Integrating Terminology into Neural Machine Translation Models

Table of Contents

1 Introduction

In translation, you often need to adhere to specific terminological requirements of a particular target text. With machine translation (MT) playing an increasingly important role in the translation industry, difficulties of MT models to comply with those terminological constraints, and the lack of domain-specific training data in specific language pairs become increasingly apparent (Chu/Wang 2018). To ensure better translation quality with regard to terminology, MT models need to be trained with domain-specific data. This process is called *domain adaptation*, which, in chapter 2 below, will be defined and situated in the Professional Machine Translation Literacy Framework and the DataLit^{MT} Framework proposed by Krüger (2022).

In the following chapters, we will describe different strategies on how to ensure the consistent use of terminology in neural machine translation (NMT) based on a fictional scenario.

2 Framing Domain Adaptation

In the context of MT, domain adaptation describes creating an MT model that is specialised on a specific domain. The resulting domain-adapted MT model will perform very well when translating *in-domain* texts and very poorly when translating *out-of-domain* texts. One way of applying domain adaptation is by first training a *generic* NMT model with out-of-domain parallel corpora, and afterwards fine-tuning it using in-domain corpora (Freitag/Al-Onaizan 2016; Chu/Wang 2018; Ala/Sharma 2020). By using domain adaptation, the overall translation quality in the specific domain is improved (Ala/Sharma 2020), making it more usable in a professional environment.

There are two different basic approaches to NMT domain adaptation: *architecture-centric* and *data-centric*. For architecture-centric approaches, you add trainable parameters to the NMT model itself. For data-centric approaches, you use targeted training data to achieve domain adaptation (Saunders 2021:31ff.). In the following chapters, we will focus only on the data-centric approach since it provides an easier way of integrating terminology into MT output.[1]

[1] Cf. Ramírez-Sánchez (2022:77): "[W]hile customization through data is well within the reach of a wider range of language professionals, customization through techniques remains the preserve of researchers and developers."

2.1 MT Domain Adaptation in the Professional Machine Translation Literacy Framework

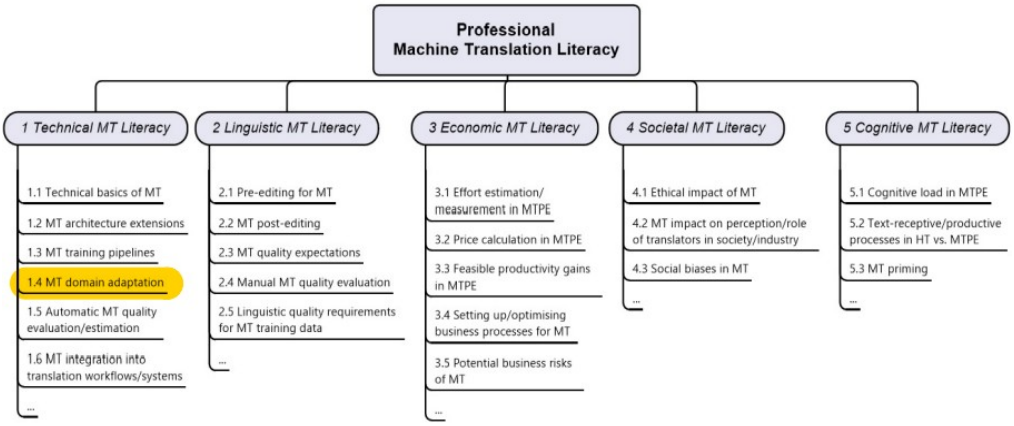


Fig. 1: Professional Machine Translation Literacy Framework (Krüger 2022:3)

Krüger (2022:3) proposed a framework to describe an extensive range of MT-related competencies that may be required in modern MT-assisted professional translation scenarios: the *Professional Machine Translation Literacy Framework*. As you can see in figure 1 above, the framework consists of five dimensions: Technical MT Literacy, Linguistic MT Literacy, Economic MT Literacy, Societal MT Literacy and Cognitive MT Literacy. The topic of this paper is associated with Technical MT Literacy, and here particularly with its fourth subdimension *MT domain adaptation* (Krüger 2022:5).

2.2 MT Domain Adaptation in the DataLit^{MT} Framework

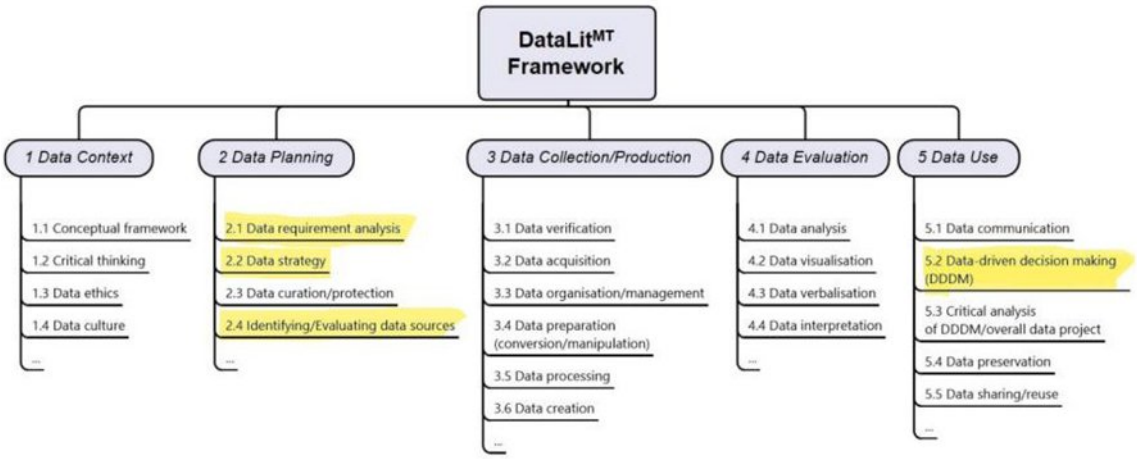


Fig. 2: DataLitMT Framework (Krüger 2022:14)

In addition to the Professional Machine Translation Literacy Framework, Krüger (2022) also developed the DataLit^{MT} Framework which attempts to describe the general data lifecycle of a machine translation project (figure 2, ibid.:14). The framework consists of five dimensions: Data Context, Data Planning, Data Collection/Production, Data Evaluation and Data Use (ibid.:14). In this framework, there are a couple of subdimensions that are particularly relevant to MT domain adaptation. These are highlighted in figure 2 and are subsumed under the main dimensions of *Data Planning* and *Data Use* as they concern searching for and choosing suitable data to achieve domain adaptation by satisfying specific requirements and prerequisites (Krüger 2022:15, 19). The decision for a specific data set is driven by evaluating its accessibility, relevance, usability and trustworthiness, but also by the impact it has on the MT system and its

translation output (Schüller 2020:31; Krüger 2022:15, 17; Ridsdale et al. 2015:38). In summary, the subdimensions highlighted in figure 2 above refer to the considerations required to choose an appropriate data set for adapting an MT model to a specific domain.

