2018-12-01

Improving the Quality of Neural Machine Translation Using Terminology Injection

Duane K. Dougal
Brigham Young University

Improving the Quality of Neural Machine Translation Using Terminology Injection

Duane K. Dougal

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Deryle W. Lonsdale, Chair
Parris K. Egbert
Casey T. Deccio

Department of Computer Science

Brigham Young University

ABSTRACT

Improving the Quality of Neural Machine Translation Using
Terminology Injection

Duane K. Dougal
Department of Computer Science, BYU
Master of Science

Most organizations use an increasing number of domain- or organization-specific words and phrases. A translation process, whether human or automated, must also be able to accurately and efficiently use these specific multilingual terminology collections. However, comparatively little has been done to explore the use of vetted terminology as an input to machine translation (MT) for improved results. In fact, no single established process currently exists to integrate terminology into MT as a general practice, and especially no established process for neural machine translation (NMT) exists to ensure that the translation of individual terms is consistent with an approved terminology collection.

The use of tokenization as a method of injecting terminology and of evaluating terminology injection is the focus of this thesis. I use the attention mechanism prevalent in state-of-the-art NMT systems to produce the desired results. Attention vectors play an important part of this method to correctly identify semantic entities and to align the tokens that represent them. My methods presented in this thesis use these attention vectors to align the source tokens in the sentence to be translated with the target tokens in the final translation output. Then, supplied terminology is injected, where these alignments correctly identify semantic entities.

My methods demonstrate significant improvement to the state-of-the-art results for NMT using terminology injection.

Keywords: attention-vector-based term injection, injection, machine translation, MT, neural machine translation, neural network, NMT, semantics, terminology

ACKNOWLEDGMENTS

I thank Parris Egbert for letting me use his computing resources and for his help and friendship for many years. I also thank Tony Martinez and David Wingate for invaluable instruction on machine learning and deep learning techniques, and Deryle Lonsdale for patience and expert guidance.

I am grateful for the support and input from friends Ryan Lee, who could be counted on at almost any hour to provide expert assistance and a good laugh, Dan Higinbotham for his excitement and encouragement about my ideas and work, and Steve Richardson for his expert knowledge in the field of machine translation.

Most importantly, I gratefully acknowledge the encouragement, support, patience, and sacrifice of my beautiful, capable wife and best friend, Jodi, and of our four wonderful and amazing sons, Caleb, Joshua, Benjamin, and Seth. Also the encouragement and support of my parents: my Dad, who did not get to see the completion of my work, and my Mother, who has been anxious from the beginning to receive a copy. And I thankfully acknowledge the encouragement and support of my wife's parents and siblings as well.

Table of Contents

List of Figures **vi**

List of Tables **vii**

1 Introduction **1**

2 Literature Review **3**

2.1 Terminology and NMT 6

2.2 Quality of Terminology 8

2.3 SMT and NMT 9

2.3.1 NMT Architecture 10

2.4 Semantic Space and the Target Vocabulary 14

3 Approach **18**

3.1 Corpora 18

3.2 The 4 Sequence Cases 19

3.3 Terminology Injection Methods 21

3.3.1 Terminology Injection Method 1 22

3.3.2 Terminology Injection Method 2 25

3.3.3 Method Description 27

4 Results and Evaluation **32**

4.1 Metrics 32

4.2 Why BLEU? 34

4.3 Application of Metrics 36
4.4 Interpretation of Metrics 36
4.4.1 Microsoft 37
4.5 Correlated Statistics 39
4.6 Injection Examples 41
4.7 Limitations of BLEU Scores 43

5 Conclusions and Future Work **46**

References **48**

List of Figures

2.1 A stacking recurrent architecture for translating a source sequence into a target sequence [22] 12

2.2 Global attention model [22] 14

2.3 Conceptual illustration of SEs in the MDSS 16

3.1 Predictions using original and substituted tokens 23

3.2 Prediction using sequence cases 24

3.3 Attention vectors in the rows and columns of the attention matrix 26

3.4 SE identification in a more complex sentence 26

4.1 BLEU evaluation scores compared with human evaluation scores [6] 35

4.2 Improvement charts for Microsoft datasets using terminology injection 38

4.3 Linear trends in Microsoft datasets using Multi-BLEU 39

4.4 Term statistics by corpus and metric 40

List of Tables

2.1 A sample dictionary lookup table 5

2.2 A sample of termbase entries 7

2.3 Polysemy as terminology 16

4.1 Improvement data for Microsoft datasets using terminology injection 37

4.2 Sample injections 42

4.3 Example approved terminology 44

Chapter 1

Introduction

Machine translation (MT) is a growing research area within natural language processing (NLP), 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 (PBMT). However, many in the field believe that "the current approach of statistical, phrase-based MT has kind of reached the end of its natural life."[1] With the recent resurgence of neural networks, some exciting advances in MT have been achieved over the last two to three years through the newest innovation, *neural machine translation* (NMT). Bahdanau et al. [2] explain that unlike the traditional phrase-based translation system, NMT uses a single, large neural network that reads a sentence and outputs a correct translation. Since the work done by Bahdanau et al., various kinds of neural networks have been used to create NMT systems. As a relatively new research area, much can be done to contribute to this emerging technology.

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. [12] make the point that enforcing "terminology. . . is a requisite, however, 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." To put this in perspective, SDL, a major translation services and technology solutions company, estimates

[1] Alan Packer, Engineering Director and head of the Language Technology team at Facebook in `https://slator.com/technology/facebook-says-statistical-machine-translation-has-reached-end-of-life/`

that a "medium-sized company" consisting of about 750 employees might translate up to 750,000 words per year. By correctly managing and applying terminology, SDL further estimates that the yearly cost savings to this company is approximately €60,600, or nearly $70,000.[2] That's significant, especially when extrapolated for larger organizations that translate millions of words into multiple languages annually.

However, comparatively little has been done to explore the use of vetted terminology as an input to MT for improved results. In fact, no established process for NMT yet exists to ensure that the translation of individual terms is consistent with an approved terminology collection. Not meaning to disparage the important work that has been done in this area over the years, the common lament, nevertheless, is that the "terminology problem" has not been solved for MT. Commenting on this, Hasler et al. [12] further state that controlling the translation output of NMT systems to adhere to user-provided terminology constraints, despite the impressive quality improvements of NMT, remains an open problem.

The greater focus in MT research has been on the general methodology of the core of the translation system. 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 results contain the approved terminology. A sentence translated using MT may be grammatically or linguistically correct, yet still may not use the terminology word choice mandated by the organization. Such a translation is, therefore, considered to be of lower quality than needed or desired [12].

In this thesis, I present two methods that I developed for injecting approved terminology into NMT that achieve improved translation results with regard to terminology selection over current state-of-the-art methods. The results from multiple datasets of varying sizes demonstrate significant gains, as high as 2.98 BLEU, the standard metric for evaluating the quality of machine-translated text (refer to section 4.1, Metrics), and more than 12% improvement in translation output from NMT.

[2]Infographic available at `https://www.sdl.com/software-and-services/translation-software/infographic/terminology-management-cost-savings.html`

Chapter 2

Literature Review

Despite the advances in MT technology and capability, most translations today are still produced by humans. This is commonly referred to as *human translation* (HT) to differentiate it from MT. MT alone, without some form of human intervention, still cannot produce the quality for every translation that is necessary in most scenarios, though it is used more and more frequently where the highest quality is not required. As a result, many variations of HT and MT combinations are in common use. *Machine-assisted human translation* (MAHT) describes a process where a human translator is assisted by various automatic tools. Perhaps the most important of these tools is the alignment tool that allows the words of a sentence to be matched with their equivalents in the translated sentence. *Human-assisted machine translation* (HAMT) can be thought of as the reverse of MAHT in that an automated translation system requires input from a human translator for certain tasks and to achieve the highest quality output. *Automated translation* (AT) is a process whereby existing, vetted translations are re-used through automatic matching algorithms to automatically produce a quality translation of a previously unseen corpus that contains the existing text portions, often through fuzzy matching. No matter the approach, the human yet remains at the center. Human intelligence and language capabilities are still unmatched for all but a few cases.

Since it first appeared in the late 1980s, SMT has dominated research and application of MT. Many advances in SMT over the years have enabled it to be of use in a variety of scenarios. Intensive research in many facets of SMT has delivered improvements such that

the performance of SMT systems for some language pairs approaches human translation capability.

Within the last two to three years, NMT has become increasingly important in MT research and use because of its ability to produce higher-quality and more-fluent translations than SMT [8, 17]. I present more information about the differences between SMT and NMT in section 2.3, SMT and NMT.

Despite producing overall better results, NMT shares some undesirable traits with SMT due to the nature of the similarities of their machine-learning and output models. A major undesirable similarity is that words or phrases not found in the training data cannot become part of a translation output that is dependent upon the training data alone. This problem is commonly referred to as the out-of-vocabulary (OOV) problem.

Similar to the OOV problem is the rare word (RW) problem. This problem is encountered when a word exists in the training data with sufficiently low frequency that the training model does not "learn" to use it. It is thus difficult to calculate meaningful and reliable probabilities for low frequency words, and impossible for absent words. Both the OOV and RW problems can result in either incorrect or poor translations, and represent a common problem for machine learning in general.

A few techniques have been developed to mitigate or resolve the OOV and RW problems. One of the most common, such as used by Luong et al. [23], is a dictionary lookup that relies on exact matches and string replacements, typically once the output translation has been generated. The dictionary lookup uses a simple aligned-values table that equates a pair of words or phrases between the source language and the target language. For example, Table 2.1 might be used to equate words for English (the source language) and Spanish (the target language).

