MeMAD

Methods for Managing
Audiovisual Data

memad.eu
info@memad.eu

Twitter – @memadproject
LinkedIn – MeMAD Project

MeMAD Deliverable

D4.3 Tools and Models for Multimodal, Multilingual and Discourse-Aware Machine Translation

Grant agreement number	780069
Action acronym	MeMAD
Action title	Methods for Managing Audiovisual Data: Combining Automatic Efficiency with Human Accuracy
Funding scheme	H2020–ICT–2016–2017/H2020–ICT–2017–1
Version date of the Annex I against which the assessment will be made	3.10.2017
Start date of the project	1.1.2018
Due date of the deliverable	31.03.2020
Actual date of submission	16.03.2020
Lead beneficiary for the deliverable	University of Helsinki
Dissemination level of the deliverable	Public

Action coordinator's scientific representative
Prof. Mikko Kurimo
AALTO–KORKEAKOULUSÄÄTIÖ, Aalto University School of Electrical Engineering, Department of Signal Processing and Acoustics
mikko.kurimo@aalto.fi

MeMAD project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 780069. This document has been produced by theMeMAD project. The content in this document represents the views of the authors, and the European Commission has no liability in respect of the content.

Authors in alphabetical order		
Name	Beneficiary	e-mail
Jorma Laaksonen	Aalto University	jorma.laaksonen@aalto.fi
Umut Sulubacak	University of Helsinki	umut.sulubacak@helsinki.fi
Jörg Tiedemann	University of Helsinki	jorg.tiedemann@helsinki.fi

Internal reviewers in alphabetical order		
Name	Beneficiary	e-mail
Sabine Braun	University of Surrey	s.braun@surrey.ac.uk
Tiina Lindh-Knuutila	LLS	tiina.lindh-knuutila@lingsoft.fi

Abstract

In this deliverable, we report on our final releases of machine translation models and tools that were developed through our efforts in WP4.

We introduce OPUS-MT, a WebSocket-based translation server, and our release of the MeMAD subtitle translation pipeline, both including pre-trained models suitable for general-purpose translation. Next, we introduce and discuss the `subalign` toolbox, and its key utilities that implement heuristics to convert between plain sentences and SRT-formatted subtitle segments with time codes. In connection with this, we introduce fine-tuned models for subtitle translation with the capability of token alignment for improved synchronisation. Afterwards, we introduce our releases of the MeMAD image caption translation and end-to-end speech translation systems. These systems are based on our work on multimodal machine translation discussed previously in D4.1, evaluated as part of our submissions to the WMT 2018 multimodal translation and IWSLT 2019 speech translation shared tasks, respectively. Furthermore, we also introduce our release of the MeMAD document-level translation models, which were developed through our experiments on discourse-aware machine translation, and evaluated as part of our submission to the WMT 2019 document-level translation shared task. Finally, we also describe our release of a dataset tailored for benchmarking document-level machine translation performance.

All of our software releases are open source with permissive licences of use, and our pre-trained models have been made freely available for download following the guidelines for open access. Our scripts and documentation have been organised into individual repositories located in the common MeMAD Github space, linking to the relevant pre-trained models (where applicable) hosted in the MeMAD community space on Zenodo. We include additional explanations and usage instructions in this deliverable as necessary.

Contents

1 Introduction

The purpose of this document is to collect information about released models, resources and tools in connection with the activities in WP4 on multilingual, multimodal and discourse-aware machine translation (MT) within the MeMAD project. In particular, the releases build on the work described in deliverables D4.1 (report on multimodal machine translation) and D4.2 (report on discourse-aware machine translation for audio-visual data). For convenience, we link the essential outcomes from a dedicated repository on MeMAD workpackages, which is available from GitHub.[1] Here, we will not repeat research and results described in various publications and deliverables that are connected with the models and their development but rather provide links to the released resources and brief information about their use. Nevertheless, we add findings and results that have not been reported before including benchmarks on project-internal test sets and multilingual translation models that have been prepared for this deliverable.