3 Use Case Scenario

After this short introduction to MT domain adaptation and its place in the DataLit^{MT} project, you probably still only have a vague idea of how this topic may be relevant to you and the translation industry. However, MT domain adaptation (here with a particular focus on terminology integration) could become a daily part of your work. The following scenario might help you understand the importance of terminology and its integration into NMT.

After graduation, you accepted a job at a biotech company in the UK that supplies diagnostic tests, instruments and digital solutions, focusing on personalised healthcare. Addressing the diagnosis and treatment of diseases, the distributed products include antigen tests for Covid-19, medical devices such as glucose meters, or medications for dermatological issues such as acne.

In this company, you work in the Technical Writing & Translation department, where you mainly coordinate text production and translation into relevant languages. Currently, the products are only distributed in European countries, which already requires your department to provide documentation in up to 20 languages. After a couple of months working in this new environment, you already learned by heart how important efficient management and standardisation of terminology, sentences, layout and other aspects are.

Some months ago and together with a language service provider (LSP), the company introduced a newly developed in-domain NMT model[2] into the translation workflow with very satisfying results, especially for the highly standardised package leaflets for drugs. The quality of the output is satisfactory enough to reduce the time needed for translation (post-editing effort). Major disadvantages, however, are the lack of control over the MT output and the inability to adapt the translation to specific situational requirements.

So far, these disadvantages have not led to removing the engine from the workflow again. Lately, however, your company has been getting a lot of complaints concerning the linguistic quality of its product documentation, which has already cost a lot of money. Still, it was not until patient A from Germany wrongly applied acne medication because they misunderstood the instructions of use that the company was forced to find a solution and change the workflow accordingly. But what was the reason for the complaints?

The company conducted a survey. Through an analysis of the answers, the reason for many of the complaints became clear:

1) The terminology in the target texts was inconsistent and ambiguous. For example, when using the diagnostic test to identify a Corona infection, the following sentences were used while conveying the same meaning:

1. *Übertragen Sie die **Probeabnahme** auf die Testkassette.*
2. *Übertragen Sie die **Lösung** auf die Testkassette.*

[2] If you want to learn more about how NMT works and how it differs from previous MT architectures, you can find popularising illustrations in Forcada (2017), van Genabith (2020) and Pérez-Ortiz et al. (2022). If you are looking for one in German, we recommend Krüger (2021).

In this example, you can see that two different terms were used for *sample*: "Probeaufnahme" and "Lösung". This needs to be solved as it leads to misunderstandings. When deciding on what term to use, you need to consider which term describes the meaning as clearly as possible. In [1], the term "Probeabnahme" does not convey that the sample needs to be put into the extraction solution before dripping it onto the sample. [2] has the opposite effect: the term "Lösung" clarifies that the solution needs to be put onto the sample, but not that the sample needs to be put into the extraction solution before this step. In conclusion, both sentences are vague and may give rise to misunderstandings. For this reason, the term should be changed to the more transparent term "Probelösung", which contains the words *sample* and *solution* and thus makes its meaning clearer.

2) Another issue was that most patients didn't understand the medical terms due to their lack of domain-specific knowledge. Consider the following sentence:

3. *Unter Proteasehemmern wurde eine CPK-Erhöhung,* ***Myalgie****,* ***Myositis*** *und in seltenen Fällen über eine Rhabdomyolyse berichtet, insbesondere im Zusammenhang mit Nukleosidanaloga.*

For people with no medical education, this sentence is very hard to understand. In the questionnaire, one patient replied that she tried to check if her muscle pain was caused by one of the company's medications she took. As *muscle pain* was not listed as one of the side effects, she didn't think of the medication as the source. Later, her muscle pain got worse and some muscles became inflamed. You see, although the terms *muscle pain* ("Muskelschmerz") and *muscle inflammation* ("Muskelentzündung") were not listed as side effects, their technical synonyms "Myalgie" and "Myositis" were. If commonly known terms had been used or added, the patient may have recognised her pain as a side effect and her condition wouldn't have gotten worse.

Such questionnaires and similar situations made it clear that the terminology of the technical documentation needed to be standardised. In a meeting with your colleagues, you discuss how to proceed, and you mention the possibility to integrate terminological constraints into the MT model. You already heard about this method in your studies, although you didn't have detailed information about how to implement these constraints. After some consideration, your boss decided to go ahead with this adjustment and tasked you and your team with researching and implementing domain adaptation using terminological constraints.

4 Strategies for Terminology Integration

In recent years, different approaches to enhancing terminology in NMT have emerged. Terminology can either be enforced at the training stage or at decoding—i.e. when the NMT model is generating the translation (Ailem et al. 2021). When you enforce desired terms at decoding, you can be certain that the desired terms are in the output. This is less the case with enforcement at training. Following this idea, Chen et al. (2020:3588) differentiate between hard lexically constrained decoding and soft lexically constrained decoding. In both cases, constraints regarding terminology are in some way implemented in the NMT model to make it "constraint-aware" (Wang et al. 2022).

Let's have a look at how you can enhance terminology in an NMT model!

4.1 Terminology Integration at Decoding: Hard Lexically Constrained Decoding

When using hard lexically constrained decoding to enhance terminology at decoding, you are going to work with a *beam search algorithm*. But what is that?

"The beam search algorithm selects multiple tokens for a position in a given sentence based on conditional probability" (Payne 2021). So, the beam search algorithm is sort of a final decision-making instance in an NMT model's decoder that selects the term with the highest probability from a set of terms at a given decoding step. For your desired term to be present in the output, you have to manipulate this beam search algorithm. There are various approaches to do so:

You can use a Grid Beam Search (GBS) to extend the beam search algorithm to a set of pre-specified translations (Hokamp/Liu 2017) or strictly constrain the words or phrases in the output using the Constrained Beam Search (CBS) (Anderson et al. 2017). Another possibility is to implement Dynamic Beam Allocation (DBA) (Post/Vilar 2018) which is a faster algorithm based on GBS that reduces decoding complexity and thus improves scaling to a larger constraint set without increasing computational overhead drastically (Dinu et al. 2019:3063).