Any time a source entry is found in an input sentence, the corresponding words in the output sentence are replaced by the target entry. If the source entry is an "unknown word"—that is, the MT system does not recognize the word—it may be passed through

Source	Target
house	casa
market	mercado
dog	perro
ice cream	helado

Table 2.1: A sample dictionary lookup table

untranslated to the output sentence. In this case, it is simply replaced by the target entry in the dictionary via string replacement to produce the final translation. For example,

(1) John eats ice cream.

(2) Juan come helado.

Regardless of what the MT system produces as the final translation, the source string "ice cream" in the source sentence is overridden by the target string in the target sentence, (2).

Another mechanism in use in SMT requires a more active approach to string replacement. A dictionary term is manually embedded in the source sentence before the sentence is processed by the system. This means that either a human must intervene to mark up the sentence before it can be translated, or some automatic pre-process must insert some markup of the source sentence using a dictionary, assuming the translation process can recognize and handle the markup. As an example, (4) is produced by (3).

(3) John eats [DICT: "ice cream" = "helado"].

(4) Juan come helado.

Although these mechanisms are commonly used in MT systems and can be beneficial, they are not ideal because they simply represent string or character replacement and have no direct association to the semantics and the abstracted meanings of the text.

The lack of an established methodology for terminology inclusion can lead to the mistranslation of terms, especially if they are infrequent (the rare word problem) or non-existent (the out-of-vocabulary problem) in the training data.

2.1 Terminology and NMT

Another problem similar to the OOV and RW problems is the terminology problem. Most organizations and disciplines increasingly use domain- or organization-specific words and phrases that have been carefully selected. These words and phrases comprise a terminology collection and are used and managed to ensure consistency and proper communication of important concepts and brand messaging [12].

Terminology is more than just words. A proper treatment of terminology recognizes that terminology is concerned with relationships—similarities and differences—between concepts, and the symbols that represent them. These symbols or tokens are what we often call *terms*. Terms have meanings, and semantics is the study of meanings and significance. This is a crucial point to understand for any measure of quality to be derived from the use of terminology.

In addition, terminology is an inseparable component of a domain of knowledge. That is, every specialized subject domain has its own terminology that represents the important and core concepts of that domain. Correct application of the terms in a domain support and enhance the communication and understanding of the knowledge inherent in the domain.

With regard to translation, the importance of terminology cannot be overstated. Effective translation relies upon the ability to faithfully communicate accurate concept equivalences, and not merely to piece together word associations. For this reason terminology is critically important, especially for MT. Not surprisingly, that is also why NMT is so effective—it focuses not on the mere representations afforded by the tokens, but on the semantic entities and the meanings behind the tokens. NMT and terminology are, therefore, of the same nature.

Although there may be many status indicators applied to individual terms for a variety of purposes within the practice of terminology management, terminology statuses typically fall into two primary categories: 1) Approved Terms where a semantic target equivalent of the source term is recognized as valid and accepted, and therefore can be used in a translation;

and 2) Rejected Terms, where an approved target replacement, if one exists, is used in the translation instead of an unacceptable (rejected) term used in either the source or the target.

For example, suppose that Organization XYZ has a newly established terminology management database, often referred to as a *termbase*, for standardizing and managing the organization's terminology. This termbase contains the equivalent term entries in both English (the source language) and Spanish (the target language), as shown in Table 2.2. Note that both of these terms are valid, but only one is approved for use in the organization.

Concept Definition	Src Term	Src Term Status	Tgt Term	Tgt Term Status
A big building where large	warehouse	Approved	almacén	Approved
amounts of goods are stored	storehouse	Rejected	depósito	Rejected

Table 2.2: A sample of termbase entries

Organization XYZ also has a large *translation memory* (TM), a database that stores *segments*, which are typically sentences or sentence-like units (headings, titles, or elements in a list) that have been translated previously. Each source and target pair of matching segments is known as a *translation unit* (TU). These TUs are precisely what is needed for training an MT system.

However, most of XYZ's currently approved terminology either has not been used historically and is not contained at all in any legacy texts or the TM that can be used for MT training, or it has been used only rarely and therefore appears in very few texts with very low frequency or in very few TUs in the TM. This poses a problem for an MT system: to produce quality translations without approved terminology or with only minimal application of approved terminology as training data.

Note that over time new translations that take advantage of the approved terminology will be added to the TM and will eventually become part of the standard training data for the MT system. This will ensure that approved terminology becomes an integral part of future translations.

In the meantime, however, Organization XYZ wants to ensure that any machine translation output contains the approved terminology (such as shown in row 1 of Table 2.2) and that this always happens, regardless of the existing training data.

The application of terminology management in MT has, at its core, two main operations: 1) validation: verifying that approved terminology is used in every case that is appropriate, and 2) substitution: *injecting* or replacing an approved term where it doesn't emerge in a translation.

2.2 Quality of Terminology

As mentioned previously, terminological data can play an important role in ensuring the quality of MT output. A standard definition of *data quality* is "the fitness of data for their intended purpose." If the terminological data used with the main terminology operations are of low quality—meaning that the terminology does not match the texts or the desired outcome for its use—then the results of utilizing terminology will suffer. What, then, determines the quality or fitness of terminological data for use in NMT? Consider the following factors.

- **Domains.** Multiple subject domains may be present in a terminology collection, though great care must be taken so as not to confuse concepts and the terms that represent them across domains. Where concept confusion exists, poor translations will ensue.
- **Collection size.** A terminology collection must be large enough to cover the significant concepts in the domain. Without proper coverage, the collection is less effective and the core domain concepts will not be communicated effectively. Conversely, a collection can be too large to be useful and may contain words that aren't terms at all. To qualify as a term, a single- or multi-word unit must have a specific meaning in a specific context in at least one specific domain. Therefore, general words—words that are used generally across all domains with the same meaning—are not terms and do not belong in a terminology collection.

- **Term length.** Terms of any length can be valuable, but if the terms in a collection are only short, single-word terms, the collection may not provide adequate coverage of the core domain concepts. Likewise, a collection containing only multi-word terms will have limited value. Longer terms are generally more precise from a conceptual standpoint and therefore more valuable. More about term length is discussed in subsection 3.3.2, Terminology Injection Method 2.
- **Term types.** Terminology can be classified into many types, including part of speech (typically nouns, verbs, adjectives, and adverbs), full forms and short forms (including abbreviations, acronyms, and initialisms), and so on. A useful terminology collection will have an appropriate assortment of various forms and types that satisfies the needs of the domains that it supports.
- **Duplicates.** True duplicates are a problem only in that they bloat a terminology collection with redundant data. However, apparent duplicates could instead be polysemes which can present a larger problem for MT. More about polysemes is presented in section 2.4, Semantic Space and the Target Vocabulary.

2.3 SMT and NMT

A *bitext* [11] is a type of *parallel corpus*, which contains one or more TUs created by a competent human translator. SMT, like NMT, uses Bayesian probability algorithms to train a model with bitexts that a computer can use to translate texts from one language to another. In its simplest form, this can be expressed as Equation 2.1.

$$p(y|x) \propto p(x|y)p(y) \tag{2.1}$$

However, the latest developments in SMT use *phrase tables* that store pre-identified phrases along with their associated usage or occurrence probabilities based on $p(y|x)$. These phrase tables are built or generated over time and often become larger than the training data

from which they are derived. Yet these phrase-based models rely entirely on word constructions and collocation patterns and not on semantics. SMT generally provides good quality when large and qualified corpora are available; however, translation output is inconsistent and unpredictable.

In contrast, an NMT system is a neural network that directly models the conditional probability $p(y|x)$ of translating a source sentence, $x_1, ..., x_n$, to a target sentence, $y_1, ..., y_m$ [22], producing a TU. The words of the source sentence are distilled into a vectorized semantic model, and only the model is stored. No phrase tables or probabilities are necessary. The TUs from the resulting bitext can be added to a TM, potentially with other TUs from unrelated bitexts.

2.3.1 NMT Architecture

Similar in some ways to an SMT system, an NMT system uses a two-step process, described as follows.

1. **Train.** The training portion of the process trains a model with what is included in the aligned source and target corpora. Naturally, the best way to incorporate terminology into a translation is to ensure that the terminology is included with sufficient frequency in the training data, and to allow the normal training process to work its mundane magic.

   Remember that if an organization has a TM (a collection of aligned bitexts), any translations in the TM can become part of the standard training data for the MT system. Ideally, the translations in the TM will also include approved terminology. This will ensure that approved terminology becomes an integral part of future MT-generated translations.

   However, the training portion of the process can be very expensive and time-consuming. Therefore, it may not be practical to frequently retrain an NMT model to include

the most up-to-date terminology in an active and growing or changing terminology collection. A more direct and less expensive way to include terminology that does not involve training may be more desirable.

Even though many modern MT systems can very quickly update a model with additional training data, it may take a long time before enough examples that contain approved terminology can positively affect the results of the model.

The approach that I contribute, and as discussed in this thesis, provides an advantage, allowing for an MT model of any domain to be augmented with both in-domain and out-of-domain terminology to produce desirable results. This can mean lower cost and shorter time to delivery than is necessary with retraining a model to include specific terminology.

2. **Translate.** Once training is done and a suitable model has been generated, translation can begin. Some important parts of the translate portion of the process relevant to this thesis include:

    - **Encoder**. Not surprisingly, the job of the Encoder is to "encode" the human-readable source sentence into a mathematical representation based upon the training model. This mathematical representation is known as the *context vector*, or s, as described by Luong et al. [22] and as expressed in Equation 2.2.
    - **Decoder**. The Decoder has the biggest job which is to "decode" the context vector from a 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. [22], the Decoder generates one target word (token) at a time and hence decomposes the conditional probability as Equation 2.2.

$$\log p(y|x) = \sum_{j=1}^{m} \log p(y_j|y_{<j}, s) \tag{2.2}$$

Pay close attention to the fact that only one target token is generated at a time, but each y is generated by looking at the y in any previous time step j (that is, $y_{<j}$). This backward glance allows for multi-token terms (a complete SE) to be generated.