In particular, we include resources for the following categories:

- **General-purpose translation**: Our efforts in developing general-purpose MT models for the focus languages of the MeMAD project trained on large datasets.
- **Subtitle translation**: Tools for translating subtitles, including the extraction of appropriately segmented text, and the alignment of translations to given time frames.
- **Image caption translation**: Multimodal MT models optimised for the translation of image captions, as the outcome of our efforts in the WMT 2018 shared task.
- **Spoken language translation**: Multimodal MT models for the translation of speech to text in another language, connected to our submissions to IWSLT shared tasks.
- **Discourse-aware translation**: Document-level MT models that process an extended context to capture discourse-level dependencies across sentence boundaries.

In addition to the models that we release, we also publish our tools, scripts and pipelines to run these models, as well as information about their training and development. As long as copyrights permit, we also release datasets for further research and replicability.

The general principle for our releases is to provide our resource with permissive licences to enable their wide application. For software and tools we adopt MIT[2] and Apache[3] licences and for the models we focus on a permissive Creative Commons licence (CC BY 4.0).[4] Software tools and documentation are stored on our project space on Github[5] and the released models and other resources are published in our project community space at Zenodo[6].

In the following sections, we list our published material together with brief instructions.

[1] https://github.com/MeMAD-project/workpackages
[2] https://mit-license.org/
[3] https://apache.org/licenses/LICENSE-2.0
[4] https://creativecommons.org/licenses/by/4.0/
[5] https://github.com/MeMAD-project
[6] https://zenodo.org/communities/memad

2 General-purpose NMT

MeMAD focuses on six European languages: Dutch, English, Finnish, French, German and Swedish. Translation is a key component in the project, enabling cross-lingual access to audiovisual information in various workflows. Translation is required in intralingual subtitling, cross-lingual information enrichment, cross-lingual entity linking, and cross-lingual search. The MeMAD prototype implements a modular toolbox for the production use case, and incorporates many of those tasks. Furthermore, multilingual subtitle production outside of that prototype is another use case that we have studied intensively in the project.

Providing the functionality of a general-purpose tool for a translation component has been one of the major efforts in WP4. Here, we emphasise the practical and versatile use of such tools and the coverage of all language pairs included in the project setup. The releases include:

- Pre-trained models for all translation directions on large, mixed datasets with state-of-the-art neural MT.
- Translation server applications for deploying scalable services that can easily be integrated in different workflows.
- Scripts and pipelines for training these models.

2.1 Translation models

Released earlier by the University of Helsinki, the OPUS-MT[7] system (Tiedemann and Thottingal, 2020) includes a collection of publicly available general-purpose MT models. We have integrated the models corresponding to our language pairs of interest into the Limecraft Flow platform, where the majority of MeMAD use case and proof-of-concept evaluations were carried out, as general-purpose MT tools available on demand. In the course of these evaluations, these models have been used to facilitate metadata enrichment, cross-lingual search, and interlingual subtitling, among other things. OPUS-MT models are organised in an auxiliary repository hosted on Github[8], where download links can be retrieved.

We have also released a set of MT models as part of our subtitle translation system release,[9] which contains 30 text-based bilingual models (all possible translation directions among the six MeMAD languages). These models were intended to take part in the subtitle translation pipelines developed for the subtitling productivity evaluations, but follow the same training setting as the OPUS-MT models, and use similar training data. Therefore, the MT models that were enclosed in this release also constitute viable general-purpose MT models. These models[10] are available through the MeMAD community space on Zenodo, with open access under the Creative Commons Attribution CC BY 4.0 licence. For a detailed description of the full pipeline as well as other models tailored for subtitle translation, please refer to Section 3.

Both sets of our general-purpose MT models are based on the transformer (Vaswani et al., 2017) implementation of Marian (Junczys-Dowmunt et al., 2018), an open source neural MT framework. The transformers were set up with 6 encoder and 6 decoder layers, 8 attention

[7] https://github.com/MeMAD-project/Opus-MT

[8] https://github.com/MeMAD-project/Opus-MT-train/tree/master/models

[9] https://github.com/MeMAD-project/subtitle-translation

[10] https://zenodo.org/record/4389209 and https://zenodo.org/record/4556121

heads, and a dropout rate of 0.1. The training procedure uses an Adam optimiser with a learning rate of 0.0003, with linear warmup for the first 16 000 batches, and inverse square root decay through the remaining ones. Decoding for validation during training uses perplexity as the validation metric, and beam search with a beam size of 12. For replicability purposes, the exact training specification can be retrieved from the OPUS-MT models training script available from the corresponding repository on Github.[11]