With this approach, however, you must bear in mind that the constraints are not changed in any way by the decoder—a copy behaviour is adapted by the model regarding the terminology (Dinu et al. 2019:3063; Chen et al. 2020:3588). While this might be acceptable for entities such as brand names or product names, it gives room for morphological errors regarding vocabulary, e.g. using the singular instead of the plural (Song et al. 2019:455; Exel/Buschbeck 2021:6; Chen et al. 2020:3588). Now, imagine that you have a morphologically complex language such as Russian, German or Latvian. A hard-constrained method will rather result in errors than provide the flexibility to adapt to different root morphemes (Bergmanis/Pinnis 2021:3105).

Additionally, even though Post/Vilar (2018:183) tackled the problem of a rapidly increasing decoding time with every constraint added, the overhead time at decoding is only reduced to a constant factor (Dinu et al. 2019:3063). But recent powerful NMT architectures, such as Transformers (Vaswani et al. 2017), already have high data processing requirements due to their need for large volumes of training data. Therefore, recent approaches rather try to eliminate computational overhead at decoding time (Dinu et al. 2019:3063).

4.2 Terminology Integration at Training: Soft Lexically Constrained Decoding

To eliminate computational overhead and thus meet the requirements and needs of the latest developments in NMT, a new approach was required. Researchers now focused less on decoding and more on training time—specifically on how to augment the training data to incorporate domain-specific terms.

4.2.1 Masking

One approach that is headed towards this objective is the masking technique. Before training the model, the source and translation terms in the training data are masked by replacing them with "placeholders" (Michon et al. 2020:3926). Due to this, when the model is trained, it has learned to produce target sentences with the corresponding placeholders. Let's see what such outputs could look like (based on Michon et al. 2020:3926):

These <term#1> can be <term#2>.

Diese <term#1> können <term#2> sein.

Naturally, that is not an output we would like to see. Instead of placeholders, we would rather like to have the correct or desired term in the output. This is why the decoding process is

changed as well: The source terms in the test set are first replaced by placeholders, which are then replaced by the target terms. This leaves us with a target sentence where the placeholders of the source terms are replaced by the desired target terms:

These Corona variants can be contagious.

Diese Coronavarianten können infektiös sein.

For recognition and copy behaviour at this specific step, it is crucial that you use the same placeholders in the training and test set for the respective terms (Michon et al. 2020:3926; Crego et al. 2016).

The masking technique can be regarded as both soft and hard lexically constrained decoding as it involves changes in both stages: training and decoding. In addition, the problem regarding copy behaviour, as explained above, still disregards any morphological adjustments to the target term because it was implemented at decoding time. Due to this, the masking technique is subject to many errors, can slow down the NMT process and lacks flexibility (Läubli 2020:39; Exel/Buschbeck:6; Michon et al. 2020:3926).

Let's have a look at the first example with placeholders again: could you have fully retained the meaning of the masked word as shown in the second example? This problem especially leads to difficulties in producing a morphologically sound translation and to a loss of adequacy and fluency in the output (Song et al. 2019:449; Exel/Buschbeck 2021:6).

4.2.2 Inline Term Annotation (Data Augmentation)

An approach that only considers augmenting the training data is Inline Term Annotation (Dinu et al. 2019). This method suggests that you add auxiliary knowledge to the training data by either replacing the source terms with the respective target terms or appending them in a pre-processing step. For example, consider the following source sentence from a training data set:

*After an overdose, one of the adverse reactions reported was **hypertension**.*

When replacing the highlighted source term with the respective target term, you get the following sentence in the training data:

*After an overdose, one of the adverse reactions reported was **erhöhter Blutdruck**.*

When appending the target term, you get the following sentence in the training data:

*After an overdose, one of the adverse reactions reported was **hypertension erhöhter Blutdruck**.*

In both cases, you can easily spot the differences in the training data and therefore can tell which term is the source term and which one is the target term. An NMT model, however, cannot make these assumptions without any additional information. This is why code-switching is needed by using *source-side factors* (Sennrich/Haddow 2016). More on that later.

In comparison to data masking, the idea of this approach is that the NMT model learns the copy behaviour at training time, which is why these models are also referred to as train-by models (Dinu et al. 2019; Exel et al. 2020). This way, additional computational overhead time at decoding is eliminated (Dinu et al. 2019:3063). They additionally included morphological variance of a term by using approximate matching (ibid.:3064), also referred to as fuzzy matching. This is "a technique that, given a target string, will find its closest match from a list of non-exact matches" (Tracanna 2021). However, the application was only successful in some cases (Dinu et al. 2019:3066).

This can become a major problem when dealing with morphologically complex languages, such as Russian, German or Latvian. They often have more than one morphological form for one word and therefore a stronger flexibility regarding the copy behaviour is necessary, especially when there is a difference between the source and the target term regarding their morphological form (Bergmanis/Pinnis 2021:3105f).

Due to this, a modification of this approach was proposed: instead of the exact target language translation, you append the target language lemma to the source term in the training data (see figure 3). This forces the model to adapt a copy-and-inflect behaviour instead of a copy behaviour (Bergmanis/Pinnis 2021:3106).

EN Src.: **LV Trg.:**	faulty engine or in transmission[..] atteice dzinējā vai transmisijas [..]
ETA:	faulty|w engine|s dzinējā|t or|w transmission|s transmisijas|t [..]
TLA:	faulty|w engine|s dzinējs|t or|w transmission|s transmisija|t [..]

Fig. 3: Target lemma annotation (TLA) (Bergmanis/Pinnis 2021:3106)

In addition to appending the target language lemmas to the source terms, it is also necessary to indicate which tokens are source and target terms and which ones are just regular source words. This can be achieved by augmenting the training data with *source factors*. Source factors are annotations to each token (Dinu et al. 2019:3063), indicating the difference between source terms (**1**), target terms (**2**), and regular source words (**0**). For example, data augmentation can be done in the following manner (based on Hieber et al. n. d.):

After|**0** *an*|**0** *overdose*|**0** *one*|**0** *of*|**0** *the*|**0** *adverse*|**0** *reactions*|**0** *reported*|**0** *was*|**0**

hypertension|**1** *erhöhter*|**2** *Blutdruck*|**2**.|**0**

As you can see, every token has a source factor attached (highlighted in blue). It is irrelevant which source factors you use—just make sure to annotate three different factors consistently and delimit them with a | symbol (Hieber et al. n. d.). In figure 3, for example, the authors used *w* (regular source word), *s* (source term) and *t* (target term) as source factors.

During data augmentation, you only annotate 10% of the training data to not overfit the NMT model and to keep a stable generalisation ability regarding sentences without source factors (Dinu et al. 2019:3064).

