Analyzing Challenges in Neural Machine Translation for Software Localization

Sai Koneru[1], **Matthias Huck**[2], **Miriam Exel**[2], and **Jan Niehues**[1]
[1] Karlsruhe Institute of Technology
[2] SAP SE, Dietmar-Hopp-Allee 16, 69190 Walldorf, Germany
{sai.koneru, jan.niehues}@kit.edu
{matthias.huck, miriam.exel}@sap.com

Abstract

Advancements in Neural Machine Translation (NMT) greatly benefit the software localization industry by decreasing the post-editing time of human annotators. Although the volume of the software being localized is growing significantly, techniques for improving NMT for user interface (UI) texts are lacking. These UI texts have different properties than other collections of texts, presenting unique challenges for NMT. For example, they are often very short, causing them to be ambiguous and needing additional context (button, title text, a table item, etc.) for disambiguation. However, no such UI data sets are readily available with contextual information for NMT models to exploit. This work aims to provide a first step in improving UI translations and highlight its challenges. To achieve this, we provide a novel multilingual UI corpus collection ($\sim 1.3M$ for English $\leftrightarrow$ German) with a targeted test set and analyze the limitations of state-of-the-art methods on this challenging task. Specifically, we present a targeted test set for disambiguation from English to German to evaluate reliably and emphasize UI translation challenges. Furthermore, we evaluate several state-of-the-art NMT techniques from domain adaptation and document-level NMT on this challenging task. All the scripts to replicate the experiments and data sets are available here.[1,2]

1 Introduction

There is a rapid increase in access to technology for people from around the globe. For software to be used by everybody, it is essential to provide User Interface (UI) texts in their native languages for monolingual speakers. However, many applications are created in English and later localized to various languages. To decrease the translation time, localization companies take the help of machine translation (MT). Human annotators use the MT system for generating initial translations and post-edit the system's output to increase efficiency in computer-assisted translation tools (Flournoy and Duran, 2009; Skadiņš et al., 2014). When producing high-quality translations, costs can be saved by decreasing the annotator's time on editing and making the localization process cheaper. Thus, enabling more companies to localize in several languages at low cost.

Neural Machine Translation (NMT) (Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017) is the current state-of-the-art (SOTA) approach for generating the initial "draft" for the translations. Although conventional NMT models are sufficient in many cases, they use only the source sentence alone to predict the target translation (*Locality Assumption*) (Maruf et al., 2021). However, this is problematic when translating several types of concise UI texts that need more information.

It has been a decade since Muntés Mulero et al. (2012) have shown the need for integrating additional context into MT models for software localization and have pointed out several issues. For example, consider translating the English source word "Login" into German. Here, the translation depends on where the text is. If it is on a button, the correct translation is the verb form "Anmelden". In the case of a title text (FAQ, Documentation, etc.), the translation might rather be the noun form "Anmeldung". Depending on the application, the translation can also be simply "Login" even on a button.

Another common phenomenon to deal with when translating these short UI texts is semantic ambiguity. For example, if we want to translate the word "Home", we need to know the context where the sentence is present. Although the translation is often the German word "Haus", it is not correct in the context of Linux applications. In this case, we would need to translate "Home" as "Benutzerverzeichnis" ($HOME directory). Apart from the issues

[1] https://github.com/saikoneru/NMT_Localization
[2] We crawled this data only for scientific research.

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 2442–2454
May 2-6, 2023

mentioned above, gender (the gender of the item in a table depends on the gender of the table title text), and consistency in terminology and style (e.g formal v/s informal) are problematic when translating UI. However, sentence-level NMT models lack the contextual information to generate suitable target translations.

Several approaches were proposed to integrate contextual information into NMT models. Domain adaptation and Multi-Domain models using tags are common techniques in NMT to inform the models about the type of content it is translating (Kobus et al., 2017; Chu et al., 2017; Pham et al., 2020; Xu et al., 2020). Knowing the domain/topic of the text enables NMT to select and generate appropriate translations in a particular setting. Another recent and growing field that tries to make models context-aware is document-level NMT (Doc-NMT) (Tiedemann and Scherrer, 2017; Voita et al., 2018; Maruf et al., 2019, 2021; Bao et al., 2021; Sun et al., 2022). The surrounding source and/or target sentences usually can provide significant cues and hints for the NMT model to understand the domain and the context in which the source sentence is occurring. These approaches are promising and could benefit localizing UI segments, but the lack of sufficient annotated UI data with contextual information is the main hindrance to applying them.

Although localization files of software like GNOME, UBUNTU, and KDE are present in the OPUS corpus (Tiedemann, 2012), they do not provide the document structure or any other metadata. Also, the segments in the corpora do not come from a vast amount of domains. Neither large amounts of UI data are publicly available nor a targeted test set to measure the model's ability to use context for UI translations. These constraints limit the potential to improve NMT for software localization and are necessary as a first step.