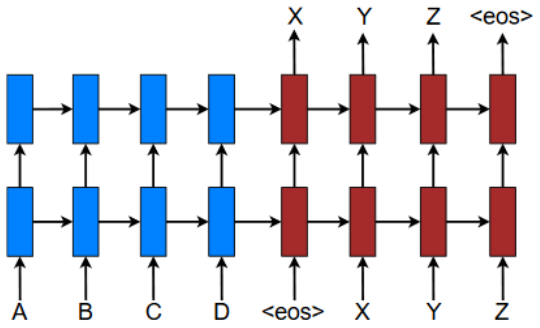


Figure 2.1: A stacking recurrent architecture for translating a source sequence into a target sequence [22]

Figure 2.1 [22] illustrates how an NMT model reads through all the source words until the end-of-sentence symbol `<eos>` is reached. It then starts emitting one target word at a time (depicted as X, Y, and Z).

- **Attention Mechanism**. The Attention Mechanism is used in the Decoder and allows the Decoder to "pay attention" to different parts of the source sentence that are more important at various stages of the translation rather than 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. [22] 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, c_t. In this model type, a variable-length

alignment vector a_t, whose size equals the number of time steps on the source side, is derived by comparing the current target hidden state h_t with each source hidden state $\bar{h}_s$, as expressed in Equation 2.3.

$$\begin{aligned} a_t(s) &= align(h_t,\ \bar{h}_s) \\ &= \frac{exp(score(h_t,\ \bar{h}_s))}{\sum_{s'} exp(score(h_t,\ \bar{h}_s))} \end{aligned} \tag{2.3}$$

The *score* function in Equation 2.3 is a dot product of the hidden states:

$$score(h_t, \bar{h}_s) = h_t^\top \bar{h}_s \tag{2.4}$$

Figure 2.2 [22] illustrates the global attention model with an attention layer detailed in Equations 2.3 and 2.4.

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 sentence based upon the semantic meaning in the vectors while generating the target sentence. Align weights are also known as *attention vectors*. Luong et al. [22] 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. [2] focuses on alignment as a way to improve translation. This thesis builds on that work as another step forward to improve NMT.

Many in the field have recognized that the word alignments afforded by NMT are not perfect. For example, NMT challenge #5 cited by Koehn and Knowles [17] 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. Koehn

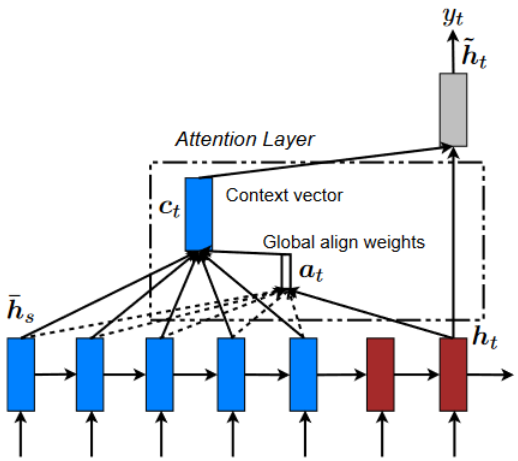


Figure 2.2: Global attention model [22]

and Knowles 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.

Like Koehn and Knowles, I have noticed a few cases in this research where the alignments generated by the NMT system were less than desirable. Nevertheless, this work demonstrates that the alignment function does indeed perform sufficiently well to be useful, not only for its intended purpose but also to accomplish the goals of this research. Any improvement to the general NMT alignment process will only serve to boost the effectiveness and value of this research as well.

2.4 Semantic Space and the Target Vocabulary

The result of an NMT training step is the creation of two important structures:

1. An n-dimensional model that encapsulates the semantic space of the training data. This can be referred to as the *multi-dimensional semantic space* (MDSS). Vector space models (VSMs) represent—or *embed*—words in a continuous vector space where semantically similar words are mapped to nearby points; that is, they are "embedded" in the VSM

near to each other. The MDSS, therefore, is commonly referred to as *word embeddings*. VSMs have played an important part in NLP for a long time, and all implementations depend to some degree on the Distributional Hypothesis, which states that "words that occur in the same contexts tend to have similar meanings" and are therefore semantically related [26].

2. 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. In fact, Jean et al. [14] 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. Denkowski and Neubig [8] indicate that this limit is in the range of 30K–50K words, whereas Jean et al. [14] indicate 30K–80K words. The OpenNMT default is 50K words [15].

When a token is encountered in the training data during training, its *semantic entity* (SE) is represented by a vector that corresponds to a location in the MDSS. This SE is the meaning that is represented by the token sequence. Throughout the training step, each subsequent occurrence of a token or recurring token sequence in a sentence context—and therefore its associated SE—refines its vectorized location in the MDSS.

What a human recognizes as a single SE may be different than what an NMT system recognizes as an SE. For example, an English-speaking human easily recognizes that the separate tokens "ice" and "cream" when taken together, and in that order, form a single SE that represents a delightful and cold dairy treat—"ice cream"—even though these tokens, when taken separately, represent two distinct SEs with different meanings. Although an NMT system can learn to associate these two tokens to form a single SE, it still treats them individually and must deal with the single-token vocabulary model.

Note also that a single token can represent more than one concept, such as listed in Table 2.3. These are known as *polysemes*: they are spelled the same but have different, though related meanings.

	Token	Meaning
1.	church	A building set apart for public worship.
2.	church	A group of worshippers; a congregation.
3.	church	A religious institution of worship.

Table 2.3: Polysemy as terminology

The dictionary lookup model as described previously, which merely looks at characters in a string and does not take the SE into account, is insufficient in the face of polysemy because a single token can represent more than one meaning.

If it were the tokens themselves being mapped to the MDSS, then each of these meanings would be obscured by and merged into a single location and become indistinguishable because the tokens are the same. But because it is the meaning, or the SE, of the token and not the token itself that is mapped to the MDSS, these SEs occupy distinct locations in the MDSS. Consider Figure 2.3.

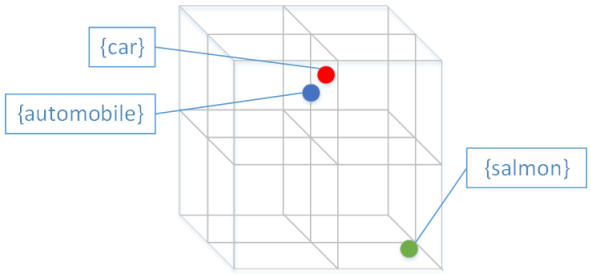


Figure 2.3: Conceptual illustration of SEs in the MDSS

As stated previously, SEs that are related or similar to each other are located in proximity to each other in the MDSS. Therefore, the SEs of the tokens "car" and "automobile" are mapped to the MDSS in relative proximity, whereas the SE for the token "salmon" is mapped farther away.

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. It

is a mathematical Interlingua. 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*.

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

Chapter 3

Approach

For my research, I chose English and Spanish as the source and target languages, respectively. Any two languages would work for my research, but I chose Spanish as the target language because of my experience and familiarity with that language, and because of the availability of resources germane to my research.

3.1 Corpora

For my research, I used two main corpora for training and testing the NMT model:

1. **EuroParl.** This bitext corpus is available in several language pairs and contains 2 million sentences in each language (for a total of 4 million aligned sentences per language pair). The corpus contains transcriptions of the proceedings of the European Parliament. The domain is government and politics. I selected this corpus because it is commonly used for MT evaluations.[1] I used the first 1 million sentences in both source and target as training data to train the NMT system and to produce the model.
2. **Microsoft C# technical documentation.** I obtained this corpus by scraping the Microsoft Developer Network (MSDN) website.[2] The nature of the MSDN website is that it contains pre-aligned bitexts for a variety of language pairs (E to X), if a display language other than English is selected. This means that obtaining large amounts of high-quality, aligned bitext data is relatively simple and fast with the right tools. The resulting corpus is comprised of 10,000 aligned sentences.

[1] Downloadable at `http://www.statmt.org/europarl/`
[2] Located at `https://docs.microsoft.com/en-us/dotnet/csharp/`

I selected this corpus primarily because it is of a very different domain than the EuroParl corpus used for training the NMT model. Additionally, Microsoft has a large publicly-available terminology collection which the MSDN documentation utilizes heavily, making evaluation of terminology-related efforts ideal.

My purpose in selecting this corpus was to evaluate the effect of terminology injection on NMT output from an 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 [17] list raising the quality of out-of-domain translations as the first of "Six Challenges for Neural Machine Translation."

3. **Microsoft terminology.** I also used an existing terminology collection available from Microsoft. I obtained the Microsoft terminology by downloading a publicly-available TermBase eXchange (TBX) file directly from the Microsoft Language Portal.[3] TBX is an international standard primarily for archiving information in a termbase or exchanging this information between systems. This TBX file contains both English and Spanish equivalent terms in approximately 30,000 concept entries.

3.2 The 4 Sequence Cases

NMT systems operate at the token level for input and output even though internally they work with vectorized SEs. Therefore, whenever a replacement is to be made between equivalent terms in a target sentence, the final tokens must be taken into consideration. A single SE can be represented by more than one token in a sentence; therefore, we can refer to the set of tokens for a single SE as a *sequence*.

Sequence pairs come in four varieties or cases, where each sequence case represents an SE equivalence between two terminology options even though the number of tokens that represent each SE may not be equivalent between the terms. I present the four sequence cases as part of my contribution.

[3]Downloadable at `https://www.microsoft.com/en-us/language`

1:1. A single token term is replaced by a corresponding single token term. For example, assume that the word "store" is a preferred term that should replace the term "market."

(5) John went to the **market**.
(6) John went to the **store**.

1:many. A single token term is replaced by a corresponding multiple-token term. For example, assume that the word "seat restraint" is a preferred term that should replace the term "seatbelt."

