (semantic), lexical (terminological), and contextual (situational) perspectives (Cabré Castellví, 2003), all three of which are relevant and salient in the embeddings.

NMT training also produces a target vocabulary containing the individual and unique target tokens in the training corpora, sorted by frequency. Ideally, this vocabulary contains every token in the training data, though it is routinely limited in practice to a specific number of the most frequently occurring tokens. Jean et al. (2015) observe that the vocabulary of NMT has a limitation, as training complexity as well as decoding complexity increase proportionally to the number of target words. Recent techniques have been developed to mitigate these limits (Denkowski and Neubig, 2017); the OpenNMT system, which we use in this paper, has a default vocabulary limit of 50K words (Klein et al., 2017; Klein et al., 2018).

Part of the translation process is encoding, where the system re-expresses the human-readable source sentence into a mathematical representation for future use. This mathematical representation is known as the context vector (Luong et al., 2015a).

The decoding stage of the translation process renders the context vector from its mathematical representation to a human-readable sentence in the target language. This is done in a two-step process: first, generate a vector representation of the target sentence based on the trained model, then tokenize the target vector into a human-readable sentence using tokens from the target vocabulary.

As described by Luong et al. (2015a), the decoder generates one target word (token) at a time, using conditional probability to generate the best word based on those in previous time steps. This backward glance allows for multi-token terms (a complete TU) to be generated.

In decoding, the attention mechanism focuses on different parts of the source sentence that are more important at various stages of the translation as an alternative to just processing the source sentence sequentially. It also permits a view into what is going on in the decoder in a straightforward way.

Luong et al. (2015a) present both a global attention model and a local attention model. The idea of a global attention model is to consider all the hidden states of the encoder when deriving the context vector. In this model type, a variable-length alignment vector whose size equals the number of time steps on the source side, is derived by comparing the current target hidden state with each source hidden state.

The purpose of the attention mechanism is to generate the align weights. These align weights are used to produce an alignment between the source sentence and the target sen-

[2] not to be confused with "translation unit"; see (TERMIUMplus, 2019)

tence based upon the vector embeddings in the alignments while generating the target sentence. Luong et al. (2015a) observe that a by-product of attentional models are word alignments. These intrinsic word alignments, not inherently available in SMT, are the key to injecting terminology in NMT. Indeed, the foundational NMT work by Bahdanau et al. (2014) focuses on alignment as a way to improve translation.

The attention mechanism helps to focus on the most probable representations. Then, tokens from the target vocabulary that represent the TU's in the target vector are selected to complete the target translation. This is referred to as the prediction.

One caveat is in order: attention vectors provided by NMT are not a perfect representation for translation. NMT challenge #5 cited by Koehn and Knowles (2017) states that the attention model for NMT does not always fulfill the role of a word alignment model, and may at times dramatically diverge. That is not to say it doesn't work at all, rather that it is as yet imperfect and can be improved. They continue this idea by observing that the word attention states match up well with the word alignments obtained with "fast_align," a common alignment tool. However, they noticed that the attention model may settle on alignments that did not correspond with their intuition or alignment points obtained with fast_align.

Human-readable tokens do not exist in a context vector (the output of the encoder). Instead, the context vector is a semantic representation of the entire source sentence. When the context vector is passed to the decoder, a new vectorized semantic representation is generated that matches the target language. This is sometimes referred to as the hypothesis.

3. Approach

This paper reports on experiments involving English and Spanish as the source and target languages, respectively, mostly due to availability of relevant resources.

3.1. Terminology Injection

The crucial first step in generating the target sentence is decoding the context vector into a target sentence vector. The decoder uses the trained model and the vector-space model to generate a target sentence vector. Once the target sentence vector has been generated by the decoder, a target sentence token index is generated. The last step in building the target sentence is to select tokens in the target vocabulary that represent the TU's in the target sentence vector.

Terminology injection involves modifying the target sentence symbols (tokens). Recall that the tokens in the target token vocabulary are human-readable symbols that represent the TU's in the target sentence vector which are found in the training corpus. Of crucial importance is the existence of a mapping between the vector space and the tokens in the vocabulary. Therefore, it is possible to modify the vocabulary and replace a token with a different token. Tokens cannot be removed from the vocabulary, but they can be modified. Doing so will allow for replacement of a TU.