4.2.3 Recap

As you can see, there are different ways to implement terminology enforcement. But what is the best way for you and your company? Let's go through the different options mentioned in chapter 2 again: the first method is an architecture-centric approach that requires changes to the architecture of the MT model. As the model is already in use and the method requires in-depth knowledge about the model as well as adequate programming skills, you already rejected this option at the beginning. The second, data-centric approach is a more suitable option, so you researched this method in more detail:

One way is hard lexically constrained decoding (see chapter 4.1), where you implement constraints using a beam search. Here, you also need programming skills to insert the algorithm into the model. Although you already know a lot about NMT and the model structure, it probably won't be enough to develop a fitting algorithm, implement it, and change it, if necessary, in the given time. This, combined with the added processing requirements at

decoding time and NMT models' need for large training data, made you doubt that this is the most appropriate method.

The next option is soft lexically constrained decoding (see chapter 4.2). Your research has shown that data augmentation (see chapter 4.2.2) shows more promising results in comparison to masking since morphological adjustments to the term and its context (e.g., articles, adjectives, etc.) are more likely to occur in a data augmentation scenario.

One way of augmenting training data is proposed by Dinu et al. (2019). Many other researchers also use Dinu et al. (2019) as a foundation for their data augmentation experiments, e.g. Exel et al. (2020). They also address problems with the approach proposed by Dinu et al. (2019) and find solutions for those issues, thus making the method feasible for practical applications.

In the following chapters of this paper, however, we will not address the practical application of these methods but present a related alternative that is easier to implement. So, is there another way of achieving consistent use of terminology in NMT output to reduce post-editing effort?

5 The Glossary Feature

5.1 Glossaries in NMT – Video Introduction and Tutorial

After your research, you consult with the LSP who helped your company introduce machine translation. They inform you that they also offer a feature for enforcing desired target terms in NMT output: the glossary feature. To use this feature, you need terminology lists containing source terms and the respective target terms. These terminology lists need to meet specific criteria. To help you learn more about glossaries and their potential in NMT, the LSP refers you to this video tutorial that introduces you to glossaries in general and the glossary feature of DeepL. In the video, you will also learn more about how to create a glossary from a termbase.

After you've watched the video, you can continue working through this paper.

5.2 What to Consider

From a technical point of view, we cannot say for certain which method MT providers such as DeepL use when enforcing target terms in their output with glossaries. However, as you can see in the video, DeepL often manages to correctly inflect the integrated terms and their context according to the grammar of the target language, which might indicate inline term annotation/data augmentation (see chapter 4.2.2) rather than masking (see chapter 4.2.1). Nevertheless, keep in mind that terminology integration is currently the subject of intensive research and that new and/or better technologies can always be released to the market. For this reason, we recommend that you keep up to date with the latest technical developments. It may also help to consult with the LSP or MT provider that you are working with.

There are some key aspects that you need to consider when working with glossaries in NMT. Some of them were already mentioned or outlined in the video, but you can use the following compilation as a reference for your experiments (Mayer/Winter 2023:228–231). In chapter 6.4, you can find examples for some of these categories.

1) How reliably are the target terms integrated?
2) Are inflected source terms recognised and correctly translated?
3) Are only one-word terms and compound terms recognised, or is it also possible to integrate multi-word terms or phrases?
4) Are target terms integrated incorrectly? If so, how often does this occur?

5) Are target terms and their context (e.g., articles, adjectives) correctly inflected?

6 Experimental Setup

In this setup, we provide some ideas to test how terminology integration can work in a CAT tool compared to the feature in DeepL (i.e., from an MT provider) that you saw in the video. Phrase TMS (formerly known as Memsource) is a web-based CAT tool that offers the machine translation hub Phrase Translate, enabling you to use various MT engines combined with additional features such as the glossary feature. There is a free 15-day trial available that you can use to experiment with the tool. The only resources that you need are a small source text (a few segments are sufficient) and a glossary with source terms and their target language equivalents. You can find a comprehensive overview of the tool and some helpful instructions (e.g. creating MT profiles) in this article.

6.1 Choosing a Language Pair

To see if terminology enforcement is successful and to reduce the time and effort spent on the experimental setup, you conclude that only one out of all the language pairs possibly included in a project should be used. For this reason, you choose English as the source language and German as the target language. This decision was driven by multiple reasons:

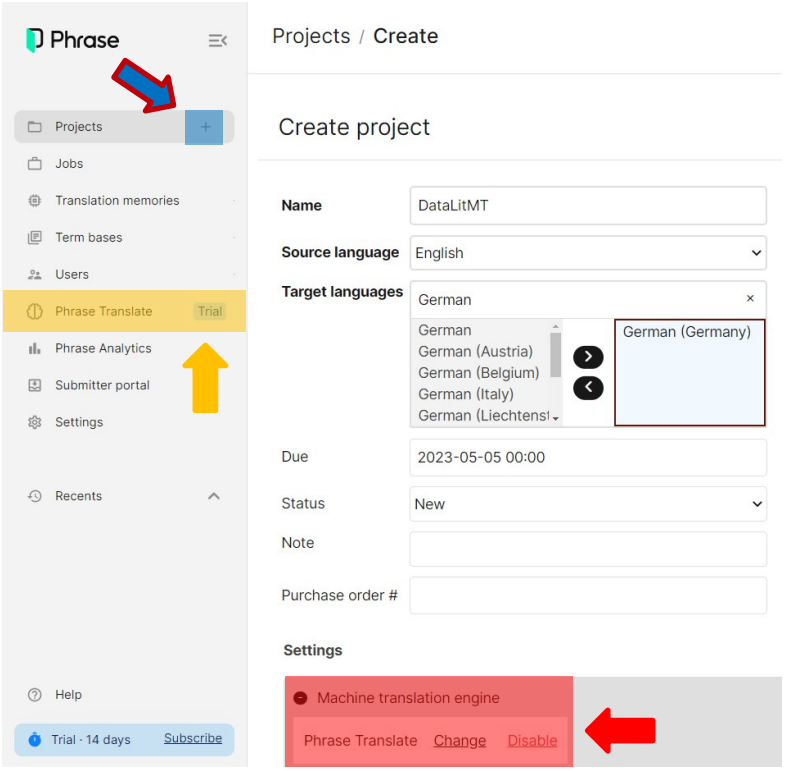
1) As your team consists mainly of native German speakers who also have a high level of proficiency in English, you can specify the correct terminology in the source and target texts. This will help with the evaluation of the MT output later.

2) English and German are high-resourced languages that can therefore be handled quite well by NMT. Bear in mind though that German is a morphologically rich language, so you need to make sure that "the approach is able to produce the target terms in an appropriate morphological form" (Exel et al. 2020:173).