(7) John used his **seatbelt**.
(8) John used his **seat restraint**.

many:1. A multiple-token sequence term is replaced by a corresponding single token term. For example, assume that the word "histogram" is a preferred term that should replace the term "bar chart."

(9) John created a **bar chart**.
(10) John created a **histogram**.

many:many. A multiple-token sequence in the source is replaced by a multiple-token sequence in the target. For example, assume that the technical term "myocardial infarction" is a preferred term that should replace the more common but less-precise term "heart attack."

(11) John suffered a **heart attack**.
(12) John suffered a **myocardial infarction**.

In the previous example pair, sentences (11) and (12), the number of tokens in the source equals the number of tokens in the target, though this may not always be the case. Consider an additional example where "Cartesian coordinate system" is a preferred term (comprised of three tokens) that should replace the term "system of rectangular coordinates" (comprised of four tokens):

(13) John used the **system of rectangular coordinates**.

(14) John used the **Cartesian coordinate system**.

These sequence varieties become particularly important when dealing with terminology inclusion methods in NMT.

3.3 Terminology Injection Methods

This paper describes two methods that I developed for including terminology in NMT, and some suggestions for further research that may enhance translation quality using terminology inclusion.

The OpenNMT PyTorch system is written in the Python language, version 3.6, and utilizes PyTorch, an open source deep learning platform built for Python; therefore, I wrote all of my code for both methods in Python 3.6 to match the OpenNMT PyTorch codebase.

Although each of the two methods can be used independently of the other, Method 2 builds on the concepts of Method 1. Therefore, it is important to understand Method 1 to fully appreciate Method 2. Regardless of the method used, the first step in generating the target sentence in an NMT Decoder is decoding the context vector into a target sentence vector. No human-readable tokens are used. The Decoder uses the trained model and the MDSS to generate a target sentence with the correct syntax and grammar of the target language, which typically differs from that of the source language. Once the target sentence vector has been generated by the Decoder, the 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 SEs in the target sentence vector.

To begin my evaluations, I first needed to obtain baseline data. I accomplished this by training the OpenNMT PyTorch system [15] using a bitext of one million aligned sentences from the EuroParl training corpus for 10 *epochs*. An epoch is a single pass through the entire training set while training the machine learning model. The number of epochs used in machine learning is something of an art, and is usually determined by "what works" or "what works best," which is open to interpretation. The number of epochs needed varies from

application to application. The default for the OpenNMT PyTorch system is 13 (using the training command line option `-epochs`).

Despite training for only 10 epochs, which took nearly two weeks to complete with 1 million sentences, the resulting model, consisting of a 2-layer Long Short-Term Memory (LSTM) network with 500 hidden units on both the encoder and decoder, produced high-quality predictions, even better than I anticipated. The off-the-shelf OpenNMT PyTorch system [15] performed very well using all other default parameters.[4]

3.3.1 Terminology Injection Method 1

The tokens in the target token vocabulary are human-readable symbols that represent the SEs in the target sentence vector. The target token vocabulary is a collection of symbols (tokens) found in the training corpus. Of crucial importance is the mapping between the MDSS and the tokens in the vocabulary. Therefore, it is possible to modify the vocabulary and replace a token with a different token. A token cannot be removed from the vocabulary, but it can be modified. Doing so will allow replacement of a term.

For example, the source sentence given in (15) can be fed into the Encoder for translation. This might result in the viable Spanish translation given in (16). Figure 3.1a shows the source sentence tokens and the generated target sentence (the prediction) along with the attention vectors.

(15) Your report is absolutely disgraceful.

(16) Su informe es absolutamente vergonzoso.

(17) Su exposición es absolutamente vergonzoso.

However, the target sentence in (17) is produced from (15) if the token "informe" is replaced in the target token vocabulary with the token "exposición". Figure 3.1b shows the generated prediction with the substituted term. Angle brackets in the matrix indicate a manual terminology substitution.

[4]Parameter options available at `http://opennmt.net/OpenNMT-py/options/train.html`

	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) Prediction using original tokens from the target token vocabulary

	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) Prediction using a substituted token from a terminology collection

Figure 3.1: Predictions using original and substituted tokens

Note also that all values in the attention vectors are exactly the same in both tables even though the second target token in the prediction in Figure 3.1b (highlighted in red) has changed. This demonstrates that the translation is not dependent upon a specific target token. Instead, it is dependent upon the SE that the target token represents and the location (index) in the target vocabulary to which the SE is mapped.

Therefore, I provide a method to inject terminology into the normal Decoder function by a careful and simple modification of the target vocabulary. Method 1 provides the capability to significantly improve NMT translation, and offers an improvement to the state of the art. However, there is at least one negative consequence and one limitation to this method.

The negative consequence centers around the fact that the target vocabulary is a collection of *unique tokens*. As indicated in Table 2.3, the polysemy problem arises when a single token represents more than one SE. As a result, if the replacement token in the target vocabulary does not also function as a suitable substitute for all of the SEs in the MDSS represented by the original target token, the token replacement may produce a poor or incorrect translation. Therefore, vocabulary token replacement, if used, must be used judiciously.

The limitation of this method centers around the fact that the target vocabulary is a collection of *single tokens*. Therefore, sequence cases of 1:1 and 1:many, which both replace

only a single token, can be handled; but sequence cases of many:1 and many:many, which replace more than one token, cannot.

	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

(a) Prediction using sequence case A

	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

(b) Prediction using sequence case B

Figure 3.2: Prediction using sequence cases

1:many replacement can be handled using this method because a single token (term) can be replaced with anything we choose, including much longer strings or token sequences. For example, looking at sentences (15) and (16), if we want to replace the token "informe" in the target vocabulary with the long string "exposición de alta calidad", (18) is produced:

(18) Su **exposición de alta calidad** es absolutamente vergonzoso.

Comparing Figures 3.2a and 3.2b, we see that once again the values in the attention vectors are identical to the previous token substitutions. This works because the token in the target vocabulary is just a string literal. The semantic mapping happens with the vectors in the MDSS and not with the vocabulary tokens. Therefore, the vocabulary tokens can be anything, and preferably something that will produce a useful result.

Nevertheless, this can cause problems with proper linguistic agreement. 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" fixes the problem. A replacement token or sequence must be in agreement with the original token, or the output may need to be post-edited.

3.3.2 Terminology Injection Method 2

With this method, I introduce a more robust and rigorous solution than Method 1 to inject terminology into a translation. This method does not suffer from the same negative consequence or limitation as does Method 1; however, it builds on the core idea in Method 1 of token replacement based upon SE identification. As mentioned previously, Method 1 can only replace single tokens. That creates a problem when trying to translate "ice cream" or some other sequence of multiple tokens. However, NMT already has this problem solved. It uses the attention mechanism prevalent in state-of-the-art NMT systems to produce the desired results. Attention vectors play an important part of this method. Therefore, I refer to this method as *attention-vector-based term injection*. I provide an explanation of the importance of attention vectors and their function, which form the basis for the inspiration for Method 2.

To understand how the attention vectors work, we need to consider the attention matrix. Naturally, the attention matrix can be represented as a collection of rows and columns. Each row of the attention matrix pertains to a single token in the generated target sentence. Each column of the attention matrix pertains to a single token in the source sentence and contains a set of probability values that represent the likelihood of an association between the source token and a token in the target sentence. The association between source and target tokens is known as an *alignment*.

In both Figures 3.1b and 3.3a, the highest value in the column under the source token "report," a value of 0.8274, indicates 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 3.3a and shows the relationship between each source token and its corresponding target token. In other words, it indicates an alignment.

Inspecting the columns of the attention matrix is useful, but to get a complete understanding of how the attention vectors work, we must also focus on the rows of the

	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 3.3: Attention vectors in the rows and columns of the attention matrix

attention matrix. Note that in the first two columns of Figure 3.3b, 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 the many:1 sequence case between source and target tokens.

The attention vectors in the matrix identify the SEs in the matrix. This allows the NMT system to correctly produce a translation sequence of “Debemos” from “We must” or of “helado” from “ice cream”. Resultantly, the ability to correctly identify an SE and its associated tokens is necessary to effectively handle terminology in an NMT system, and is the very capability that makes Method 2 a possibility with some additional processing.

	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 3.4: SE identification in a more complex sentence

To this point, all previous examples have been relatively simple. To demonstrate that the principle still applies to more complex sentences, Figure 3.4 provides a relevant example. Points of interest in the example in Figure 3.4 include:

- The same many:1 token construction at the beginning of the sentence in Figure 3.4 (columns C0 and C1) as we saw in Figure 3.3b.
- The complexity of grammatical structure differences between source and target languages is apparent. In this case with English as the source language and Spanish as the target language, the order of the tokens between source and target languages in the English term "European Works Councils" is reversed.
- The target translation of the source term "European Works Councils" has experienced *text expansion*—that is, there are 3 tokens in the source term and 4 tokens in the target term. This also represents an example of the many:many sequence case between source and target tokens.

3.3.3 Method Description

I outline the process I developed to correctly identify and apply appropriate terminology that builds on the concepts of Method 1.

1. Scan the source sentence for all occurrences of any term from the terminology collection and add the identified terms to a "term candidates" collection.
2. Iterate through the term candidates collection and tag the term candidates in the source sentence. For example, the term "report" is *tagged* as follows:

   (19) Your {report} is absolutely disgraceful.

   Accurate tagging is non-trivial and critical to the success of the method for producing the best results. Note also that this is only one method for tagging source terms and was selected for its obvious visual-tracking benefits for evaluation purposes; other methods may be devised for identifying and tracking (tagging) source terms for target substitution. (It is assumed that a production system would not utilize a visual tagging method, except perhaps in a "verbose" mode.) However, whatever method is employed, adherence to the following rules is necessary:

- Evaluate the term candidates in order of length, longest to shortest. This will typically reduce or eliminate ambiguity among possible term candidates and deliver the best results. Consider the following sentence:

  (20) The committee reviewed your finance report.