This work addresses the limitations mentioned above by creating a corpus for UI data with document-level information extracted from multiple domains, a targeted test suite, and baselines using current SOTA methods. Our main contributions in this paper are the following

- We present a task of translating UI texts and show their unique properties requiring additional context. Furthermore, we provide a novel multilingual UI corpus covering multiple domains with contextual information to enable NMT for this task. (Section 2)
- To identify the limitations in current NMT systems, we propose a targeted evaluation framework with test sets replicating realistic conditions and solving disambiguation. (Section 3)
- We analyze domain adaptation and Doc-NMT techniques on our collected corpora and evaluation sets to highlight challenges in the proposed task. (Section 4)

2 Addressing Data Scarcity for UI

For improving NMT for UI segments, the first challenge to address is the lack of parallel data with contextual information. To achieve this, we present a novel UI corpus that we assemble from software localization files of publicly available repositories.

First, we briefly explain the structure and contents of the localization files which we search and collect to create our data set. Then, we highlight what types of contextual information are available in these files that are useful for NMT. Finally, we describe how we scraped large amounts of these PO files for multiple languages.

2.1 Portable Object for Localization

Portable Object (PO) files[3] are one of the standard file formats used in the localization industry. These are plain text-based files and do not need specialized tools for reading. Although this is not the only format used for localization, it is widely used in the GNU gettext[4] tool for free software's.

Figure 1 shows an example of a PO file along with a screenshot. A PO file consists of several entries, each containing a unique source and target pair. Each typical entry consists of the following items:

- **msgid:** The source string that needs to be translated and is usually in English for software applications (Game, View, Control, etc.).
- **msgstr:** The translation of the text present inside double quotes in the **msgid** entry (Spiel, Ansicht, Steuerung, etc.).

[3]For detailed information on PO files, please refer to this blog post. http://pology.nedohodnik.net/doc/user/en_US/ch-poformat.html

[4]https://www.gnu.org/software/gettext/

[5]Screenshot from https://gitlab.gnome.org/GNOME/aisleriot/-/blob/master/po/de.po#L1726. Accessed Date: 09/02/2023 (dd/mm/yyyy)

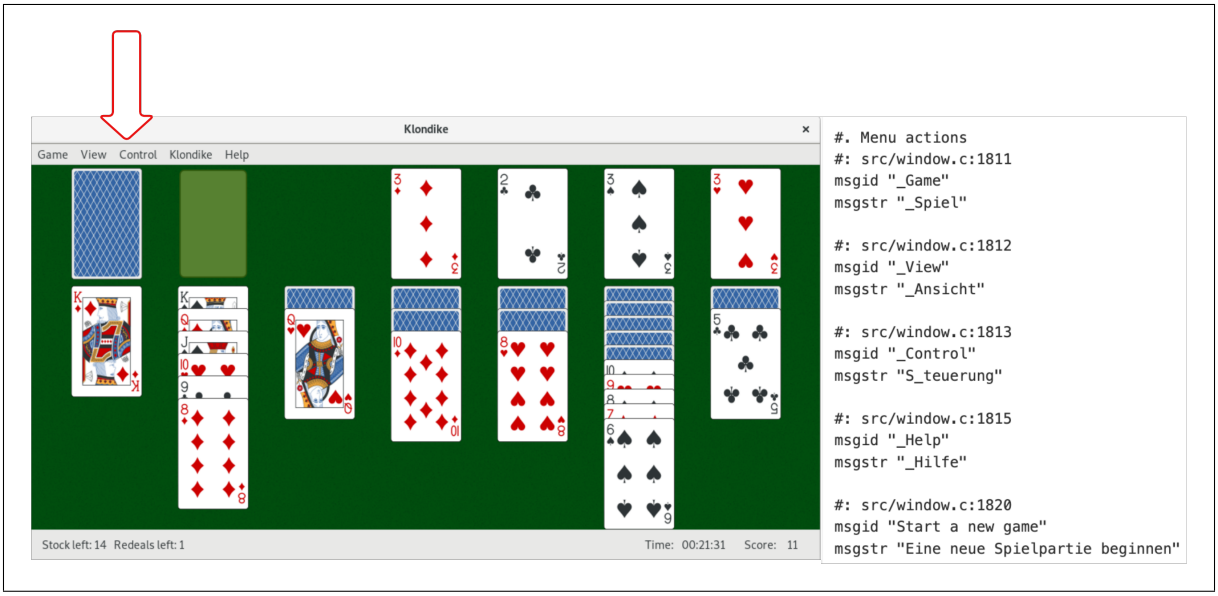

Figure 1: Screenshot from Klondike[5](Solitaire) game in GNOME with UI texts (LEFT). The German translations of the texts in the menu bar (RED ARROW) with their corresponding translations in the PO file (RIGHT)

- **Source Reference**[6]**:** Comment above each entry indicating where the texts are extracted from including the file name and line number (*#: src/window.c:1811*). Translators can look up the source code for more context. Note that this can be more than one file and can include multiple references.

2.2 Contextual Information in PO

PO files contain more information than the source and target sentence pairs. However, it is not straightforward how and what information to include in the NMT models. Moreover, it is unlike the contextual information in the traditional sense to address phenomena like co-reference resolution or gender (Stojanovski and Fraser, 2019; Wong et al., 2020; Lopes et al., 2020). Below, we describe three sources of information that are relevant and useful for UI translation:

Domain: All segments in a PO file belong to one application or software. Hence, we can loosely classify these segments belonging to finer-grained domains containing specific properties. Furthermore, new UI texts added to the software should maintain consistency according to terms and phrasing with previous translations. Therefore, knowing that the sentence is from a particular PO file can help NMT choose a similar translation style.