3) The EN-DE language pair is the third most suitable language combination for which raw MT (without pre- and post-editing) can be used (Vashee 2020). In the actual ranking (MT with pre- and post-editing), it ranks 12th out of 49 listed languages (ibid.). Generally, EN-DE and DE-EN language combinations seem to produce high-quality MT output with satisfactory BLEU scores and Levenshtein distances (Koehn/Wiggins 2019). The BLEU score calculates the similarity between translations based on counting the number of n-grams that differ between an MT output and a reference human translation, which is the "golden standard" (ideal translation) (Kit/Wong 2015:226). The Levenshtein edit distance describes the difference between two sequences, i.e. the sum of minimal required insertions, deletions, or substitutions to transform the MT output into the golden standard (Shuwandy et al. 2020). In a nutshell: a high BLEU score and a low Levenshtein distance indicate good quality.

Note: If you want to learn more about automatic MT quality evaluation, you can find various interesting learning resources on the DataLit^{MT} website. The respective resources can be accessed here. They introduce you to basic concepts that help you understand how the different metrics such as the BLEU score work. As for the Levenshtein edit distance you just read about, these learning resources use a variation of it: the Translation Edit Rate (TER). The difference is that TER also counts *shifts* and then normalises the combined value by the length of the reference human translation. TER and the other metrics are explained in detail in the learning resource at the Advanced Level.

6.2 Phrase Translate



After starting the trial, you can create a project (blue in figure 4) to which the MT engine (meaning MT profile in this case) *Phrase Translate* should be assigned (red) automatically. To access the settings for Phrase Translate (i.e. your MT profiles), click in the sidebar on Phrase Translate (orange).

Fig. 4: Setting up a project in Phrase Translate

Phrase Translate provides you with various MT engines: Amazon Translate, DeepL, Google Translate, Microsoft Translator, Phrase NextMT, Rozetta T-400 Realtime and Tencent (see figure 5). You can add more engines if you have a license key. Among all activated engines in an MT profile, Phrase automatically selects the "best" engine, considering language pair and domain of the source text. However, you can also disable engines (see Rozetta and Tencent in figure 5). If you have activated more than one engine, you can see in the CAT editor which engine provided the target text segments. You can also create additional MT profiles to test engines individually. Keep in mind, however, that you need to attach your glossary to each additional MT profile that you create (updating the glossaries might take a few minutes). In addition, remember to attach the MT profile to your project.

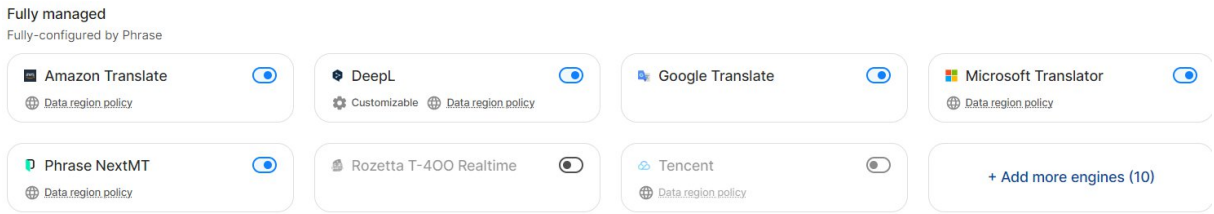


Fig. 5: Overview of the fully managed MT engines in Phrase TMS

6.3 Glossary and Source Text

For this experiment, we used the following source text (terms of interest are **highlighted**):

> *Many people suffer from **high blood pressure**. But with these new drugs, patients with **hypertension** can be treated. Possible side effects are **myalgia** and **myositis**.*

In general, the goal in this scenario is to use more commonly known terms instead of highly domain-specific terms and to apply the chosen terms consistently and unambiguously, meaning synonyms in the source text should be translated with the same target term (*high blood pressure* vs. *hypertension*). You can add frequently used and not-so-frequently used terms to your glossary as well as both single- and multi-word terms to test if multi-word terms are correctly integrated (and inflected if necessary). For example, you can include the terms "erhöhter Blutzucker" (high blood glucose) and "erhöhter Blutdruck" (high blood pressure). You can also include one of the terms that were mentioned in the questionnaires: "Muskelschmerz" instead of "Myalgie". We used the glossary in figure 6, but feel free to experiment with different glossaries or source texts.

English	German
high blood pressure	erhöhter Blutdruck
hypertension	erhöhter Blutdruck
myositis	Muskelentzündung
myalgia	Muskelschmerz

Fig. 6: Sample glossary

6.4 Analysing the MT Output

For this experiment, we compared the output of DeepL, Google Translate and Phrase NextMT both without and with a glossary.

First segment	DeepL	Google Translate	Phrase NextMT
without glossary	Viele Menschen leiden unter **hohem Blutdruck**.	Viele Menschen leiden unter **Bluthochdruck**.	Viele Menschen leiden unter **Bluthochdruck**.
with glossary	Viele Menschen leiden unter **erhöhtem Blutdruck**.	Viele Menschen leiden unter **erhöhter Blutdruck**.	Viele Menschen leiden unter **erhöhtem Blutdruck**.
Second segment	**DeepL**	**Google Translate**	**Phrase NextMT**
without glossary	Aber mit diesen neuen Medikamenten können Patienten mit **Bluthochdruck** behandelt werden.	Aber mit diesen neuen Medikamenten können Patienten mit **Bluthochdruck** behandelt werden.	Aber mit diesen neuen Medikamenten können Patienten mit **Bluthochdruck** behandelt werden.
with glossary	Aber mit diesen neuen Medikamenten können Patienten mit **erhöhtem Blutdruck** behandelt werden.	Aber mit diesen neuen Medikamenten können Patienten mit **erhöhter Blutdruck** behandelt werden.	Aber mit diesen neuen Medikamenten können Patienten mit **erhöhtem Blutdruck** behandelt werden.
Third segment	**DeepL**	**Google Translate**	**Phrase NextMT**
without glossary	Mögliche Nebenwirkungen sind **Myalgie** und **Myositis**.	Mögliche Nebenwirkungen sind **Myalgie** und **Myositis**.	Mögliche Nebenwirkungen sind **Myalgie** und **Myositis**.
with glossary	Mögliche Nebenwirkungen sind **Muskelschmerzen** und **Muskelentzündungen**.	Mögliche Nebenwirkungen sind **Muskelschmerz** und **Muskelentzündung**.	Mögliche Nebenwirkungen sind **Muskelschmerzen** und **Muskelentzündungen**.

Fig. 7: Comparison of the MT output of DeepL, Google Translate and Phrase NextMT with and without a glossary

First segment:

The output (without glossary) of Google Translate and Phrase NextMT is identical: *high blood pressure* has been translated as "Bluthochdruck". Only DeepL used the variant "hohem Blutdruck". All versions are grammatically and semantically correct but none uses our preferred target term "erhöhter Blutdruck".