For example, an NMT system might translate the source sentence given in (1) as the viable Spanish translation given in (2).[3] However, in a given organizational context (3) or even (4) might instead be the approved translation.

(1) Your report is absolutely disgraceful.

(2) Su informe es absolutamente vergonzoso.

(3) Su exposición es absolutamente vergonzoso.

(4) Su exposición de alta calidad es absolutamente vergonzoso.

Terminology injection allows for the requisite manipulation of the target sentence in these cases. The internal state of the NMT system is open to inspection; crucially, for injection, the attention matrix can be examined for source/target alignments. Figure 1a shows different versions of the translation with their associated attention matrices. Angle brackets and red highlighting in the matrix indicate the locus of terminology substitution in the target output, illustrated in the first column.

	Your	report	is	absolutely	disgraceful	.
Su	0.2562	0.2725	0.0989	0.1220	0.1184	0.1320
informe	0.0308	0.8274	0.0238	0.0382	0.0567	0.0231
es	0.0302	0.0297	0.2937	0.1647	0.2682	0.2135
absolutamente	0.0274	0.0126	0.0337	0.4720	0.3788	0.0754
vergonzoso	0.0073	0.0027	0.0060	0.0297	0.9105	0.0437
.	0.0583	0.0148	0.0511	0.0201	0.0607	0.7949

(a) Raw NMT output based on trained bitext

	Your	report	is	absolutely	disgraceful	.
Su	0.2562	0.2725	0.0989	0.1220	0.1184	0.1320
<exposición>	0.0308	0.8274	0.0238	0.0382	0.0567	0.0231
es	0.0302	0.0297	0.2937	0.1647	0.2682	0.2135
absolutamente	0.0274	0.0126	0.0337	0.4720	0.3788	0.0754
vergonzoso	0.0073	0.0027	0.0060	0.0297	0.9105	0.0437
.	0.0583	0.0148	0.0511	0.0201	0.0607	0.7949

(b) NMT injected output (1-to-1) from a termbase

	Your	report	is	absolutely	disgraceful	.
Su	0.2562	0.2725	0.0989	0.1220	0.1184	0.1320
<exposición de alta calidad>	0.0308	0.8274	0.0238	0.0382	0.0567	0.0231
es	0.0302	0.0297	0.2937	0.1647	0.2682	0.2135
absolutamente	0.0274	0.0126	0.0337	0.4720	0.3788	0.0754
vergonzoso	0.0073	0.0027	0.0060	0.0297	0.9105	0.0437
.	0.0583	0.0148	0.0511	0.0201	0.0607	0.7949

(c) NMT injected output (1-to-many) from a termbase

Figure 1: Predictions using original and substituted tokens

Note that all values in the attention vectors are exactly the same in each table no matter the substitution. Since tokens in the target output are just string literals, we can substitute them freely from an external termbase.

A limitation of this method so far involves the fact that it can handle cases of 1-to-1 and 1-to-many replacement, where only a single source token can be handled; but

[3] Sentences (1) and (2) are from EuroParl bitext.

many-to-1 and many-to-many, which replace more than one source token, cannot.

Note that injection can also cause problems with proper linguistic agreement in the target language. The word "exposición" in Spanish is a feminine form while the word "informe" is masculine. Replacing "informe" with "exposición de alta calidad" causes an agreement problem because the adjective "vergonzoso" is a masculine form that agrees with the grammatical gender of the word "informe" but not with the grammatical gender of the word "exposición". In this case, a simple grammatical gender change from the masculine "vergonzoso" to the feminine "vergonzosa" would fix the problem. When a replacement token or sequence fails to agree with its new target context, correcting the output will require post-editing.

4. The algorithm

We now extend the current account to allow for variable-length source substitutions, building on the core idea of token replacement based on TU identification. Each row of the attention matrix pertains to a single token in the generated target sentence, whereas each column pertains to a single source sentence token. Table cells contain probability values that represent the likelihood of an association (i.e., the alignment) between the relevant source and target tokens in the current sentence. But now we can extend these attention probabilities to contiguous cells in both directions.

In both Figures 1b and 2a, the highest value in the column under the source token "report," a value of 0.8274, triggers the target token selection. That is, the token "informe" is the correct target equivalent of the source token "report" with an 82.74% confidence. The highest value in each column is indicated in Figure 2a and shows the relationship between each source token and its corresponding target token. In other words, it indicates an alignment.