Neighbours: The order of the entries in the PO file maintains some relation to what users see when using the software. Although there is no semantic relation between the entries in the document, surrounding segments can provide information about the current entry. For example, consider a scenario where we must translate the word "Home" into German. If we know that the surrounding two sentences are "Change Directory" and "Print Working Directory," we can infer that it is in the context of a Linux application. Hence, we can translate it as "Benutzerverzeichnis" by inferring information from other entries/neighbors.

Source Reference: Specific to PO files, source reference can provide high-quality information. Almost all entries have a reference to the source code from which it was extracted. Human translators often look at the source code in case of ambiguity. NMT models can also exploit such information while predicting. For translating "Login" to German, we need to decide on choosing the noun (title text) or verb form (button). If all the texts having the same source reference were translated as the verb form, it is more likely that the current word is also a verb form. Therefore, source references can also be beneficial to integrate into NMT models.

Developers can also add context to each entry in the form of comments or by adding a **msgcxt** in the entry. We illustrate such occurrences with the toy

[6]**Source Reference** is the file path in the source code for a source-target sentence pair. We follow the convention and clarify that it is unrelated to the source or reference texts.

example from the blog post[3]:

```
#. TRANSLATORS: First letter in 'Scope'
#: tools/observinglist.cpp:700
msgid "S"
msgstr ""

#: tools/observinglist.cpp:700
msgctxt "First letter in 'Scope'"
msgid "S"
msgstr ""
```

Note that there is no standard or consistent process on how developers provide context to the translators. For example, it can be the case that a comment for an entry also applies to the subsequent entries. However, it is trivial for humans to determine which entries are related to the comment. Also, some entries do not have any comments or *msgcxt*. Furthermore, this information is in natural language needing complex models to use this information. Therefore, it is not straightforward to leverage the comments/*msgcxt* as a context in an NMT model, and we do not include this information while conducting our baselines in Section 4.

One possible workaround is to use the actual source code using the source reference entry to provide more information. Extracting contextual information from code can save developers time annotating and enable the NMT model to translate a wide range of applications.

Another category of contextual metadata that could be beneficial for translating UIs with NMT models is the element type of the text (button, title, table item, etc.,). However, this is rarely present in open-source data, and developers do not explicitly mention such information in PO files usually. Moreover, there is also no consistency in this case to specify such information.

2.3 Multilingual UI Corpus

Although we have shown that PO files are useful, there is no large corpus containing these files. Having UI texts from a limited amount of applications does not cover different domain-specific choices or styles. Collecting UI data from as many different domains as possible is crucial. Therefore, we create the UI corpus by searching for PO files in public GitHub repositories.

By convention, developers name the PO file with the language code and *.po* extension. For example, the name of PO files in German is **de.po | De.po | DE.po | dE.po**. Therefore, we search and download files ending with such names depending on the language in publicly available repositories. We use Sourcegraph to query the repositories and download PO files for multiple languages. Afterward, we parse the PO files using the polib library and extract translation pairs with the contextual information described in Section 2.2. Table 1 shows the number of source-target pairs we extracted for different languages.

Language	Repositories	Total Sentence Pairs
German	22248	1.33M
Japanese	13196	0.89M
Spanish	20996	1.38M
Hindi	3095	128K

Table 1: Total number of repositories and parallel sentence pairs extracted between English ↔ German, Japanese, Spanish or Hindi. Note that PO files for many more languages are often present and can be scraped. We provide links along with commit hashes (to replicate our data set) to download the corpus from public Github repositories.

Although we extract bilingual translation data, it is possible to create a multi-way UI data set using repositories containing PO files for multiple languages. Therefore, such data can also be fruitful in improving Multilingual NMT for non-English translation directions (Aharoni et al., 2019; Liu et al., 2020).

3 Realistic & Targeted Evaluation

The previous section described how we collected the data and its characteristics. The next step is to design a realistic evaluation framework to showcase challenges in UI translation and estimate the NMT model's quality reliably.

In this section, first, we propose two test sets considering different use cases while evaluating. Then, to highlight one fundamental phenomenon when dealing with UI strings, we create a challenge set for disambiguation based on heuristics and human annotation. We explain how we curated the different test sets below.

3.1 Intra & Cross-application Test Sets

Splitting the data appropriately into train/dev/test sets is vital in evaluating NMT models. Furthermore, the splits should also capture realistic condi-

tions. Therefore, we propose the following scenarios:

Intra-application: The software gets updated regularly. With the updates, we need to translate newly added and modified strings. However, we have access to human-annotated translations for the strings in the previous version providing accurate information about the domain. Therefore, we consider a case where the test set is from applications that the model has seen during training. From the training data, we randomly select samples and discard them from our training set.

Cross-application: Localization companies also need to translate new applications for which there is no gold human-annotated data. To imitate these scenarios, we keep out a few applications entirely for evaluation. Instead of randomly selecting for evaluation, we choose 10 applications containing between 100 and 300 translation pairs. We set these bounds to maximizing the number of domains covered during testing.

Both test sets also provide insights into how much contextual information is consistent across applications. For example, suppose the context-aware NMT system performs better than sentence-level NMT on the Intra and not the Cross-application test set. In that case, we can conclude that such information does not generalize for new applications. However, it can be used only to improve translations for applications already present in the training data.