(BLEU ↑)	∗ → de	∗ → en	∗ → fi	∗ → fr	∗ → nl	∗ → sv
de → ∗	—	25.60	16.07	19.20	21.20	19.20
en → ∗	29.16	—	22.90	27.11	29.86	28.56
fi → ∗	14.08	16.96	—	14.16	16.57	17.78
fr → ∗	21.39	25.28	16.76	—	20.85	19.26
nl → ∗	22.27	27.22	18.79	20.62	—	22.20
sv → ∗	21.21	26.88	21.36	20.21	23.70	—

Table 1: BLEU scores from benchmarking the MeMAD subtitle translation models on the held-out internal development sets sampled from the OpenSubtitles corpus. The scores were calculated each on a random selection of 100 000 sentence pairs, sampled from movies released in even-numbered years between 1970 and 2018 (inclusive), for which the OpenSubtitles 2018 release had data available in all 6 focus languages. Subtitles for movies released in odd-numbered years have been instead used as training data.

The OPUS-MT models were trained using all parallel data available for each corresponding language pair in OPUS (Tiedemann, 2012), a large collection of open parallel corpora for training MT models. Similarly, the training data for the subtitle translation models have been compiled from OPUS, except a small multi-parallel sampling of subtitle data from the OpenSubtitles[12] corpus was excluded from training, and instead held out as an internal development set (see Table 1 for the relevant performance metrics). Since the OPUS collection is always evolving, and the training sets for the two sets of models have been compiled about eight months apart, there should also be other minor differences resulting from OPUS corpora that were added or updated in this window. Furthermore, OPUS-MT contains one model per translation direction, trained until convergence with early stopping, reverting to the best snapshot of the model after 10 consecutive validation cycles in which the model did not improve. In contrast, the subtitle translation pipelines include ensembles of 5 randomly-seeded models for each direction, each model trained equally for 72 hours on 4 parallel Nvidia V100 GPUs.

OPUS-MT is meant to be used as a translation service, and was packaged to facilitate the process of setting up a server (see Section 2.2 for further details and instructions). In contrast, the subtitle translation models were intended to be set up locally, as modules for a pipeline to process SRT-formatted subtitle data. Nonetheless, an option to perform translation on plain text data (segmented as sentences, rather than subtitles) was built into the translation interface. This option can be activated via the `--plain-text-mode` flag, as in the invocation example below:

[11] https://github.com/MeMAD-project/OPUS-MT-train/blob/master/lib/train.mk

[12] https://www.opensubtitles.com

(BLEU ↑)	∗ → de	∗ → en	∗ → fr
de → ∗	—	32.97	29.28
en → ∗	28.41	—	—
fr → ∗	24.31	—	—

Table 2: BLEU scores from benchmarking the MeMAD subtitle translation models on WMT 2020 test sets. Translation models other than de ↔ en and de ↔ fr did not have corresponding test sets in the 2020 release, and previous years of releases were not used for benchmarking due to a partial overlap with the training data.

```
./translate.py --src-lang de \
               --tgt-lang en \
               --input your/data/sample.de \
               --output your/data/sample.en \
               --gpu-devices 4 \
               --verbose \
               --log process.log \
               --plain-text-mode
```

Note that, while this option skips the initial sentence parsing and final subtitle segmentation steps, it still uses the other pipeline modules (i.e. preprocessing, restoration, and postprocessing). All the software dependencies listed in the `subtitle-translation` repository, except for `OpusTools-perl` and `subalign`, must still be installed and configured. Likewise, segmentation, restoration, and translation models must still be downloaded and unpacked as instructed. Tables 1 and 2 show benchmarking results for the subtitle translation models when used in this way, on the held-out development sets, and on the WMT 2020 test sets, respectively.