Note that in the first two columns of Figure 2b, the highest values are both in the first row. This is because the two source tokens "We" and "must" both correctly align with the single target token "Debemos" in Spanish. This is an example of many-to-1 mapping between source and target tokens. The attention vectors in the matrix identify the translation equivalents in the matrix. This allows the NMT system to correctly produce a translation sequence of "Debemos" from "We must".

Figure 3 provides an example of a more complex sentence involving many-to-1 injection. It involves a many-to-many (3-to-4 term) injection with substantial reordering (the adjective "European" at the beginning of the source term has its corresponding Spanish translation ending up in term-final position in the target language).

We now sketch the process to correctly identify and apply appropriate terminology, given an external source/target termbase and an NMT alignment matrix. More details are available elsewhere (Dougal, 2018). This process will inject supplied terms into the NMT decoder output by matching the TU's in the attention matrix.

1. Create a "term candidates" collection by scanning the source sentence for all occurrences of any term from the terminology collection. Compute the longest-cover match between term candidates.

2. Generate the attention matrix using the attention vectors from the NMT decoder.

3. Map each token in the source sentence to a column in the attention matrix, token 0 to column 0, token 1 to column 1, and so on as shown in Figure 1.

4. For each term in the source sentence (possibly more than one token), identify the highest probability value in the column for both the first and last tokens, and identify the corresponding row in the attention matrix for each.

   The set of probability values, P, for a given column in the attention matrix, C, can be represented by Equation 1 as follows, where r is the total number of rows in the matrix and p is a single probability value in C. The entire attention matrix, then, can be expressed by Equation 2, where n is the total number of columns in

		C0	C1	C2	C3	C4	C5
		Your	report	is	absolutely	disgraceful	.
R0	Su	0.2562	0.2725	0.0989	0.1220	0.1184	0.1320
R1	informe	0.0308	0.8274	0.0238	0.0382	0.0567	0.0231
R2	es	0.0302	0.0297	0.2937	0.1647	0.2682	0.2135
R3	absolutamente	0.0274	0.0126	0.0337	0.4720	0.3788	0.0754
R4	vergonzoso	0.0073	0.0027	0.0060	0.0297	0.9105	0.0437
R5	.	0.0583	0.0148	0.0511	0.0201	0.0607	0.7949

(a) Attention vectors in the columns

	We	must	update	the	file	.
Debemos	0.1510	0.3075	0.2445	0.0323	0.0609	0.2038
actualizar	0.0106	0.0246	0.7341	0.0718	0.1002	0.0587
el	0.0208	0.0053	0.0357	0.2018	0.4018	0.3346
expediente	0.0028	0.0007	0.0137	0.0076	0.8947	0.0805
.	0.0211	0.0042	0.0377	0.0311	0.1778	0.7281

(b) Attention vectors in the rows

Figure 2: Attention vectors in the LSTM attention matrix

		C0	C1	C2	C3	C4	C5	C6	C7
		We	must	strengthen	the	European	Works	Councils	.
R0	Debemos	0.1396	0.3394	0.2986	0.0309	0.0336	0.0132	0.0215	0.1233
R1	reforzar	0.0095	0.0291	0.7231	0.0934	0.0251	0.0248	0.0451	0.0500
R2	los	0.0244	0.0092	0.0304	0.1472	0.0565	0.1068	0.3289	0.2966
R3	comités	0.0023	0.0017	0.0094	0.0046	0.0230	0.2336	0.6981	0.0273
R4	de	0.0028	0.0026	0.0064	0.0055	0.1637	0.4767	0.2936	0.0487
R5	empresa	0.0000	0.0000	0.0007	0.0008	0.0057	0.9297	0.0612	0.0018
R6	europeos	0.0182	0.0102	0.0343	0.0122	0.7237	0.0650	0.0502	0.0863
R7	.	0.0508	0.0223	0.0682	0.0090	0.0678	0.0129	0.0277	0.7413

Figure 3: TU identification in a more complex sentence

the matrix. It follows, then, that the row of a given token in target term T in attention matrix M can be expressed as R_{S_i}, where R is the row in M of the i^{th} token in the current source term, S, as given in Equation 3. Note that the $argmax$ function returns the $index$ of the max element of the specified array, and not the max element itself.