3.2 Targeted Disambiguation Test Set

The test sets described above assess the model's ability in different scenarios. However, it does not explicitly measure the model's ability to capture context. The need for targeted test sets (Bawden et al., 2018; Voita et al., 2018; Stojanovski et al., 2020) to use context was shown in Doc-NMT. Several complicated architectures were proposed for integrating context using limited amounts of document-level annotated data and showed improvements on standard test sets. However, proper regularization parameters on the sentence-level model led to the same performance improvements (Li et al., 2020; Sun et al., 2022). Hence, it is necessary to evaluate context-aware NMT models on targeted test sets to not draw false conclusions.

In the case of UI texts, a prevalent phenomenon where context is needed is disambiguation. Long sentences often contain enough context for the models to generate the translation correctly. However, a large number of short sentences are present in UI data (refer to Appendix A.3). Therefore, we create a targeted disambiguation test set when translating English to German as a benchmark to evaluate context-aware NMT models.

We use heuristics to extract and filter source sentences with multiple target translations. Then, we manually evaluate these instances and filter those where the context is insufficient, or it is unnecessary as the translations are paraphrases. We explain the process in the following.

3.2.1 Automatic Filtering

After lower-casing the data, we extract source sentences with multiple target translations and ended up with numerous pairs ($\sim 50k$) consisting of many paraphrases. Therefore, to only extract disambiguation pairs, we perform the following steps:

1. We only keep word-word translations. This step was necessary to filter out paraphrases using a database later.

2. Translations from the same PO files are more likely to be paraphrased as they belong to the same domain. Hence, we only keep where target translation occurs across different files.

3. The source-target pairs that occur only once were filtered out as they might be noise from incorrect translations.

4. We discard pairs consisting of characters such as "#,,_,!" (IP-address v/s IPaddress) by checking for punctuation symbols.

5. We match with the German paraphrase database[7] (Ganitkevitch et al., 2013; Ganitkevitch and Callison-Burch, 2014) to eliminate target translations that are synonyms. We only keep target words that occur in the database but are not considered paraphrases. We do not set a threshold score and consider all the entries in the database to have high precision.

After several steps of filtering, we end up with 293 segment of pairs. Upon manual inspection, we find it impossible to disambiguate several instances, even when we look at the PO files for

[7]Note that using the multilingual paraphrase database (Ganitkevitch and Callison-Burch, 2014), it is possible to extend this approach into several languages

Context Source	Context Target	Source Word	Target Word
%s was written onto %s Select Image	%s wurde auf %s geschrieben Abbild auswählen	Image	**Abbild**, Bilder
Journal Author Note	Zeitschrift Autorhinweise	Volume	**Volumen**, Band
Vendor Locations Variant Count	Lagerorte der Lieferanten Variantenanzahl	Volume	Volumen, **Band**
You are not currently subscribed to any active threads There are no active author subscriptions.	Sie haben momentan keine aktiven Diskussionen abonniert Es gibt keine aktiven Abonnements nach Autor.	Thread	**Diskussion**, Faden

Table 2: Examples from the targeted disambiguation test set. We only show 2 surrounding entries in the table but the human annotators were shown 5 to provide more information. The actual translation is highlighted in **bold**.

context. In some cases, it is not clear if the translation of the text is a noun or verb, even by looking at the whole file. Although it is possible that looking at the source code may prove beneficial, we ignore these pairs. Moreover, words like "Settings" and "Preferences" are not paraphrased according to the database but are usually synonymous in the UI context. To eliminate such occurrences, we perform a final step of manual filtering and clean the data using native speakers.

3.2.2 Manual Annotation

We split the test set into three parts. Then, we send each part to two annotators for the final filtering. We gave the annotators the surrounding 5 translation pairs (context) and the source word. Given this information, they were given multiple target translations as options, and we asked them to select the appropriate target given the source and context. Furthermore, we allow them to choose multiple options if they believe that more than one translation is appropriate. Moreover, we also provide a "None of the above" option for every question if they think the context is insufficient or the translations are incorrect. Finally, we selected the ones where both annotators chose the same target word as translation. After annotation, we had 95 entries in the final disambiguation test set. We present a few examples from the test set in Table 2.

Note that many pairs were discarded due to insufficient context but it might be possible to disambiguate with screenshots of application showing the specific UI element. However, we do not have text-image aligned data but it shows the potential advantage of Multi-Modal NMT (Elliott et al., 2017). An overview of the sizes of the data splits for our English $\leftrightarrow$ German experiments can be found in Appendix Table 7.

4 Baselines

To evaluate the current NMT approaches with contextual information on UI translations, we provide baselines (English $\rightarrow$ German) using the current SOTA techniques. We build context-aware NMT models on our UI corpus and measure their ability to use context and highlight their shortcomings.

First, we describe how we integrate multiple types of contextual information in NMT. Then, we present the results and analyze the performance of the models on the multiple test sets.