  The terminology collection may contain the terms "finance," "report," and "finance report." All three of these terms would be added to the term candidates collection because they can all be found in the sentence. However, only the term "finance report" should be tagged in the sentence, as indicated in (21). In this case, the term candidates "finance" and "report" are discarded because they do not figure anywhere else in the sentence.

  (21) The committee reviewed your {finance report}.

- Do not tag substrings or overlapped terms. Tagging substrings or overlapped terms interrupts the relationship between the SE and the token sequence that represents the SE. Therefore, term candidates that appear within other term candidates should be ignored. For example, separately tagging the substring "report" in sentence (22) will produce undesirable results:

  (22) The committee reviewed your {finance {report}}.

  Likewise, tagging the overlapping terms "Cartesian coordinate" and "coordinate system" will produce poor and jumbled results:

  (23) Mary used the {Cartesian {coordinate} system}.

- Tag every separate occurrence of a term in the sentence that is not a substring of another term.
- Tag any occurrence of a term only once.

3. Generate the attention matrix using the attention vectors from the NMT Decoder.

4. 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. For example, refer to Figure 3.1.

5. 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. Note that the first and last tokens in a single-token term both refer to the same token.

   The set of probability values, P, for a given column in the attention matrix, C, can be represented by Equation 3.1 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 3.2, where n is the total number of columns in 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 3.3. Note that the $argmax$ function returns the *index* of the max element of the specified array, and not the max element itself.

$$P_C = \{p_0, p_1, p_2, \cdots, p_r\} \tag{3.1}$$
$$M = \{P_0, P_1, P_2, \cdots, P_n\} \tag{3.2}$$
$$R_{S_i} = argmax(P_{C_i}) \tag{3.3}$$

   For example, consider Figure 3.3a. The row in M that corresponds to the first token of the target term, T_1, is the row that corresponds to the highest probability value in the column of the first token of the source term, S_1. The values from Figure 3.3a provide the result:

$$P_1 = \{0.2725, 0.8274, 0.0297, 0.0126, 0.0027, 0.0148\}$$
$$R_{S_1} = argmax(P_1) = 1$$

   Therefore, the single-token source term "report" in column 1 is correlated to the single-token target term "informe" in row 1 ($R_{S_1} = 1$). More importantly, the SE in the source

sentence is correlated with the SE in the target sentence. This makes substitution of the target word with approved terminology simple and precise.

Figure 3.4 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 SE 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 SE and E represents the ending token in the SE:

$$B = min(R_{S_{first}}, R_{S_{last}}) \tag{3.4}$$
$$E = max(R_{S_{first}}, R_{S_{last}}) \tag{3.5}$$

Plugging in the values from Figure 3.4, as an example, we get:

$$P_4 = \{0.0336, 0.0251, 0.0565, 0.0230, 0.1637, 0.0057, 0.7237, 0.0678\}$$
$$P_6 = \{0.0215, 0.0451, 0.3289, 0.6981, 0.2936, 0.0612, 0.0502, 0.0277\}$$
$$R_{S_4} = argmax(P_4) = 6$$
$$R_{S_6} = argmax(P_6) = 3$$
$$B = min(6, 3) = 3$$
$$E = max(6, 3) = 6$$

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

This works well for left-to-right (LTR) languages such as English and Spanish. For right-to-left (RTL) languages, such as Arabic or Hebrew, the begin (B) and end (E) functions would need to be reversed.

6. Look up the source term in the terminology collection and retrieve the target equivalent. Bear in mind that polysemes can cause problems with target term selection. More about polysemes is presented in section 2.4, Semantic Space and the Target Vocabulary.

7. Substitute the original target tokens in the NMT prediction with the tokens of the equivalent target term.

Following this process will inject supplied terms into the NMT Decoder output by matching the SEs in the attention matrix.

Chapter 4

Results and Evaluation

The evaluation information in this chapter only applies to Method 2, a complete terminology injection method that handles all four sequence cases. Although I provide both Method 1 and Method 2 as contributions to the state of the art for NMT, Method 1 is a springboard to Method 2 and as such I do not provide an evaluation of Method 1.

4.1 Metrics

A variety of metrics for evaluation of MT systems and output are available. Among these are the following:

- **Bilingual Evaluation Understudy (BLEU)[27].** BLEU has become the standard algorithm for evaluating the quality of machine-translated text. Of the several algorithms for measuring MT quality (only some are listed here), BLEU remains the *de facto* standard. The scale is 0 to 1, with 1 being best, though in practice the scale is often shifted to 0 to 100 to represent a percentage (that is, a BLEU score of 0.3468 is often represented as 34.68%, or just 34.68).

  Several implementations of the BLEU algorithm are available and each may give slightly different results. Multi-BLEU is a commonly used implementation as is also the NIST-BLEU implementation. Consequently, I used both of these BLEU implementations for my evaluations.

- **National Institute of Standards and Technology (NIST)[9].** NIST is based on the BLEU metric, but with some alterations. Where BLEU simply calculates n-gram

precision adding equal weight to each one, NIST also calculates how informative a particular n-gram is: More weight is given to rarer n-grams. The scale is 0 to 10, with 10 being best. Note that the NIST metric is not the same as the NIST-BLEU metric.

- **Recall-Oriented Understudy for Gisting Evaluation (ROUGE)[20].** Six variations of this metric exist:
  - **ROUGE-N:** Overlap of n-grams between the system and reference summaries.
    - **ROUGE-1** refers to the overlap of unigrams (each word) between the system and reference summaries.
    - **ROUGE-2** refers to the overlap of bigrams between the system and reference summaries.
  - **ROUGE-L:** Longest Common Subsequence (LCS)-based statistics. The longest common subsequence problem naturally takes into account sentence-level structure similarity and automatically identifies longest n-grams co-occurring in a sequence.
  - **ROUGE-W:** Weighted LCS-based statistics that favor consecutive LCSs.
  - **ROUGE-S:** Skip-bigram-based co-occurrence statistics. A skip-bigram is any pair of words in their sentence order.
  - **ROUGE-SU:** Skip-bigram plus unigram-based co-occurrence statistics.
- **F-Measure.** The harmonic average of precision and recall. The scale is 0 to 1, with 1 being best.
- **Word Error Rate (WER).** Derived from the Levenshtein distance, which is a metric for measuring the difference between two sequences. More precisely, the Levenshtein distance between two words is the minimum number of single-character edits (insertions, deletions, or substitutions) required to change one word into the other. This is also known as *edit distance*. This metric is used more often in speech recognition than MT.

4.2 Why BLEU?

BLEU is calculated as follows, according to Papineni et al. [27]: Given that c is the length of the candidate translation (prediction) and r the length of the reference translation, a brevity penalty, BP, is calculated using the formula in Equation 4.1.

$$BP = \begin{cases} 1, & \text{if } c > r \\ e^{(1-r/c)}, & \text{if } c \leq r \end{cases} \tag{4.1}$$

The geometric average of modified n-gram precisions, p_n, is calculated using n-grams up to length N (the standard values for N are 1 to 4 inclusive) and positive weights, w_n, that sum to one. BLEU is then calculated according to Papineni et al. [27] using the final formula given in Equation 4.2.

$$BLEU = BP \cdot exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{4.2}$$

Papineni et al. [27] explain that few translations will attain a score of 1 (the highest possible score) unless they are identical to a reference translation. For this reason, even a human translator will not necessarily score 1. Translation is not deterministic—given the grammar of the language and the richness of the vocabulary for the topic, there could be many possible correct translations for a given source. Indeed, Papineni et al. provide an example of this point where a human translator scored only 0.3468 against four references on a test corpus of about 500 sentences.

To put this in perspective, Koehn [16] notes that the test sentences used in his research all received BLEU scores in the range of 0.21 to 0.37. The data reported by Coughlin [6], and shown in Figure 4.1, show that most of the BLEU scores in her research also fall within that same general range or perhaps slightly lower, with a few outliers receiving higher scores. Likewise, Bahdanau et al. [2] report BLEU scores in the range of 0.1393 to 0.3615 for their evaluations of various experimental NMT methods. Of course, many factors play a part in

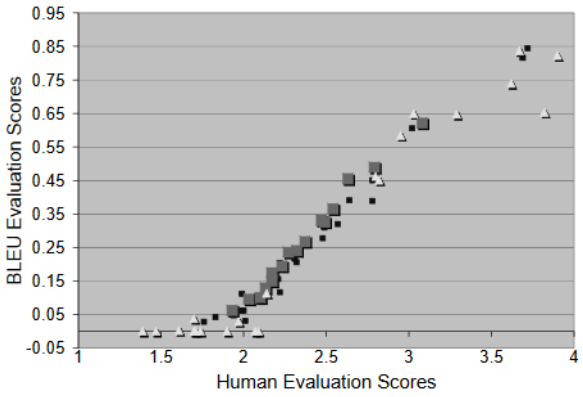


Figure 4.1: BLEU evaluation scores compared with human evaluation scores [6]

the calculation of BLEU scores, such as unknown words (the OOV problem), sentence length, the degree of syntactic and semantic divergence, and how the input and reference translation were generated—and both may be translations from a third language [16].

Any increase in a BLEU score of a full point (1.0, or a full percentage point) or more from a previous scoring of the same corpus is generally considered by MT practitioners to be a good rule-of-thumb to indicate a significant and statistically-relevant improvement. Indeed, according to Koehn [16], the results are within the 95% statistical significance range if the BLEU score difference is at least 2–3% for a test set size of 300 sentences or more. This "bootstrap resampling method" validates the common BLEU score improvement rule-of-thumb. For examples, see section 4.4, Interpretation of Metrics.