2.2 Software for building translation services

We have implemented a WebSocket translation service application based on Marian that can be deployed on modern GNU/Linux distributions using the setup published in OPUS-MT. The implementation includes a translation router daemon process that connects an arbitrary number of individual translation services that may contain multilingual as well as domain-specific services using a simple WebSocket API. A simple JSON configuration file can be used to specify the services to be included. Figure 1 shows an example for a configuration with two translation services, one for translating Finnish to English running on the same machine as the translation router (localhost), and another service for translating French to Estonian or Finnish running on a remote machine.

The API can be called using another simple JSON input format specifying the text to be translated, the source language code and the target language code:

```
{
    "localhost:20000" : {
        "source-languages" : "fi",
        "target-languages" : "en"
    },
    "192.168.1.14:21100" : {
        "source-languages" : "fr",
        "target-languages" : "et+fi"
    }
}
```

Figure 1: Translation router configuration.

```
{
    "text": "Mitä kuuluu?",
    "source": "fi",
    "target": "en"
}
```

The source language can also be omitted and the server will in that case try to automatically detect the input language using the Chrome compact language detection library (version 2). However, for short messages, this may not be very reliable. The developers mention that the software is designed for web pages of at least 200 characters and that it is not expected to do well very short text, lists of proper names, part numbers, etc. [13]. Therefore, please, use this option with care.

The system also supports domain-specific models assuming that several task-specific models have been deployed in the backend. The use of alternative models can be activated by adding an attribute "model" to the description of the translation system in order to separate it, for example, from other domains or other task-specified applications. Figure 2 shows and example for two different German-Finnish translation services with one of them optimised for the WMT news translation task.

The model can be specified in the API call by adding the model argument in the request:

```
{
    "text": "Wie geht's?",
    "source": "de",
    "target": "fi",
    "model": "wmt"
}
```

The result of API requests is also formatted in JSON providing various types of output (see Figure 3):

[13] https://github.com/CLD2Owners/cld2

```
{
    "192.168.1.19:20004" : {
        "source-languages" : "de",
        "target-languages" : "fi"
    },
    "192.168.1.12:20008" : {
        "model" : "wmt",
        "source-languages" : "de",
        "target-languages" : "fi"
    }
}
```

Figure 2: Translation router configuration with domain-specific models

- `result`: translation result in plain text
- `source-sentences`: list of source sentences segmented in plain text
- `target-sentences`: list of translated target sentences in plain text
- `source-segments`: list of source sentences segmented as subword units
- `target-segments`: list of translated target sentences segmented as subword units
- `alignment`: cross-lingual token alignments for each sentence (subword unit alignment)
- `source/target/server`: source language, target language, and translation server used

The segmented input and output are handy for the token alignments provided by the model. Note that the token alignment comes from cross-lingual attention, and that a model must be trained with the guided alignment feature of Marian in order to make this information useful for any further processing.

The example in Figure 3 illustrates BPE-segmented model output that has been pre-tokenised as well. For most of our current models, we skip tokenisation and run SentencePiece subword segmentation on raw text instead. All pre- and post-processing will be handled internally in the translation server and should not affect the format of the final result retrieved from the service except for attributes that provide the segmented strings from the internal representations.

2.3 Deploying translation servers

Installing the translation service software is straightforward and has been tested on Ubuntu Linux distributions versions 14.04, 16.04 and 18.04. Here are the main steps to be taken:

- Clone the software and install all pre-requisites

```
git clone https://github.com/Helsinki-NLP/Opus-MT.git
cd Opus-MT/install
```

```
{
    "alignment": [
        "0-0 0-2 1-1 2-3",
        "0-0 1-1 3-2 4-3 5-4"
    ],
    "result": "How are you? The translation is fun.",
    "server": "192.168.1.18:20001",
    "source": "fi",
    "source-segments": [
        "Mit\u00e4 kuuluu ?",
        "K\u00e4\u00e4@@ nn\u00f6@@ s on hauskaa ."
    ],
    "source-sentences": [
        "Mit\u00e4 kuuluu?",
        "K\u00e4\u00e4nn\u00f6s on hauskaa."
    ],
    "target": "en",
    "target-segments": [
        "How are you ?",
        "The translation is fun ."
    ],
    "target-sentences": [
        "How are you?",
        "The translation is fun."
    ]
}
```