4.1 Context-aware NMT models

As shown in Section 2.2, we have access to different sources of contextual information. The next step is to integrate them into the NMT model. We perform experiments by including them in the following ways:

Domain: We want to exploit the fact that we have already seen a particular PO file. Therefore, we take the help of tags to indicate this information similar to Kobus et al. (2017). We assign a unique domain tag to each repository and prepend it to each source sentence in a file. During training, we initialize each tag with a random embedding. While testing, the model can know the origin of the file by looking at the domain tag to translate appropriately. If it is from an unseen repository, we assign a random embedding
Example: *<KLONDIKE> _Game ||| _Spiel*

Neighbours: In this scenario, we consider the whole PO file as a document. Knowing the surrounding entries in the document can provide cues about the current segment. Therefore, we exploit them by following approaches in Doc-NMT (Tiedemann and Scherrer, 2017). We concatenate the source with the surrounding (left and right if possible) two segments in the file. Note that we only add the source context and not the target side.

Example: *_View _Control <TAG> _Game ||| _Spiel*

Source Reference: Specific to PO files, file paths of the string in the source code can capture standard naming conventions across different applications. Like the neighbor's approach above, we concatenate the path with the source sentence. However, we remove the punctuation to maximize the overlap across paths having similar naming structures.
Example: *src window c <TAG> _Game ||| _Spiel*

4.2 Experimental Setup

Note that we initially pre-train the models on WMT 14 English-German data (Luong et al., 2015) and then fine-tune on our corpus. We perform this two-step training following the success in transfer learning (Zoph et al., 2016; Kocmi and Bojar, 2018) and improve the translation capabilities. The pre-processing and training parameters for both stages is described in Section A.1.

4.3 Results

We present results on the different evaluation scenarios described in Section 3. First, to understand the performance of context-aware models in realistic scenarios, we report the results on *Intra* and *Cross* test sets in Table 3. Then, we evaluate the models on the targeted disambiguation test set to highlight the need for context in this challenging task and present scores in Table 4. Instead of BLUE (Papineni et al., 2002) or CharacTER (Wang et al., 2016), we report the accuracy for this test set as both source and target are word-translations. The scores for $De \rightarrow En$ demonstrate the ability to map different German words to the same English word. However, it is not completely accurate as the English translations that the model generates can be paraphrased.

Do context-aware NMT models perform better where training data is from known applications? All the context-aware models in Table 4 obtain higher score than the sentence-level model on the *Intra-application* test set. The only exception being the *Neighbour* approach in English $\rightarrow$ German on BLEU (49.21 ± 0.26) but par with the standard NMT baseline (49.03 ± 0.76).

We hypothesize that the model knows how to use the context when translating segments from the *Intra-applicaton* test set. For example, consider the *Source Reference* approach and the file in Figure 1. By knowing that the text "*Control*" present in "*src/window.c*" was translated as a noun ("*Steuerung*") and not a verb ("*Steuern*"), the model knows how to translate ambiguous texts in that file. Such phenomena are also possible when using the other context-aware NMT approaches. Therefore, we need to integrate contextual information when translating new segments from seen applications.

Does the contextual information generalize to entirely new software? Contrary to the gains observed above, all the context-aware models do not obtain significant improvements on the *Cross* test set. Although there is a slight improvement in a few conditions, it is around 0.5 in BLEU and 0.4% in CharacTER. Moreover, in the case of *Domain Tag* approach, all the texts are assigned a new random domain tag that was not seen during training. Hence, it performs significantly worse than the other methods.

The results show that the contextual information does not generalize well to entirely new applications. Therefore,

we need better context-aware NMT approaches than the baselines we proposed, which capture context more abstractly and benefits software localization in realistic conditions.

How well do the context-aware NMT models perform on the challenging targeted disambiguation test set? While the sentence-level NMT model obtains an accuracy of 20.0% on the targeted test set, the *Domain Tag* approach reaches 41.0%. Furthermore, it performs the best out of all other approaches, with the next best model reaching only 29.4%.

However, it can be the case that most segments in the disambiguation test belong to applications present in the training data. Hence, allowing the *Domain Tag* approach to perform better. Nevertheless, if the example is from a cross-application, this approach can perform worse due to the random initialization of the tag. Moreover, the other approaches are better than the sentence-level model by only a maximum of 10%. The proposed baseline methods are insufficient in this challenging test set and call for better context-aware NMT models.

Are the context-aware NMT models ignoring the context? We also need to investigate the influence of context, if at all, on the model's translations. If the system uses the context, providing false/incorrect context to the model should hurt the performance. Otherwise, we can conclude that the current NMT model ignores the contextual in-

Context Type	*Evaluation Setup*							
	BLEU (↑)				*CharacTER (↓ %)*			
	Cross		Intra		Cross		Intra	
	En → De	De → En	En → De	De → En	En → De	De → En	En → De	De → En
No context	47.15 ± 0.12	57.14 ± 0.44	49.03 ± 0.76	56.05 ± 0.29	37.04 ± 0.08	29.51 ± 0.02	34.75 ± 0.33	29.18 ± 0.32
Domain Tag	44.99 ± 0.26	54.59 ± 0.36	**52.10 ± 0.06**	**57.90 ± 0.45**	39.08 ± 0.07	30.55 ± 0.18	**31.64 ± 0.17**	**26.94 ± 0.27**
Neighbour (2)	46.93 ± 0.06	57.30 ± 0.27	49.21 ± 0.26	**56.92 ± 0.38**	36.94 ± 0.12	**29.13 ± 0.16**	**34.04 ± 0.21**	**28.22 ± 0.11**
Source Reference	**47.79 ± 0.09**	57.82 ± 0.33	**51.09 ± 0.33**	**58.53 ± 0.35**	**36.67 ± 0.14**	**29.12 ± 0.04**	**32.38 ± 0.41**	**27.00 ± 0.20**