$$P_C = \{p_0, p_1, p_2, \cdots, p_r\} \quad (1)$$

$$M = \{P_0, P_1, P_2, \cdots, P_n\} \quad (2)$$

$$R_{S_i} = argmax(P_{C_i}) \quad (3)$$

Figure 3 includes a multi-word or multi-token source term, "European Works Councils." As mentioned previously, the target equivalent in Spanish is also a multi-token term, but due to text expansion it contains more tokens than the source equivalent. Correct identification of the entire TU in the target sentence requires an additional pair of functions. These functions are applicable to terms with any number of tokens, where B represents the beginning token in the TU and E represents the ending token in the TU:

$$B = min(R_{S_{first}}, R_{S_{last}}) \quad (4)$$

$$E = max(R_{S_{first}}, R_{S_{last}}) \quad (5)$$

Therefore, the TU in the source sentence is represented by source tokens 4 through 6, whereas the TU in the target sentence is represented by target tokens 3 through 6.

5. Look up the source term for the optimized sequence in the terminology collection and retrieve the target equivalent, which are then substituted in place of the original target tokens in the NMT prediction.

5. Experiments

For training and testing the NMT model, we used three pertinent language resources. The base system we used was the OpenNMT PyTorch system (Klein et al., 2017; Klein et al., 2018), an open source deep-learning platform; additional code for our new functionality was also written in Python. To provide a baseline, we trained the system on one million EuroParl sentence pairs for 10 epochs. Though the system default is 13 epochs, 10 seemed appropriate for our purposes. The nearly two-week training stage produced a model consisting of a 2-layer Long Short-Term Memory (LSTM) network with 500 hidden units on both the encoder and decoder. The system worked well, even using default settings.[4]

EuroParl: This is a multilingual bitext corpus containing transcriptions of the proceedings of the European Parliament, therefore involving a domain of government and politics.[5] It is commonly used for MT evaluations. We used the first half of the 2 million aligned English-Spanish sentence pairs to train the system.

Microsoft C# technical documentation: We collected this corpus by scraping the Microsoft Developer Network (MSDN) website.[6] It contains high-quality pre-aligned bitexts for a variety of language pairs; our corpus comprises 10,000 aligned sentences. We selected this corpus primarily because of its very different domain from EuroParl, allowing us to evaluate the effect of terminology injection on NMT output from a largely unmatched training domain. In the real world, this is actually a very common scenario and subsequently a very useful evaluation. In fact, Koehn and Knowles (2017) list raising the quality of out-of-domain translations as the first of "Six Challenges for Neural Machine Translation."

Microsoft terminology: We also used an existing terminology collection available from Microsoft. We converted a publicly available TermBase eXchange (TBX) file directly from the Microsoft Language Portal, thus obtaining both English and Spanish equivalent terms in approximately 30,000 concept entries.[7]

6. Results and discussion

For our evaluation we used the NMT model trained on EuroParl as our baseline system. We then used terminology injection with an externally specified termbase to see what effect would be produced. See Table 1 for comparative statistics regarding both text types and characteristics of the terminology matches.

We then computed scoring metrics for the raw output. Finally, we scored the output with terminology injection, and calculated the difference from the raw output (baseline) scores. As is common practice, we report our results using standard metrics: the Bilingual Evaluation Understudy (BLEU) score (Papineni et al., 2002) and its variants Multi-BLEU and NIST-BLEU.

We first attempted to evaluate terminology injection on the EuroParl corpus using IATE, the official EU terminology resource, as a termbase input.[8] Coverage for EuroParl term usage, though, was very low, hence yielding somewhat disappointing results: with injection the Multi-BLEU and NIST-BLEU scores showed consistently slight decreases. On the other hand, as expected, coverage by the Microsoft termbase of terms used in the Microsoft documents was high, and as a result terminology injection was substantial and yielded excellent results (see Table 2). Both terminology collections used in this study—IATE and the Microsoft collection—are the vetted and recommended terminology collections for the respective datasets.