The output (with glossary) of DeepL and Phrase NextMT is identical: in both cases, the preferred target term has been integrated and correctly inflected. According to the key aspects to consider when using glossaries in NMT (see chapter 5.2), this could be categorised as aspects **3)** correct integration of multi-word terms and **5)** correct inflection. Google Translate, however, only replaced the source with the target term and didn't inflect it, leading to a grammar mistake (aspect 5). This means that using the glossary feature has fixed one mistake but created another!

Second segment:

The output (without glossary) of all engines is identical: *hypertension* has been translated as "Bluthochdruck". All versions are grammatically and semantically correct but none uses our preferred target term "erhöhter Blutdruck". It is interesting to note that only DeepL is inconsistent with the use of terminology (when comparing the first and second segments).

The output (with glossary) of DeepL and Phrase NextMT is identical: in both cases, the preferred target term has been integrated and correctly inflected (aspects 3) and 5)). As in the first segment, Google integrated the term but didn't inflect it (aspect 5).

Third segment:

The output (without glossary) of all engines is identical: the uncountable nouns *myalgia and myositis* have been translated as "Myalgie" and "Myositis". This version is grammatically and semantically correct but it doesn't use the more commonly known terms "Muskelschmerz" and "Muskelentzündung" that we would prefer to see in the output.

The output (with glossary) of the engines is almost identical. All three integrated our preferred target terms. Interestingly, DeepL and Phrase NextMT produced plural nouns ("Muskelschmerzen" and "Muskelentzündungen"), which might originate from the ability to inflect integrated target terms. Google Translate, however, produced singular nouns ("Muskelschmerz" and "Muskelentzündung").

So what does that mean?

Depending on which MT engine you use in your translation workflow, the quality of your results can vary, which also affects the amount of post-editing required. Next, you could experiment with other source texts, glossaries and language pairs. For example, you could test the other language direction (German into English), which should be very interesting regarding aspect **2)** correct recognition and translation of inflected source terms.

Note, however, that we tested the performance of the different engines on just three segments. To make an informed judgment in a real scenario, you should always use a larger set of text that is representative of your use case. The results can then be used to assess the strengths and weaknesses of the respective engines (e.g. no inflection with Google Translate). If you want to compare the post-editing effort of the various documents produced during this kind of test, feel free to check out this learning nugget, where you can automatically evaluate MT quality at document level.

7 Conclusion

Congratulations, you completed your introduction to terminology integration in an NMT-assisted translation workflow. You got an impression of what domain adaptation is and how you can enforce terminology in NMT output, be it from a technical point of view (masking and data augmentation) or from the point of view of language professionals (glossary feature).

Keep in mind that terminology integration is currently the subject of intensive research and that new and/or better technologies can always come onto the market. For this reason, we recommend that you keep up to date with the latest technical developments and ideas in the industry.

References

Ailem, Melissa/Liu, Jinghsu/Qader, Raheel (2021): Encouraging neural machine translation to satisfy terminology constraints. In: Zong, Chengqing/Xia, Fei/Li, Wenjie/Navigli, Roberto (Eds.): *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, 1450–1455. https://aclanthology.org/2021.findings-acl.125/ (18 February 2023).

Ala, Hema/Sharma, Dipti (2020): AdapNMT: Neural machine translation with technical domain adaptation for Indic languages. In: Sharma, Dipti M./Ekbal, Asif/Arora, Karunesh/ Naskar, Sudip K./Ganguly, Dipankar/L, Sobha/Mamidi, Radhika/Arora, Sunita/Mishra, Pruthwik/Mujadia, Vandan (Eds.): *Proceedings of the 17th international conference on natural language processing (ICON): Adap-MT 2020 shared task*. Patna: NLP Association of India, 6–10. https://aclanthology.org/2020.icon-adapmt.2/ (17 February 2023).

Anderson, Peter/Fernando, Basura/Johnson, Mark/Gould, Stephen (2017): Guided open vocabulary image captioning with constrained beam search. In: Palmer, Martha/Hwa, Rebecca/Riedel, Sebastian (Eds.): *Proceedings of the 2017 conference on empirical methods in natural language processing*. Copenhagen: Association for Computational Linguistics, 936–945. http://dx.doi.org/10.18653/v1/D17-1098.

Aulamo, Mikko/Sulubacak, Umut/Virpioja, Sami/Tiedemann, Jörg (2020): OpusTools and parallel corpus diagnostics. In: Calzolari, Nicoletta/Béchet, Frédéric/Blache, Philippe/Choukri, Khalid/Cieri, Christopher/Declerck, Thierry/Goggi, Sara/Isahara, Hitoshi/Maegaard, Bente/Mariani, Joseph/Mazo, Hélène/Moreno, Asuncion/Odijk, Jan/Piperidis, Stelios (Eds.): *Proceedings of the twelfth language resources and evaluation conference*. Marseille: European Language Resources Association, 3782–3789. https://aclanthology.org/2020.lrec-1.467 (17 February 2023).

Bergmanis, Toms/Pinnis, Mārcis (2021): Facilitating terminology translation with target lemma annotations. In: Merlo, Paola/Tiedemann, Jorg/Tsarfaty, Reut (Eds.): *Proceedings of the 16th conference of the European chapter of the Association for Computational Linguistics: Main volume*. Online: Association for Computational Linguistics, 3105–3111. http://dx.doi.org/10.18653/v1/2021.eacl-main.271.

Bruckner, Christine (2018): Terminologie und maschinelle Übersetzung – Herausforderungen und Möglichkeiten einer Integration. In: *edition* 14(2), 5–11.

Chen, Guanhua/Chen, Yun/Wang, Yong/Li, Victor O.K. (2020): Lexical-constraint-aware neural machine translation via data augmentation. In: Bessiere, Christian (Ed.):

Proceedings of the twenty-ninth International Joint Conference on Artificial Intelligence. n. p.: International Joint Conference on Artificial Intelligence, 3587–3593. https://www.ijcai.org/proceedings/2020/0496.pdf (17 February 2023).

Chu, Chenhui/Wang, Rui (2018): A survey of domain adaptation for neural machine translation. In: Bender, Emily M./Derczynski, Leon/Isabelle, Pierre (Eds.): *Proceedings of the 27th International Conference of Computational Linguistics*. Santa Fe: Association for Computational Linguistics, 1304–1319. https://aclanthology.org/C18-1111 (17 February 2023).

Chunyu, Kit/Wong Tak-ming, Billy (2015): Evaluation in machine translation and computer-aided translation. In: Sin-wai, Chan (Ed.): *The Routledge encyclopedia of translation technology*. London: Routledge, 213–236.