Table 3: Baseline experiments using different sources of contextual information described in Section 2.2 and evaluated on both Intra and Cross application scenarios. We perform each experiment 3 times to account for randomness and report using both BLEU (↑) and CharacTER % (↓) metrics. We highlight the score in **bold** if the gains are statistically significant compared to the baseline model (*No context*)

	Accuracy (%)	
Context Type	En → De	De → En
None	21.05 ± 0.85	43.85 ± 0.49
Domain Tag	**41.03 ± 0.85**	**56.46 ± 2.14**
Neighbour (2)	26.26 ± 1.5	45.93 ± 1.79
Source Reference	30.8 ± 0.98	52.23 ± 1.79

Table 4: Accuracy of context-aware NMT models on targeted disambiguation test set ($En \leftrightarrow De$). The scores from $De \leftrightarrow En$ denote the accuracy in predicting the same English word given different German words. We run the experiment 3 times and report the confidence intervals.

formation as shown in previous works (Sun et al., 2022).

For this purpose, we conduct an experiment using the "*Neighbors*" approach and provide incorrect context by randomly sampling consecutive segments from different PO files. We report the results in Table 5. Although it is not better than a simple NMT system trained without context, there is always a drop in performance when using incorrect context. Thus, the model uses the context for generating translations but with degradation in general translation quality.

To show the role of context, we provide an example in our test sets in Table 6. The word "*driver*" in this case can be translated to German as either "*Fahrer*" (vehicle driver) or "*Treiber*" (software driver), depending on the domain. In both cases, the model correctly predicts the translation using the context. However, the scores do not show overall improvement, indicating that the context might add additional noise, causing a drop in general translation quality.

5 Related Work

Domain-Adaption in NMT: NMT models for software localization have to deal with texts coming from vast amount of domains. Hence, Domain-Adaptation techniques (Saunders, 2022) are highly relevant in this challenging task. Many works use different domain-tagging schemes to indicate the type of data where the segment's labels are either known or unknown (Kobus et al., 2017; Poncelas et al., 2018; Mino et al., 2020; Wang et al., 2021). Another line of approach is to modify the network to indicate and generate domain-specific translations (Zeng et al., 2018; Pham et al., 2021; Lin et al., 2021). Building on this, few works also propose to add and adapt using domain-specific parameters (Bapna and Firat, 2019; Abdul-Rauf et al., 2020).

Document-level NMT: Exploiting the context from the surrounding texts is necessary for both software localization and Doc-NMT. The initial straightforward approaches for Doc-NMT explored concatenating the source/target with surrounding sentences (Tiedemann and Scherrer, 2017; Agrawal et al., 2018; Ma et al., 2020). These methods do not change the NMT architecture or training methods but can be applied by simply altering the data or the embedding. In contrast, several works investigated creating a different representation for the context sentences using additional encoders (Jean et al., 2017; Zhang et al., 2018; Voita et al., 2018; Werlen et al., 2018; Wang et al., 2019a). Few recent works also propose either modifying the architecture of augmenting data to show the benefit of Document-Document translation (Bao et al., 2021; Sun et al., 2022). For targeted evaluation, Gonzales et al. (2017) also manually constructs a word-sense disambiguation test in English ↔ German. However, our test set is mostly automatic and calls for

Evaluation Setup	True Context		False Context		No Context	
	En → De	De → En	En → De	De → En	En → De	De → En
Intra	49.21 ± 0.26	56.92 ± 0.38	47.95 ± 0.28	55.0 ± 0.35	49.03 ± 0.76	56.05 ± 0.29
Cross	46.93 ± 0.06	57.30 ± 0.27	46.38 ± 0.22	55.94 ± 0.08	47.15 ± 0.12	57.14 ± 0.44

Table 5: BLEU scores for the "*Neighbor (2)*" model evaluated using True and False context during inference. For "*No Context*", we report the scores of the baseline model trained simply on the parallel data without any additional information. We report the confidence interval by running the experiment 3 times.

Source	Current Driver
Reference	Aktueller Fahrer
Source + True Context	Create a new vehicle status <tag> Date when the vehicle has been immatriculated <tag> Current Driver
Neighbour (2) Hypothesis	Aktueller Fahrer
Source + False Context	% s updates a cartridge <tag> Unknown file <tag> Current Driver
Neighbour (2) Hypothesis	Aktueller Treiber

Table 6: Example from our test set where the Neighbour approach (Table 3) uses the context and generates the proper translation. *Source + True context* is the concatenation of the true neighbors in the same PO file with the source sentence whereas *Source + False Context* appends a random context from another PO file.

solving the disambiguation by focusing on neighboring entries and not the sentence itself.

NMT for UI: One of the major limitations for building NMT for UI is the lack of data. The OPUS corpus (Tiedemann, 2012) makes such data available but with no contextual information. The closest to our work is Wang et al. (2019b), building NMT for mobile applications. However, the data is neither public nor addresses the several contextual issues occurring in software localization.