Our results demonstrate a significant improvement in NMT output using terminology injection versus raw output for the Microsoft content. Furthermore, our results, based on limited data, indicated that percentage improvement increases, on average, with the size of the corpus as measured with BLEU, though this may not be true in all cases. However, more evaluations will be necessary to validate that as-

[4] http://opennmt.net/OpenNMT-py/options/train.html

[5] www.statmt.org/europarl/

[6] https://docs.microsoft.com/en-us/dotnet/csharp/

[7] https://www.microsoft.com/en-us/language

[8] See https://iate.europa.eu.

		Average words per sentence	Average terms per sentence	Average term n-gram size	% of sentences with terms
EuroParl	**Source**	28.86	0.65	2.07	42.61
	Target	27.83	0.63	2.47	
Microsoft	**Source**	16.76	3.35	1.10	90.83
	Target	16.39	3.14	1.19	

Table 1: Document and translation characteristics

Dataset	Sentences	Multi-BLEU			NIST-BLEU		
		Raw	Result	Gain	Raw	Result	Gain
EuroParl 1K	997	33.31	-0.67	-2.01%	33.57	-0.68	-2.03%
EuroParl 3K	3,497	33.60	-0.23	-0.68%	33.82	-0.23	-0.68%
EuroParl 5K	5,000	32.00	-0.48	-1.50%	32.27	-0.47	-1.46%
EuroParl 10K	9,934	32.25	-0.30	-0.93%	32.51	-0.32	-0.98%
Microsoft Original	1,003	26.03	**+1.70**	**6.53%**	26.30	**+1.77**	**6.73%**
Microsoft Combo	1,103	26.00	**+1.90**	**7.31%**	26.32	**+2.31**	**8.78%**
Microsoft Medium	3,000	23.00	**+2.81**	**12.22%**	23.43	**+2.85**	**12.16%**
Microsoft Large	10,000	23.77	**+2.95**	**12.41%**	24.39	**+2.98**	**12.22%**

Table 2: BLEU performance data for datasets using terminology injection

sertion or to determine in what circumstances it may apply. This is particularly interesting since the Microsoft dataset reflects out-of-domain results.

These results are very different from the expectation of an improvement score that remains uniform and average regardless of the size of the dataset. Still, these data support the finding that our injection method improves NMT output by a significant margin (Koehn, 2004).

7. Introducing TREU

Several considerations documented elsewhere (Coughlin, 2003; Koehn, 2004; Och, 2003; Papineni et al., 2002) make BLEU somewhat suboptimal for evaluating translations that have dense terminological content. In this section we introduce a new metric that is comparable to BLEU and is more sensitive to terminological content. We call this scoring algorithm Terminology Recall Evaluation Understudy (TREU). Consider the following three tokenized sentences.

(5) The dog chased the cat . (*Refs*)

(6) The dog chased the cat . (*Pred*)

(7) The dog chased the {feline} . (*Pmod*)

Note that (7) is identical to (5) with the exception that the term "feline" is injected and tagged, replacing "cat" in (7). Though the terms are semantically similar, a preference was made to use the term "feline" instead. Using BLEU, these two sentences would not receive the same score since BLEU is based on orthographic similarity. Thus BLEU inherently penalizes use of approved terminology even when it is used appropriately. TREU scores translation output using a combination of standard string-matching metrics while also taking terminology injection into account.

Following is the process for calculating TREU scores:

1. Calculate the overlap between the reference sentence, *Refs*, and the raw prediction, *Pred*, as well as between *Refs* and the modified prediction, *Pmod*.

The overlap, o, between two sentences, S_1 and S_2, is the summation of the minimum occurrence of all tokens common between the two sentences. A token, t, is an element of the shared vocabulary, V, which is the set of all unique tokens common to both sentences. If a token exists in one sentence but not the other, that token does not "overlap" both sentences and it is not counted.

$$o = \sum_{t \in V} \min(count(t, S_1), count(t, S_2)) \quad (6)$$

A credit value, C, is necessary to account for terminology tokens, t', that will not be counted when calculating the overlap, o, between the reference sentence, *Refs*, and the modified prediction, *Pmod*, using orthographic matching. Note that C is only calculated on a translated sentence into which terminology has been injected. If neither S_1 nor S_2 is the modified prediction, *Pmod*, there is no need to provide a credit for semantic equivalence and C will be zero. Of necessity, the calculation of C assumes that all terminology injections are correct.

$$C = count(t', Pmod) \quad (7)$$

The complete overlap, O, is the sum of the initial overlap, o, and the terminology credit, C.