Crego, Josep/Kim, Jungi/Klein, Guillaume/Rebollo, Anabel/Yang, Kathy/Senellart, Jean/Akhanov, Egor/Brunelle, Patrice/Coquard, Aurélien/Deng, Yongchao/Enoue, Satoshi/Geiss, Chiyo/Johanson, Joshua/Khalsa, Ardas/Khiari, Raoum/Ko, Byeongil/Kobus, Catherine/Lorieux, Jean/Martins, Leidiana/Nguyen, Dang-Chuan/Priori, Alexandra/Riccardi, Thomas/Segal, Natalia/Servan, Christophe/Tiquet, Cyril/Wang, Bo/Yang, Jin/Zhang, Dakun/Zhou, Jing/Zoldan, Peter (2016): SYSTRAN's pure neural machine translation systems. *arXiv*. https://doi.org/10.48550/arXiv.1610.05540.

Dinu, Georgiana/Mathur, Prashant/Federico, Marcello/Al-Onaizan, Yaser (2019): Training neural machine translation to apply terminology constraints. In: Nakov, Preslav/Palmer, Alexis (Eds.): *Proceedings of the 57th annual meeting of the Association for Computational Linguistics*. Florence: Association for Computational Linguistics, 3063–3068. http://dx.doi.org/10.18653/v1/P19-1294.

Domingo, Miguel/García-Martínez, Mercedes/Helle, Alexandre/Casacuberta, Francisco/Herranz, Manuel (2018): How much does tokenization affect neural machine translation? *arXiv*. https://doi.org/10.48550/arXiv.1812.08621.

Exel, Miriam/Buschbeck, Bianka (2021): Enforcing terminology in neural machine translation. In: *edition* 17(1), 5–12.

Exel, Miriam/Buschbeck, Bianka/Brandt, Lauritz/Doneva, Simona (2020): Terminology-constrained neural machine translation at SAP. In: Martins, André/Moniz, Helena/Fumega, Sara/Martins, Bruno/Batista, Fernando/Coheur, Luisa/Parra, Carla/Trancoso, Isabel/Turchi, Marco/Bisazza, Arianna/Moorkens, Joss/Guerberof, Ana/Nurminen, Mary/Marg, Lena/Forcada, Mikel L. (Eds.): *Proceedings of the 22nd annual conference of the European Association for Machine Translation*. Lisbon: European Association for Machine Translation, 271–280. https://aclanthology.org/2020.eamt-1.29 (17 February 2023).

Forcada, Mikel L. (2017): Making sense of neural machine translation. In: *Translation Spaces* 6(2), 291–309. https://doi.org/10.1075/ts.6.2.06for.

Freitag, Markus/Al-Onaizan, Yaser (2016): Fast domain adaptation for neural machine translation. *arXiv*. https://doi.org/10.48550/arXiv.1612.06897 (17 February 2023).

Gage, Philip (1994): A new algorithm for data compression. In: *C Users Journal* 12(2), 23–38.

Hieber, Felix/Domhan, Tobias/Denkowski, Michael/Post, Matt (n. d.): Sockeye. *GitHub*. https://github.com/awslabs/sockeye (17 February 2023).

Hieber, Felix/Domhan, Tobias/Denkowski, Michael/Vilar, David (2020): Sockeye 2: A toolkit for neural machine translation. In: Martins, André/Moniz, Helena/Fumega, Sara/Martins, Bruno/Batista, Fernando/Coheur, Luisa/Parra, Carla/Trancoso, Isabel/Turchi, Marco/Bisazza, Arianna/Moorkens, Joss/Guerberof, Ana/Nurminen, Mary/Marg, Lena/Forcada, Mikel L. (Eds.): *Proceedings of the 22nd annual conference of the European Association for Machine Translation*. Lisbon: European Association for Machine Translation, 457f. https://aclanthology.org/2020.eamt-1.50 (18 February 2023).

Hieber, Felix/Domhan, Tobias/Denkowski, Michael/Vilar, David/Sokolov, Artem/Clifton, Ann/Post, Matt (2017): Sockeye: A toolkit for neural machine translation. *arXiv*. https://doi.org/10.48550/arXiv.1712.05690.

Hokamp, Chris/Liu, Qun (2017): Lexically constrained decoding for sequence generation using grid beam search. In: Barzilay, Regina/Kan, Min-Yen (Eds.): *Proceedings of the 55th annual meeting of the Association for Computational Linguistics (volume 1: Long papers)*. Vancouver: Association for Computational Linguistics, 1535–1546. http://dx.doi.org/10.18653/v1/P17-1141.

Koehn, Philipp/Hoang, Hieu/Birch, Alexandra/Callison-Burch, Chris/Federico, Marcello/Bertoldi, Nicola/Cowan, Brooke/Shen, Wade/Moran, Christine/Zens, Richard/Dyer, Chris/Bojar, Ondrej/Constantin, Alexandra/Herbst, Evan (2007): Moses: Open source toolkit for statistical machine translation. In: Ananiadou, Sophia (Ed.): *Proceedings of the 45th annual meeting of the Association for Computational Linguistics Companion volume proceedings of the demo and poster sessions*. Prague: Association for Computational Linguistics, 177–180. https://aclanthology.org/P07-2045 (18 February 2023).

Koehn, Philipp/Knowles, Rebecca (2017): Six challenges for neural machine translation. In: Luong, Thang/Birch, Alexandra/Neubig, Graham/Finch, Andrew (Eds.): *Proceedings of the first workshop on neural machine translation*. Vancouver: Association for Computational Linguistics, 28–39. http://dx.doi.org/10.18653/v1/W17-3204.

Koehn, Philipp/Wiggins, Dion (2019): Machine translation primer. Current technology and future direction. *YouTube*. https://www.youtube.com/watch?v=K9uqL7vDikU (18 February 2023).

Krüger, Ralph (2021): Die Transformer-Architektur für Systeme zur maschinellen Übersetzung – eine popularisierende Darstellung. In: *trans-kom* 14(2), 278–324.

Krüger, Ralph (2022): Integrating professional machine translation literacy and data literacy. In: *Lebende Sprachen* 67(2), 247–282. https://doi.org/10.1515/les-2022-1022.

Mayer, Felix/Winter, Tom (2023): Terminologiearbeit (auch) für die NMÜ. In: Drewer, Petra/Mayer, Felix/Pulitano, Donatella (Eds.): *Terminologie: Tools und Technologien. Akten des Symposions Mannheim, 2.-4. März 2023*. München: Deutscher Terminologie-Tag e.V.: 225–237.