6 Conclusion

We presented a multilingual UI corpus with additional meta-information for researchers in the community to exploit and build context-aware NMT models for software localization. We also have proposed two evaluation setups to replicate conditions occurring in localization companies and show the difficulty in tackling new applications. Furthermore, we experiment using domain adaptation and Doc-NMT techniques to provide baselines and present the benefit of using different types of context in intra-applications while showing its ineffectiveness in cross-application scenarios. Moreover, we suggest an automatic procedure to create a targeted disambiguation test set where context is necessary to generate the correct target word translation. Finally, we show that the baseline systems fail in such challenging settings and call for sophisticated context-aware NMT models to improve the process of software localization.

7 Limitations

While we have presented UI data with contextual information, it does not contain any meta-data (button, table title, etc.,) of the textual elements. Having access to such resources can prove highly beneficial and are lacking in our presented data set. Furthermore, we do not provide any visual context to develop Multi-Modal NMT systems that implicitly contain the meta-data. However, we consider them as potential directions and include them in our future work. Finally, we do not analyze or experiment on non-European languages that might contain specific and unique properties which need to be addressed during localization. Another important limitation is the quality of the translations. We don't know whether they have been produced by professional translators or by multilingual speakers creating inaccurate translations. Although we filter the fuzzy (e.g MT outputs) translations, it can be that not all of them are marked completely. Therefore, qualitative analysis of the data is further necessary.

Acknowledgement We thank Christian Lieske and Jens Scharnbacher for providing insights into

the localization process. We also thank Fabian Retkowski, Christian Huber, and Leonard Bärmann for annotating and making it possible to create the disambiguation test set.

References

Sadaf Abdul-Rauf, José Carlos Rosales, Minh Quang Pham, and François Yvon. 2020. Limsi@ wmt 2020. In *Conference on Machine Translation*.

Ruchit Rajeshkumar Agrawal, Marco Turchi, and Matteo Negri. 2018. Contextual handling in neural machine translation: Look behind, ahead and on both sides. In *21st annual conference of the European association for machine translation*, pages 11–20.

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Guangsheng Bao, Yue Zhang, Zhiyang Teng, Boxing Chen, and Weihua Luo. 2021. G-transformer for document-level machine translation. In *ACL/IJCNLP (1)*.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548.

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating discourse phenomena in neural machine translation. In *16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1304–1313.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–391, Vancouver, Canada. Association for Computational Linguistics.

Desmond Elliott, Stella Frank, Loïc Barrault, Fethi Bougares, and Lucia Specia. 2017. Findings of the second shared task on multimodal machine translation and multilingual image description. In *Proceedings of the Second Conference on Machine Translation*, pages 215–233.

Raymond Flournoy and Christine Duran. 2009. Machine translation and document localization at adobe: From pilot to production. *MT Summit XII: proceedings of the twelfth Machine Translation Summit*, pages 425–428.

Juri Ganitkevitch and Chris Callison-Burch. 2014. The multilingual paraphrase database. In *LREC*, pages 4276–4283. Citeseer.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764.

Annette Rios Gonzales, Laura Mascarell, and Rico Sennrich. 2017. Improving word sense disambiguation in neural machine translation with sense embeddings. In *Proceedings of the Second Conference on Machine Translation*, pages 11–19.

Sebastien Jean, Stanislas Lauly, Orhan Firat, and Kyunghyun Cho. 2017. Does neural machine translation benefit from larger context? *arXiv preprint arXiv:1704.05135*.

Catherine Kobus, Josep M Crego, and Jean Senellart. 2017. Domain control for neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 372–378.

Tom Kocmi and Ondřej Bojar. 2018. Trivial transfer learning for low-resource neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 244–252.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180.

Bei Li, Hui Liu, Ziyang Wang, Yufan Jiang, Tong Xiao, Jingbo Zhu, Tongran Liu, and Changliang Li. 2020. Does multi-encoder help? a case study on context-aware neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3512–3518.

Huan Lin, Liang Yao, Baosong Yang, Dayiheng Liu, Haibo Zhang, Weihua Luo, Degen Huang, and Jinsong Su. 2021. Towards user-driven neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4008–4018.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

António Lopes, M Amin Farajian, Rachel Bawden, Michael Zhang, and André FT Martins. 2020. Document-level neural mt: A systematic comparison. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 225–234.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Shuming Ma, Dongdong Zhang, and Ming Zhou. 2020. A simple and effective unified encoder for document-level machine translation. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 3505–3511.

Sameen Maruf, André FT Martins, and Gholamreza Haffari. 2019. Selective attention for context-aware neural machine translation. In *Proceedings of NAACL-HLT*, pages 3092–3102.

Sameen Maruf, Fahimeh Saleh, and Gholamreza Haffari. 2021. A survey on document-level neural machine translation: Methods and evaluation. *ACM Computing Surveys (CSUR)*, 54(2):1–36.

Hideya Mino, Hideki Tanaka, Hitoshi Ito, Isao Goto, Ichiro Yamada, and Takenobu Tokunaga. 2020. Content-equivalent translated parallel news corpus and extension of domain adaptation for nmt. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3616–3622.