$$O = o + C \quad (8)$$

For example, if we assume that (5) is the reference sentence, *Refs*, and (6) is the prediction, *Pred*, the value of O is 6 when comparing these two sentences: The number of orthographically identical tokens is 6 and no terminology credit is needed, leaving C at zero ($O = 6 + 0$). However, when comparing (5) and (7), O is still 6 even though "cat" and "feline" are orthographically distinct. In this case, there are 5 orthographically identical tokens in common between the two sentences ($o = 5$) and the terminology credit, C,

is 1 because the term "feline" is tagged in (7). This allows us to accept semantic equivalence between two terms despite orthographic differences.

2. Once the values for overlap have been calculated for both *Pred* and *Pmod*, compute the value for Recall. Count the total number of tokens in *Refs* to compute Recall using the overlap results.

$$r = \begin{cases} 1, & \text{if } O > count(t, Refs) \\ \frac{O}{count(t,Refs)}, & \text{if } O = count(t, Refs) \end{cases} \quad (9)$$

$$R = \begin{cases} r, & \text{if } O > 0 \\ 0, & \text{if } O = 0 \end{cases} \quad (10)$$

3. While Recall, R, (Equation 10) will always rely upon the number of tokens in the reference sentence, *Refs*, the Precision equation, P, (Equation 12) will use either the unmodified prediction, *Pred*, or the modified prediction, *Pmod*, depending upon the intended comparison. Therefore, φ is used to represent either *Pred* or *Pmod*, as the case may be. Compute Precision accordingly.

$$p = \begin{cases} 1, & \text{if } O > count(t, \varphi) \\ \frac{O}{count(t,\varphi)}, & \text{if } O = count(t, \varphi) \end{cases} \quad (11)$$

$$P = \begin{cases} p, & \text{if } O > 0 \\ 0, & \text{if } O = 0 \end{cases} \quad (12)$$

4. Given these values, the F_1 score is computed as usual, which is then used to compare sentences.

These equations are based on the standard equations for the Recall, Precision, and F_1 metrics. Table 3 shows scoring over the baseline for both datasets as described by TREU, the new metric more sensitive to terminology matches from a standard terminology resource. TREU shows an improvement over the basic NMT baseline when terminology injection is used, even when the match between the termbase (IATE) and the text (EuroParl) is not a very good one, as discussed earlier. As shown in the Microsoft listings, results are even more impressive when the termbase/text match is better (compare Table 1).

Dataset	Sentences	TREU		
		Raw	Result	Gain
EuroParl 1K	997	0.6523	**+0.0159**	**2.44%**
EuroParl 3K	3,497	0.6455	**+0.0132**	**2.04%**
EuroParl 5K	5,000	0.6334	**+0.0138**	**2.18%**
EuroParl 10K	9,934	0.6296	**+0.0213**	**3.38%**
Microsoft Original	1,003	0.6258	**+0.0716**	**11.44%**
Microsoft Combo	1,103	0.6255	**+0.0726**	**11.60%**
Microsoft Medium	3,000	0.5943	**+0.0578**	**9.73%**
Microsoft Large	10,000	0.5963	**+0.0577**	**9.67%**

Table 3: TREU measurements for terminology injection

According to Koehn (2004), if the BLEU score difference is at least 2-3% for a test set size of 300 sentences or more then the results are within the 95% statistical significance range. Although TREU and BLEU are distinct metrics and are used for different purposes, they are comparable in their methods. Table 3 shows that the percentage gain for all datasets used in this study falls within the statistical significance range indicated by Koehn (2004).

8. Conclusions and Future Work

Given the need for correct usage of vetted terminology in the translation process, an effective integration with termbases in various MT approaches is desirable. This is especially true for NMT, which already often produces more natural or "fluent" translations. Terminology injection leverages the inherent advantages of expanded context (via the LSTM capability) and semantically expressive word embeddings. By associating a bilingual termbase with an NMT system and associating source and target terms with the alignment matrix, terms can be injected even when they involve length mismatches or reorderings.

Using standard MT evaluation metrics, we show that injection is effective, especially when the termbase's coverage in the source text is more extensive. The Microsoft technical documentation and associated terminology collection are good illustrations of this.