Automated metrics such as the BLEU score can be very good as quality indicators, where "quality," such as in this case, can be defined as the similarity between the output of an MT system and what a professional human translator can produce. Therefore, BLEU is generally considered to correlate well with human judgment [6, 16, 24, 27]. Indeed, Koehn [16] observes that an improvement in BLEU is taken as evidence for improvement in translation quality since it has been shown that the BLEU score correlates with human judgment. Koehn further observes that while this may not always be the case, it is generally accepted in practice.

4.3 Application of Metrics

Once the OpenNMT system was trained and a model generated from the EuroParl training corpus, I needed to generate a baseline. To obtain this baseline, I translated a dataset from the test corpus without using terminology injection methods. Refer to section 3.1, Corpora. I then compared the output predictions against the corresponding reference translations from the test corpus. The baseline is a score using the standard BLEU algorithm for evaluating the quality of MT output. For a description of the BLEU metric, see section 4.7, Limitations of BLEU Scores.

Once the baseline for a given dataset had been determined, I translated the same dataset again using terminology injection. As before, the output predictions were compared against the same corresponding reference translations from the test corpus and a new BLEU score calculated.

Finally, I compared the BLEU scores of the output without terminology injection (the baseline) to the output with terminology injection, and calculated an improvement score using Equation 4.3.

4.4 Interpretation of Metrics

My results demonstrate a significant improvement to NMT output using terminology injection Method 2 based upon the significance calculations provided by Koehn [16]. To validate improvement in the BLEU scores, I used the standard formula for calculating an improvement percentage as provided in Equation 4.3, where x is the baseline BLEU score and y is the BLEU score with terminology injection.

$$I = 100\left(\frac{y-x}{x}\right) \tag{4.3}$$

I give an example in Equation 4.4 showing that the values provided fall squarely within the 2–3% minimum range necessary to satisfy the 95% statistical significance target cited by

Koehn [16]. Along with this example, refer also to Figure 4.2 and Table 4.1. The baseline and terminology injection BLEU scores for this example are supplied by the data in Figures 4.2a and 4.2b, and the resultant improvement score is listed in Table 4.1.

$$12.41 = 100\left(\frac{26.72 - 23.77}{23.77}\right) \tag{4.4}$$

Based upon this information, all of the results from Method 2 presented here show statistically significant improvements. Note once again that, according to Koehn [16], the results are within the 95% statistical significance range "if the true BLEU score difference is at least 2–3%" for a test set size of 300 sentences or more. This demonstrates that Method 2 provides the capability to significantly improve NMT translation, and offers an improvement to the state of the art.

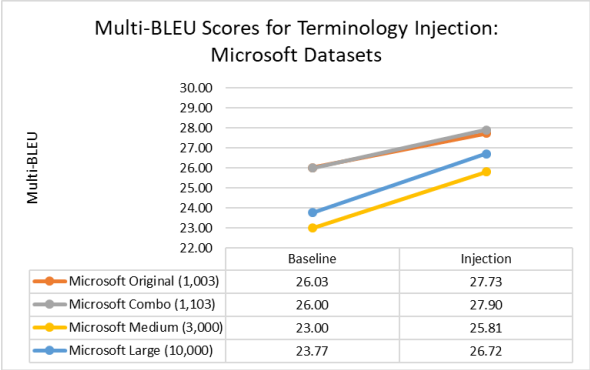
4.4.1 Microsoft

Figure 4.2 shows significant improvement in the translations of the Microsoft datasets over the baseline using the Multi-BLEU, NIST-BLEU, and NIST scoring methods. Exact improvements are listed in Table 4.1 for the same three scoring methods. This is particularly interesting because the Microsoft dataset results are out-of-domain results. These data clearly show that there is value in using this method to achieve improved translation results with off-the-shelf NMT, even on out-of-domain corpora. Note in Figure 4.2c that NIST awards the highest scores to the largest datasets.

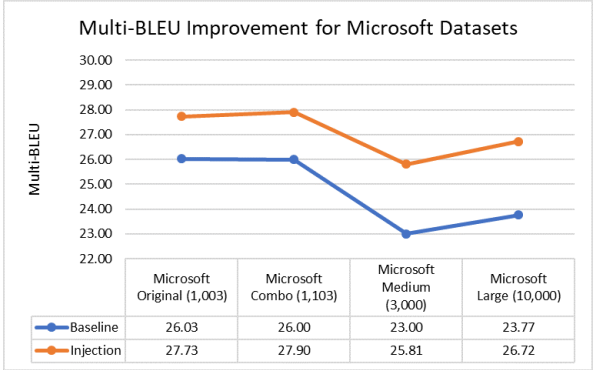
Dataset	**Sentences**	**Multi-BLEU**		**NIST-BLEU**		**NIST**	
		Result	**Improvement**	**Result**	**Improvement**	**Result**	**Improvement**
Microsoft Original	1,003	**+1.70**	6.53%	**+1.77**	6.73%	**+0.42**	6.65%
Microsoft Combo	1,103	**+1.90**	7.31%	**+2.31**	8.78%	**+0.50**	7.95%
Microsoft Medium	3,000	**+2.81**	12.22%	**+2.85**	12.16%	**+0.65**	10.12%
Microsoft Large	10,000	**+2.95**	12.41%	**+2.98**	12.22%	**+0.70**	10.26%

Table 4.1: Improvement data for Microsoft datasets using terminology injection

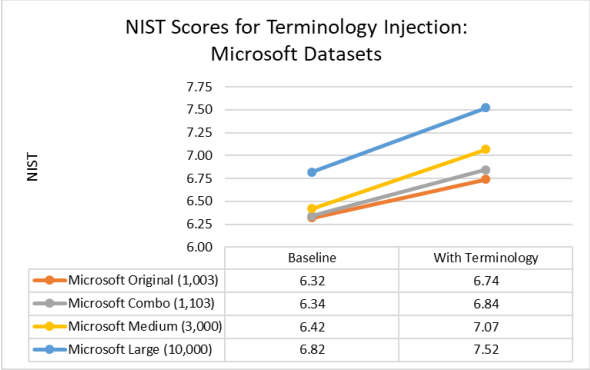
Figure 4.3 shows a correlation in the increase in percentage improvement with the increase in the size of the corpus that is quite interesting. That is, as the number of sentences



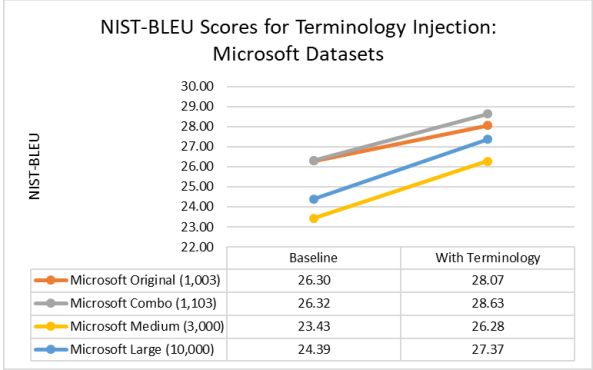
(a) Evaluation results for Microsoft datasets using Multi-BLEU



(b) Improvement results for Microsoft datasets using Multi-BLEU



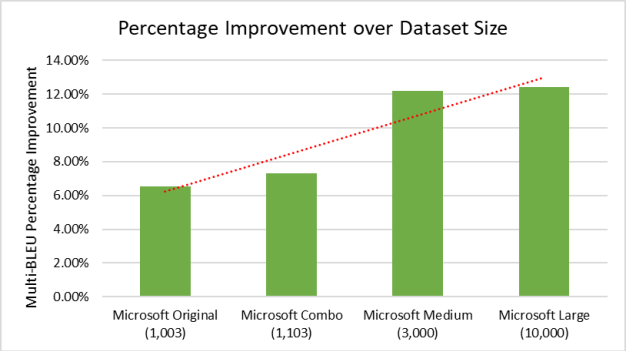
(c) Evaluation results for Microsoft datasets using NIST



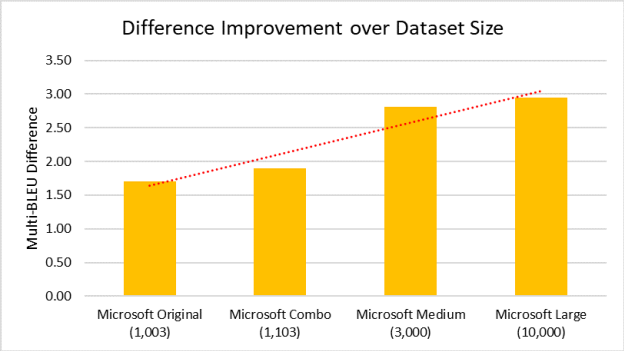
(d) Evaluation results for Microsoft datasets using NIST-BLEU

Figure 4.2: Improvement charts for Microsoft datasets using terminology injection

in the dataset increased, the improvement percentage also increased. These results are very different from the expectation of an improvement score that remains uniform and average regardless of the size of the dataset. For example, one might assume a roughly 3% increase across the board for all datasets, regardless of the dataset size. These data seem to tell a different story. According to these limited data, it would appear that a larger dataset will result, on average, in a larger improvement. However, more evaluations will be necessary to validate that assertion or to determine in what circumstances it may apply. Still, these data support the finding that Method 2 improves NMT output by a significant margin [16].