Michon, Elise/Crego, Josep/Senellart, Jean (2020): Integrating domain terminology into neural machine translation. In: Scott, Donia/Bel, Nuria/Zong, Chengqing (Eds.): *Proceedings of the 28th international conference on computational linguistics*. Barcelona: International Committee on Computational Linguistics, 3925–3937. https://aclanthology.org/2020.coling-main.348.pdf (18 February 2023).

Ottmann, Angelika/Weilandt, Annette (2021): Maskierung und Provokation – Interview mit Samuel Läubli zur NMT. In: *edition* 17(1), 38–40.

Payne, Matt (2021): What is beam search? Explaining the beam search algorithm. *Width.ai*. https://www.width.ai/post/what-is-beam-search (17 February 2023).

Pérez-Ortiz, Juan A./Forcada, Mikel L./Sánchez-Martínez, Felipe (2022): How neural machine translation works. In: Kenny, Dorothy (Ed.): *Machine translation for everyone. Empowering users in the age of artificial intelligence*. Berlin: Language Science Press, 141–164. https://doi.org/10.5281/zenodo.6760019.

Post, Matt/Vilar, David (2018): Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In: Walker, Marilyn/Ji, Heng/Stent, Amanda (Eds.): *Proceedings of the 2018 conference of the North American chapter of the Association for Computational Linguistics: Human language technologies, volume 1 (long papers)*. New Orleans: Association for Computational Linguistics, 1314–1324. http://dx.doi.org/10.18653/v1/N18-1119.

Prechelt, Lutz (2012): Early stopping – but when? In: Montavon, Grégoire/Orr, Geneviève B./Müller, Klaus-Robert (Eds.): *Neural networks: Tricks of the trade. Lecture notes in computer science.* 2nd edition. Berlin: Springer, 53–67. https://doi.org/10.1007/978-3-642-35289-8_5.

Ramírez-Sánchez, Gema (2022): Custom machine translation. In: Kenny, Dorothy (Ed.): *Machine translation for everyone. Empowering users in the age of artificial intelligence.* Berlin: Language Science Press, 165–186. https://doi.org/10.5281/zenodo.6653406.

Ridsdale, Chantel/Rothwell, James/Smit, Mike/Ali-Hassan, Hossam/Bliemel, Michael/Irvine, Dean/Kelley, Daniel/Matwin, Stan/Wuetherick, Brad (2015): Strategies and best practices for data literacy education. Knowledge synthesis report. *Dalhousie University*. http://hdl.handle.net/10222/64578 (18 February 2023).

Saunders, Danielle (2021): *Domain adaptation for neural machine translation*. University of Cambridge: PhD Thesis. https://dcsaunders.github.io/thesis.pdf (21 April 2023).

Schüller, Katharina (2020): *Future skills: A framework for data literacy.* Hochschulforum Digitalisierung. https://hochschulforumdigitalisierung.de/de/future-skills-framework-data-literacy (18 February 2023).

Sennrich, Rico/Haddow, Barry (2016): Linguistic input features improve neural machine translation. In: Bojar, Ondřej/Buck, Christian/Chatterjee, Rajen/Federmann, Christian/Guillou, Liane/Haddow, Barry/Huck, Matthias/Yepes, Antonio J./Névéol, Aurélie/Neves, Mariana/Pecina, Pavel/Popel, Martin/Koehn, Philipp/Monz, Christof/Negri, Matteo/Post, Matt/Specia, Lucia/Verspoor, Karin/Tiedemann, Jörg/Turchi, Marco (Eds.): *Proceedings of the first conference on machine translation: Volume 1, research papers*. Berlin: Association for Computational Linguistics, 83–91. http://dx.doi.org/10.18653/v1/W16-2209.

Sennrich, Rico/Haddow, Barry/Birch, Alexandra (2016): 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)*. Berlin: Association for Computational Linguistics, 1715–1725. http://dx.doi.org/10.18653/v1/P16-1162.

Shuwandy, Moceheb L./Zaidan, B. B./Zaidan, A. A./Albahri, A. S./Alamoodi, A. H./Albahri, O. S./Alazab, Mamoun (2020): mHealth authentication approach based 3D touchscreen and microphone sensors for real-time remote healthcare monitoring system:

Comprehensive review, open issues and methodological aspects. In: *Computer Science Review* 38. https://doi.org/10.1016/j.cosrev.2020.100300.

Song, Kai/Zhang, Yue/Yu, Heng/Luo, Weihua/Wang, Kun/Zhang, Min (2019): Code-switching for enhancing NMT with pre-specified translation. In: Burstein, Jill/Doran, Christy/Solorio, Thamar (Eds.): *Proceedings of the 2019 conference of the North American chapter of the Association for Computational Linguistics: Human language technologies, volume 1 (long and short papers)*. Minneapolis: Association for Computational Linguistics, 449–459. http://dx.doi.org/10.18653/v1/N19-1044.

Tiedemann, Jörg (2012): Parallel data, tools and interfaces in OPUS. In: Calzolari, Nicoletta/Choukri, Khalid/Declerck, Thierry/Doğan, Mehmet U./Maegaard, Bente/Mariani, Joseph/Moreno, Asuncion/Odijk, Jan/Piperidis, Stelios (Eds.): *Proceedings of the eighth international conference on language resources and evaluation*. Istanbul: European Language Resources Association, 2214–2218. http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf (18 February 2023).

Tracanna, Lorenzo (2021): How to perform approximate string matching in one line of code. *Towards Data Science*. https://towardsdatascience.com/how-to-perform-approximate-string-matching-in-one-line-of-code-76fae5d7efc (18 February 2023).

van Genabith, Josef (2020): Neural machine translation, In: Porsiel, Jörg (Ed.): *Maschinelle Übersetzung für Übersetzungsprofis*. Berlin: BDÜ Fachverlag, 59–115.

Vashee, Kirti (2020): American Machine Translation Association (AMTA2020) conference highlights. *eMpTy Pages*. http://kv-emptypages.blogspot.com/2020/12/american-machine-translation.html (18 February 2023).

Vaswani, Ashish/Shazeer, Noam/Parmar, Niki/Uszkoreit, Jakob/Jones, Llion/Gomez, Aidan N./Kaiser, Łukasz/Polosukhin, Illia (2017): Attention is all you need. *arXiv*. https://doi.org/10.48550/arXiv.1706.03762.

Wang, Shuo/Tan, Zhixing/Liu, Yang (2022): Integrating vectorized lexical constraints for neural machine translation. In: Muresan, Smaranda/Nakov, Preslav/Villavicencio, Aline (Eds.): *Proceedings of the 60th annual meeting of the Association for Computational Linguistics (volume 1: long papers)*. Dublin: Association for Computational Linguistics, 7063–7073. http://dx.doi.org/10.18653/v1/2022.acl-long.487.