We also proposed another MT metric, TREU, that is more sensitive to the presence of approved terminology in the target translation. We showed that, even in domains where termbase coverage is limited—such as the EuroParl proceedings—TREU reflects an increase over the baseline because of the terms that injection does recognize and process.

Several avenues of future research are possible to continue this work.

As discussed earlier, injection of target terminology sometimes introduces target-language grammatical errors when incompatibility with the rest of the sentence arises. Perhaps the NMT decoder could be harnessed to provide correct agreement processing, or more likely a separate trained LSTM network could be used to make proper adjustments to the target sentence to correct problems that arise.

The current implementation only uses a flat list of non-hierarchical source-target term pairs (often referred to as a "glossary") to seed the injection process, which increases the likelihood of polysemic collisions. On the other hand, terminology management practices involve much more complex termbases that express several types of lexical, domain, semantic, hierarchical (taxonomic), ontological, overlapping, and nesting relationships among terms. A more complex interaction between sophisticated concept-based termbase content and the injection process should be possible.

Finally, more empirical work could be done to extend the evaluation:

- study human judgments of quality and correctness of the TU-injected output and their correlation with TREU metrics;
- run comparative tests against the two alternative processing modalities discussed earlier; and
- quantify how well the system scales up with more training data, larger termbases, and larger test sets.

9. Bibliographical References

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *Computing Research Repository (CoRR)*, abs/1409.0473.

Cabré Castellví, M. T. (2003). Theories of terminology: Their description, prescription and explanation. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, 9(2):163–199.

Coughlin, D. (2003). Correlating automated and human assessments of machine translation quality. In *Proceedings of MT summit IX*, pages 63–70.

Denkowski, M. and Neubig, G. (2017). Stronger baselines for trustable results in neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 18–27. Association for Computational Linguistics.

Dinu, G., Mathur, P., Federico, M., and Al-Onaizan, Y. (2019). Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy, July. Association for Computational Linguistics.

Dougal, D. K. (2018). Improving the quality of neural machine translation using terminology injection. Master's thesis, Brigham Young University, Provo, UT.

Hasler, E., Gispert, A., Iglesias, G., and Byrne, B. (2018). Neural machine translation decoding with terminology constraints. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 506–512. Association for Computational Linguistics.

Hokamp, C. and Liu, Q. (2017). Lexically constrained decoding for sequence generation using grid beam search. *CoRR*, abs/1704.07138.

Hu, J. E., Khayrallah, H., Culkin, R., Xia, P., Chen, T., Post, M., and Van Durme, B. (2019). Improved lexically constrained decoding for translation and monolingual rewriting. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 839–850, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Jean, S., Cho, K., Memisevic, R., and Bengio, Y. (2015). On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10. Association for Computational Linguistics.

Klein, G., Kim, Y., Deng, Y., Senellart, J., and Rush, A. (2017). OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada, July. Association for Computational Linguistics.

Klein, G., Kim, Y., Deng, Y., Nguyen, V., Senellart, J., and Rush, A. (2018). OpenNMT: Neural machine translation toolkit. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 177–184, Boston, MA, March. Association for Machine Translation in the Americas.

Koehn, P. and Knowles, R. (2017). Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39. Association for Computational Linguistics.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In Dekang Lin et al., editors, *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 388–395. Association for Computational Linguistics.

Luong, M.-T., Pham, H., and Manning, C. D. (2015a). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1412–1421. Association for Computational Linguistics.

Luong, T., Sutskever, I., Le, Q., Vinyals, O., and Zaremba, W. (2015b). Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 11–19. Association for Computational Linguistics.

Marking, M. (2016). Facebook says statistical maching translation has reached end of life. In *Slator: Language Industry Intelligence*. https://slator.com/technology/facebook-says-statistical-machine-translation-has-reached-end-of-life/.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 160–167. Association for Computational Linguistics.

Pantel, P. (2005). Inducing ontological co-occurrence vectors. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL '05)*, pages 125–132. Association for Computational Linguistics.

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 of the Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.

SDL. (2017). Infographic: What cost savings can be achieved through terminology management? https://www.sdl.com/software-and-services/translation-software/infographic/terminology-management-cost-savings.html.

TERMIUMplus. (2019). Government of Canada / Gouvernement du Canada, December. Record for "terminology unit", queryable at https://termiumplus.gc.ca.