Víctor Muntés Mulero, Patricia Paladini Adell, Cristina España Bonet, and Lluís Màrquez Villodre. 2012. Context-aware machine translation for software localization. In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation: EAMT 2012: Trento, Italy, May 28th-30th 2012*, pages 77–80.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Minh Quang Pham, Jitao Xu, Josep M Crego, François Yvon, and Jean Senellart. 2020. Priming neural machine translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 516–527.

MinhQuang Pham, Josep Maria Crego, and François Yvon. 2021. Revisiting multi-domain machine translation. *Transactions of the Association for Computational Linguistics*, 9:17–35.

Alberto Poncelas, Dimitar Shterionov, Andy Way, Gideon Maillette de Buy Wenniger, and Peyman Passban. 2018. Investigating backtranslation in neural machine translation. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation: 28-30 May 2018, Universitat d'Alacant, Alacant, Spain*, pages 249–258. European Association for Machine Translation.

Danielle Saunders. 2022. Domain adaptation and multi-domain adaptation for neural machine translation: A survey. *Journal of Artificial Intelligence Research*, 75:351–424.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 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)*, pages 1715–1725.

Raivis Skadiņš, Mārcis Pinnis, Andrejs Vasiljevs, Inguna Skadiņa, and Tomáš Hudík. 2014. Application of machine translation in localization into low-resourced languages. In *Proceedings of the 17th Annual conference of the European Association for Machine Translation*, pages 209–216.

Dario Stojanovski and Alexander Fraser. 2019. Improving anaphora resolution in neural machine translation using curriculum learning. In *Proceedings of Machine Translation Summit XVII: Research Track*, pages 140–150.

Dario Stojanovski, Benno Krojer, Denis Peskov, and Alexander Fraser. 2020. Contracat: Contrastive coreference analytical templates for machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4732–4749.

Zewei Sun, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Shujian Huang, Jiajun Chen, and Lei Li. 2022. Rethinking document-level neural machine translation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3537–3548.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Lrec*, volume 2012, pages 2214–2218. Citeseer.

Jörg Tiedemann and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*. The Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

E Voita, P Serdyukov, I Titov, and R Sennrich. 2018. Context-aware neural machine translation learns anaphora resolution. In *ACL 2018-56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, pages 1264–1274.

Ke Wang, Guandan Chen, Zhongqiang Huang, Xiaojun Wan, and Fei Huang. 2021. Bridging the domain gap: Improve informal language translation via counterfactual domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13970–13978.

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. 2016. Character: Translation edit rate on character level. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 505–510.

Xinyi Wang, Jason Weston, Michael Auli, and Yacine Jernite. 2019a. Improving conditioning in context-aware sequence to sequence models. *arXiv preprint arXiv:1911.09728*.

Xu Wang, Chunyang Chen, and Zhenchang Xing. 2019b. Domain-specific machine translation with recurrent neural network for software localization. *Empirical Software Engineering*, 24(6):3514–3545.

Lesly Miculicich Werlen, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-level neural machine translation with hierarchical attention networks. In *EMNLP*.

KayYen Wong, Sameen Maruf, and Gholamreza Haffari. 2020. Contextual neural machine translation improves translation of cataphoric pronouns. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5971–5978.

Jitao Xu, Josep M Crego, and Jean Senellart. 2020. Boosting neural machine translation with similar translations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1580–1590.

Jiali Zeng, Jinsong Su, Huating Wen, Yang Liu, Jun Xie, Yongjing Yin, and Jianqiang Zhao. 2018. Multi-domain neural machine translation with word-level domain context discrimination. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 447–457.

Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. 2018. Improving the transformer translation model with document-level context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 533–542.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575.

A Appendix

A.1 Pre-processing and Training Parameters

For pre-processing, we first tokenize the data with the Moses Tokenizer (Koehn et al., 2007). Then, we learn a joint sub-word vocabulary from $30k$ merge operations using BPE (Sennrich et al., 2016) and apply on the data.

For experiments in Table 4, we first pre-train the model using WMT14 English-German data (Luong et al., 2015). We use the standard Transformer model (Vaswani et al., 2017) with 6 encoder and decoder layers. We use 4 attention heads for every layer an embedding dimension of 512. We use 0.1 for label-smoothing 0.2 for dropout. We set max-tokens to 3000 and a learning rate of 0.0001. For the experiments, we use the Fairseq toolkit (Ott et al., 2019) and set all other parameters to default. While fine-tuning, we reload the model and continue using the same optimization and regularization parameters.

A.2 Data Split Overview

Data Split	*Number of Sentence Pairs*
Train	1.26M
Intra	2.6K (2.6K)
Cross	2.6K (2.4K)
Disambiguation	95

Table 7: Overview of UI data split for English $\leftrightarrow$ German. () indicates the number of validation sentences for the Intra and Cross application test sets.

A.3 Characteristricts of UI segments

	Average Sentence Length English	Average Sentence Length German	Vocabulary Size English	Vocabulary Size German
UI	36	43	167375	321602
News Domain	142	157	626914	1444840

Table 8: Comparing UI (Git scraped) v/s WMT 14 English-German Data. It can be seen the UI sentences are much shorter than the data available in the general domain. Also in both data types, we see that the German sentences consist of more unique words due its complex morphology