(a) Linear trend percentage improvement with increasing dataset size



(b) Linear trend difference improvement with increasing dataset size

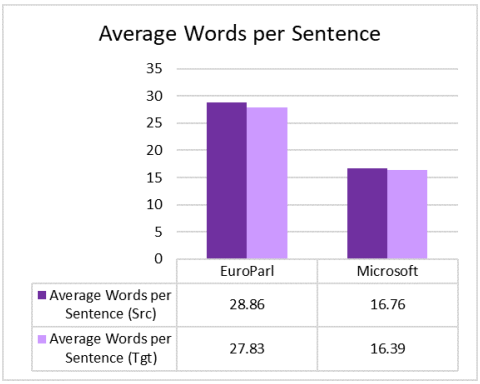
Figure 4.3: Linear trends in Microsoft datasets using Multi-BLEU

4.5 Correlated Statistics

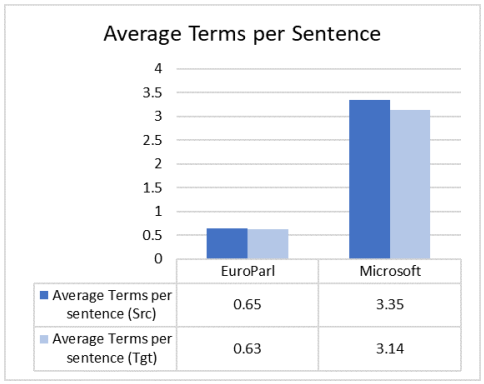
As mentioned previously, I selected the Microsoft corpus because I expected it to be different from the EuroParl corpus. I present here some empirical data to corroborate that expectation.

Figure 4.4a indicates that the EuroParl and Microsoft corpora are indeed quite different with regard to the average number of words per sentence in both source and target. The Microsoft corpus has nearly half the number of words per sentence than the EuroParl corpus. Similarly, Figure 4.4b shows the differences of term saturation between the two corpora. On average, a sentence in the Microsoft corpus contains about three terms, whereas a sentence in the EuroParl corpus contains, on average, less than one. The number of terms per sentence is quite higher in the Microsoft corpus than in the EuroParl corpus. Compare this to Figure 4.4c that shows the average percentage of sentences that contain terms in the Microsoft corpus is nearly twice that of the EuroParl corpus. Note that the number of sentences with identified terms is the same for both source and target; that is, for each term in the source sentence there will be an equivalent term in the corresponding target sentence. Finally, Figure 4.4d indicates the average gram size of injected terms for each corpus for both source and target. A single term is an n-gram, where n is the number of tokens in the term. For example, the term "ice cream" is a 2-gram, also known as a *bigram*.

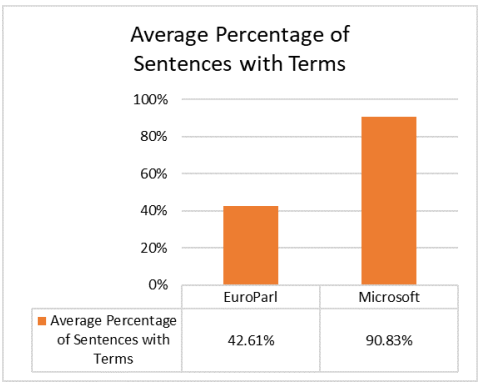
To put this in perspective, the EuroParl corpus has a large number of words per sentence, on average, but less than half of those sentences contain terms. This will affect the yield of terminology injection. Furthermore, the number of terms per sentence is very low, although when a term exists in a sentence it is usually a multi-word term (an n-gram, where n is 2 or greater). Now compare that to the Microsoft corpus which has shorter sentences (fewer words per sentence, almost half the length of EuroParl sentences), but almost every sentence contains one or more terms. Additionally, the number of terms per sentence is very high (more than five times greater than that of EuroParl, on average), though terms are generally shorter (typically where n is 1, a *unigram*, or more commonly known as a *word*).



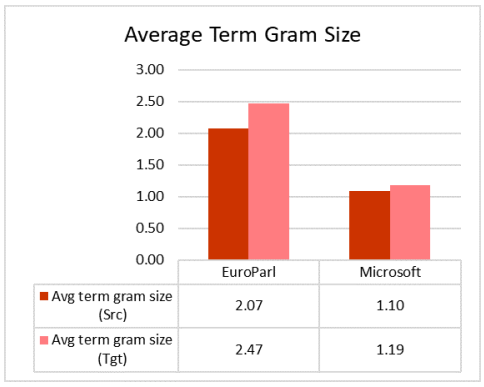
(a) Average words per sentence



(b) Average terms per sentence



(c) Average percentage of sentences with terms



(d) Average term gram size

Figure 4.4: Term statistics by corpus and metric

4.6 Injection Examples

Table 4.2 lists some interesting examples of terminology injection results produced by Method 2. Note that TAGS refers to the tagged source sentence with approved terms identified and tagged, PRED refers to the prediction (the normal output from the NMT Decoder), PMOD refers to the modified prediction with injected terms that match the tagged terms in TAGS, and REFS refers to the reference target sentence. When evaluating NMT output with terminology injection using a metric such as BLEU, PMOD (without any tagging artifacts) is compared to its corresponding REFS to produce a score.

1. Sample 1 is interesting because the prediction incorrectly used the word *patrón* as the translation for *template*. Using terminology injection, the term *plantilla*, which is the correct equivalent for *template*, was inserted into the correct position to bring the final machine translation closer to the reference translation.
2. This sample is interesting for at least four reasons:
   (a) It demonstrates multiple term injections into a single sentence, with each term correctly recognized and correctly placed.
   (b) This sample contains an "unknown" word—a word that the NMT system did not know and did not know how to handle. The word *add* was simply allowed to pass through to the prediction by the NMT Decoder. This represents a case of either the OOV or the RW problem. However, injection was able to catch that term and correct it: The English word *add* became the Spanish word *agregar*, which is the correct term used in the reference.
   (c) The injection was able to add a term that was left out of the prediction: the bigram *push operation* was incorrectly translated as just the unigram *operación* when, in this case, it should have been *operación de inserción*. While the injection did not make the translation perfect, it did restore a portion that had been left out altogether, making the final machine translation better and closer to the reference.

	Type	Sentence
1	**TAGS**	The {template} creates four files for you.
	PRED	El patrón crea cuatro expedientes.
	PMOD	El {plantilla} crea cuatro expedientes.
	REFS	La plantilla crea cuatro archivos.
2	**TAGS**	As a final {step}, let's {add} the {information} for the last {push} {operation}.
	PRED	Como último paso, add la información para la última operación.
	PMOD	Como último {paso}, {agregar} la {información} para la última {operación} {insertar}.
	REFS	Como paso final, vamos a agregar la información de la última operación de inserción.
3	**TAGS**	Run the {sample}, and {check} the {output}.
	PRED	Run la muestra y comprobar la producción.
	PMOD	Run la {muestra} y {cheque} la {salida}.
	REFS	Ejecute el ejemplo y compruebe la salida.
4	**TAGS**	You'll need to {setup} your {machine} to {run} .NET {core}.
	PRED	'll necesita setup su máquina de gobernar.
	PMOD	'll necesita {configuración} su {máquina} de {ejecutar} {núcleo}.
	REFS	Deberá configurar la máquina para ejecutar .NET core.
5	**TAGS**	This {method} leverages {C#} out parameters to indicate if the input {string} can be converted to a double.
	PRED	Este método leverages los parámetros para indicar si la cadena de aportación puede convertirse en un doble.
	PMOD	Este {método} leverages {C#} parámetros para indicar si la {cadena} de aportación puede convertirse en un doble.
	REFS	Este método aprovecha parámetros de salida de C# para indicar si la cadena de entrada se puede convertir en un doble.
6	**TAGS**	You can now {instantiate} a Book {object}, invoke both its unique and inherited members, and pass it as an {argument} to a {method} that expects a {parameter} of {type} Publication or of {type} Book, as the following example shows.
	PRED	Ahora puede instantiate un archivo de un libro, invocar tanto a sus miembros únicos y inherited como un argumento para un método que espera un parámetro de homologación o de tipo Book, como demuestra el siguiente ejemplo.
	PMOD	Ahora puede {crear una instancia} un {objeto} de un libro, invocar tanto a sus miembros únicos y inherited como un {argumento} para un {método} que espera un {parámetro} de {tipo} o de {tipo} Book, como demuestra el siguiente ejemplo.
	REFS	Ahora se puede crear una instancia de un objeto Book, invocar sus miembros únicos y heredados, y pasarla como argumento a un método que espera un parámetro de tipo Publication o de tipo Book, como se muestra en el ejemplo siguiente.
7	**TAGS**	These properties have {built-in} conversions from the {string} {type} (which is what the {JSON} packets contain) to the {target type}.
	PRED	Estas propiedades han cambiado-en conversions del tipo de string (que es lo que contienen los paquetes JSON).
	PMOD	Estas propiedades han cambiado-en conversions del {tipo} de {cadena} (que es lo que contienen los paquetes {JSON}) {tipo de destino}.
	REFS	Estas propiedades tienen integradas conversiones de tipo cadena (que es lo que contienen los paquetes JSON) para el tipo de destino .

Table 4.2: Sample injections

(d) Also note that the Spanish equivalent for the English word *step* in the source sentence was identified by the injection method. Although the correct Spanish equivalent, *paso*, already existed in the prediction, the injection simply re-inserted it in the correct location without introducing any negative consequences.

3. Sample 3 demonstrates both failure and success with the terminology collection.

   (a) The Spanish equivalent for the English term *sample* is incorrectly listed for this case as *muestra*. According to the reference, it should be *ejemplo*.

   (b) The prediction generated *comprobar* (an infinitive) for *compruebe* (a conjugated form of *comprobar*) provided in the reference. The injection correctly substituted the incorrect term *check* as a replacement for *comprobar* based on the terminology collection.

   (c) The injection correctly substituted *salida* for *producción* in the prediction as the equivalent for the English term *output*, matching the reference.

4. This sentence also demonstrates multiple injections. More importantly, the prediction is not very much like the reference. However, injection adds the correct terms to bring the prediction closer to the reference.

5. Microsoft-specific terminology, *C#*, is used in this example.

6. Multiple correctly-placed injections and a 1:many sequence figure in this long sentence.

7. In this example, a many:many sequence is handled appropriately: *tipo de destino*, the equivalent for *target type*, is correctly injected, even though the prediction omitted it.

4.7 Limitations of BLEU Scores

Nevertheless, NMT output may tend to fare worse with automated BLEU scoring than SMT output, despite the fact that NMT results are often judged as more fluent, natural, and

preferred when reviewed by human experts. Furthermore, a BLEU score, like all other scoring methods, can be irrelevant if it is used improperly or is unqualified.

Koehn [16] points out another limitation in that BLEU does not work on single sentences, since 4-gram precision is often 0. Regarding 4-gram precision, Coughlin [6] observes that a three-word sentence that is identical to the reference sentence would receive a BLEU score of zero. Therefore, scoring individual examples is not possible with BLEU.

When comparing an NMT output translation (prediction) with a test corpus reference translation, the BLEU score can only reflect how similar the two are. The BLEU score cannot directly indicate a translation improvement with terminology enhancement methods. That is to say, a translation may actually be improved according to human judgment using terminology injection, but if it does not correlate to the existing reference translation (which may not even contain the desired terminology), the automated BLEU score may be lower than what the actual improvement would merit.

To demonstrate this disparity, consider the following example sentences together with the approved terminology in Table 4.3.

Source	Target
report	informe

Table 4.3: Example approved terminology

(24) The committee reviewed your **report**. (*source*)

(25) El comité examinó su **informe**. (*NMT output, approved terminology*)

(26) El comité examinó su **exposición**. (*human-translated target reference*)

If (26) does not use the approved terminology as provided in Table 4.3, then the resulting BLEU score when comparing (25) to (26) will be less than perfect. However, (25) is a perfectly acceptable translation and even more correct or preferred than (26) because it uses the approved terminology—"report" is translated as "informe"—whereas (26) does not use approved terminology—"report" is translated as "exposición" instead.

This is a potential limitation of the BLEU score in this evaluation. In fact, a small percentage of the reference translations in the Microsoft corpus did not adhere in every case to the official terminology collection. This means that although terms were correctly tagged in the source sentence and the NMT output was corrected accordingly with approved terminology, in some cases the reference translations—which should have contained the approved terminology but did not—resulted in a mismatch, and potentially a lower BLEU score than deserved. Many such mismatches between the output set and the reference set would incorrectly lower the overall BLEU score. I point out that all current MT metrics rely on the "output versus reference" model, and therefore have this same limitation as BLEU. However, the Microsoft corpus used in this evaluation is very faithful to its companion terminology collection.

Despite the limitations of BLEU, it is still a valuable measurement and can still be used to indicate significant improvements in a supervised translation.

Chapter 5

Conclusions and Future Work

I have presented my research and approaches to injecting supplied terminology into NMT. I have contributed to the state of the art by introducing the concept of sequence cases and two methods for terminology injection. I further demonstrate that one of these methods can deliver significantly-improved translations for all four sequence cases. This method, Method 2, uses the attention mechanism prevalent in NMT systems. Additionally, I have demonstrated that this method is useful for significantly improving translations for in-domain as well as out-of-domain corpora, and also contributes to solving or mitigating the OOV problem as well as the RW problem, both common problems in MT systems.

The following paragraphs describe possibilities for future work in NMT terminology injection.

As advances and changes are made to the attention mechanisms used in NMT, the methods I present here as contributions must be updated as well. To maintain relevance, these contributions must stay current with advances in attention mechanisms in particular, and NMT in general.

As mentioned previously, OpenNMT PyTorch uses single-token vocabularies. Although this is done for good reason, it poses limitations on the relationships between the attention vectors and the final output. Extending Method 1 to handle all four sequence cases will be of great benefit to terminology injection for improvement of NMT output.

In the current implementation of Method 2, I provide a static term replacement. The form of the static term does not always agree linguistically with the surrounding text (refer

to subsection 3.3.1, Terminology Injection Method 1). Perhaps the NMT Decoder could be harnessed to provide the correct agreement as it does with the rest of the translation, or perhaps a separate trained Long Short-Term Memory (LSTM) network module could be used to make proper adjustments to the static term to agree with its placement in the translation.

In the current implementation of Method 2, I provide a simple one-to-one mapping of source to target terms using a Python dictionary such as the following:

```
terms = {
    'house': 'casa',
    'market': 'mercado',
    'dog': 'perro',
    'ice cream': 'helado'
}
```

This allows Method 1 to access supplied terminology quickly and efficiently, and is similar to Table 2.1. However, the ability to select a target term from a list of potentially multiple, approved options would be valuable. Also, the ability to identify whether an existing target term is viable in cases where target variation exists would be valuable.

My implementation of Method 1 handles both Approved and Rejected terms using two different terminology dictionaries; however, Method 2 currently only handles Approved terms using only a single terminology dictionary. Method 2 could be updated to handle Rejected terms as well.

Although there are many useful metrics for measuring MT translations, a metric specifically for measuring the effect of terminology injection on a translation, whether relying on a reference translation or not, does not exist. Comparing an MT output translation with correctly applied terminology to a reference translation that does not contain the terminology is fraught with incommensurability. Future work could include the development of such a metric.

References

[1] Ehsan Amjadian, Diana Inkpen, Tahereh Paribakht, and Farahnaz Faez. Local-global vectors to improve unigram terminology extraction. In *Proceedings of the 5th International Workshop on Computational Terminology (Computerm2016)*, pages 2–11. The COLING 2016 Organizing Committee, 2016.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *Computing Research Repository (CoRR)*, abs/1409.0473, 2014. URL `http://arxiv.org/abs/1409.0473`.

[3] Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. Findings of the 2014 Workshop on Statistical Machine Translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58. Association for Computational Linguistics, 2014.

[4] H. Choi, K. Cho, and Y. Bengio. Context-Dependent Word Representation for Neural Machine Translation. *ArXiv e-prints*, 2016.

[5] Merley Conrado, Thiago Pardo, and Solange Rezende. A machine learning approach to automatic term extraction using a rich feature set. In *Proceedings of the 2013 NAACL HLT Student Research Workshop*, pages 16–23. Association for Computational Linguistics, 2013.

[6] Deborah Coughlin. Correlating automated and human assessments of machine translation quality. In *Proceedings of MT summit IX*, pages 63–70, 2003.

[7] Leonard Dahlmann, Evgeny Matusov, Pavel Petrushkov, and Shahram Khadivi. Neural machine translation leveraging phrase-based models in a hybrid search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1411–1420. Association for Computational Linguistics, 2017.

[8] Michael Denkowski and Graham Neubig. 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, 2017.

[9] George Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, pages 138–145, 2002.

[10] Loïc Dugast, Jean Senellart, and Philipp Koehn. Statistical post-editing on SYSTRAN's rule-based translation system. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 220–223. Association for Computational Linguistics, 2007.

[11] Brian Harris. Bi-text, a new concept in translation theory. *Language Monthly*, 54:8–10, 1988.

[12] Eva Hasler, Adrià Gispert, Gonzalo Iglesias, and Bill Byrne. 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, 2018.

[13] Amir Hazem and Emmanuel Morin. Improving bilingual terminology extraction from comparable corpora via multiple word-space models. In N. Calzolari (Conference Chair), K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, and S. Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 4184–4187. European Language Resources Association (ELRA), 2016. ISBN 978-2-9517408-9-1.

[14] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 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, 2015.

[15] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72. Association for Computational Linguistics, 2017.

[16] Philipp Koehn. Statistical significance tests for machine translation evaluation. In Dekang Lin and Dekai Wu, editors, *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 388–395. Association for Computational Linguistics, 2004.

[17] Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39. Association for Computational Linguistics, 2017.

[18] Philippe Langlais. Opening statistical translation engines to terminological resources. In Birger Andersson, Maria Bergholtz, and Paul Johannesson, editors, *Natural Language Processing and Information Systems*, pages 191–202. Springer, 2002.

[19] Marie-Claude L'Homme. Management of terminology in a machine-translation environment. *International Journal of Theoretical and Applied Issues in Specialized Communication*, 1(1):121–135, 1994.

[20] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In Marie-Francine Moens and Stan Szpakowicz, editors, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81. Association for Computational Linguistics, 2004.

[21] Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. Neural machine translation with supervised attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3093–3102. The COLING 2016 Organizing Committee, 2016.

[22] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 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, 2015.

[23] Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 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, 2015.

[24] Franz Josef Och. 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, 2003.

[25] Tsuyoshi Okita, Ali Hosseinzadeh Vahid, Andy Way, and Qun Liu. DCU terminology translation system for medical query subtask at WMT14. In *Proceedings of the Ninth*

Workshop on Statistical Machine Translation, WMT@ACL 2014, June 26-27, 2014, Baltimore, Maryland, USA, pages 239–245, 2014.

[26] Patrick Pantel. 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, 2005.

[27] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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, 2002.

[28] Jérôme Rocheteau and Béatrice Daille. TTC TermSuite - a UIMA application for multilingual terminology extraction from comparable corpora. In *Proceedings of the IJCNLP 2011 System Demonstrations*, pages 9–12. Asian Federation of Natural Language Processing, 2011.

[29] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics, 2016.

[30] Rico Sennrich, Barry Haddow, and Alexandra Birch. Edinburgh neural machine translation systems for WMT16. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 371–376. Association for Computational Linguistics, 2016.

[31] Qi Zhang, Yang Wang, Yeyun Gong, and Xuanjing Huang. Keyphrase extraction using deep recurrent neural networks on Twitter. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 836–845. Association for Computational Linguistics, 2016.

[32] Ying Zhang, Stephan Vogel, and Alex Waibel. Interpreting BLEU/NIST scores: How much improvement do we need to have a better system? In *Proceedings of Language Evaluation and Resources Conference (LREC)*, pages 2051–2054, 2004.