Figure 3: Translation output from the WebSocket server. Non-ASCII characters are encoded with their corresponding Unicode character code with the JSON specific encoding scheme. Source and target segments are tokenized in this example and source segments are also segmented into subword units using BPE. The token separator is '@@' indicating that the subsequent space character is to be deleted.

```
make all
sudo make install
cd ..
```

- Adjust the configuration file to meet your plans about the server to be run
- Download and set up a specific language pair (make sure that a model exists for that language pair), here we show the example of Finnish to English:[14]

```
sudo make SRC=fi TRG=en OPUSMT_PORT=10000 MARIAN_PORT-20000 all
```

[14] Admin rights are necessary to install the software daemon and startup scripts in default locations. The models and translation cache database is also set up in globally shared directories that require admin rights to modify in standard Linux systems.

This should start 3 daemons that run the service:

- The Marian NMT server that translates plain text segmented into subword units (running on `MARIAN_PORT`)
- The language-specific OPUS-MT server (running on `OPUSMT_PORT`) that performs pre- and post-processing and interacts with the Marian NMT server
- the OPUS-MT router server that connects various individual OPUS-MT services (running on port 8080 by default)

Configuration files and models will be installed in `/usr/local/share/opusMT/` and startup scripts for the individual services are in `/etc/init.d`.

Starting an additional translation service is easy by re-running the installation script with new parameters, for example a service for translation German to English:

```
sudo make SRC=de TRG=en MARIAN_PORT=10001 OPUSMT_PORT=20001 opusMT-
  server
```

It is still necessary to edit the configuration file to include the new server; add

```
    "localhost:20001" : {
        "source-languages" : "de",
        "target-languages" : "en"
    },
```

and re-install the configuration, and re-start translation services and the router daemon:

```
sudo make opusMT-router
sudo service marian-opus-de-en restart
sudo service opusMT-opus-de-en restart
sudo service opusMT restart
```

Note that the translation servers also build up a cache for efficiency reasons to avoid re-translating identical sentences that have been translated by the same model before. The cache is stored using an SQLite database in `/var/cache/opusMT/`.

Additional recipes for removing services and further details are given in the documentation and the makefile in the OPUS-MT repository[15].

3 Subtitle translation

This section describes tools and resources tailored towards the translation of subtitles. We present pre- and post-processing tools and models that are optimised for subtitle translation.

[15] https://github.com/Helsinki-NLP/Opus-MT/blob/master/Makefile

3.1 Pre- and post-processing

Subtitle translation requires some special treatment to be used with standard machine translation models. The `subalign` software package implements various tools to perform the necessary pre- and post-processing to work with SRT files and their translation. This enables a streamlined pipeline of subtitle translation with regular, sentence-based translation engines. For document-level approaches, please refer to Section 6.

The toolbox includes the following scripts:

`srt2xml`: A tool that converts subtitles in SRT format to simple OPUS-style XML format. It performs sentence splitting and tokenisation using regular expressions and language-specific non-breaking prefixes (taken from the Europarl corpus version tools). The tool also converts between character encodings (using explicit BOM detection or explicit parameters), and implements various heuristics to merge lines in cases of sentences that continue on subsequent lines as well as in subsequent subtitle blocks. It also splits subtitle blocks in case of detected sentence boundaries, and produces sentence boundary markup while keeping time information in place.

`srtalign`: A tool for aligning subtitle files based on time information. The system looks for sentence alignments that maximise the time overlap based on the output produces by `srt2xml`. Time information is extrapolated to match sentence boundaries based on a simple linear correlation between the length of characters and the time span dedicated to that string. Furthermore, the tool implements synchronisation procedures (Tiedemann, 2008) based on lexical anchor points that can be detected using cognate heuristics or bilingual dictionaries. The `subalign` toolbox provides 361 dictionaries extracted from automatic word alignment (about 360,000 dictionary entries altogether).

`mt2srt`: A tool that aligns translated text to a given subtitle template to fill the given time slots with information coming from an MT system. The tool implements a length-based approach for this alignment, using various adjustments for the subtitle alignment case. Basically, it restricts the alignment to 1-to-m alignment types, using the original text in a given time slot as the source segment, and sentence fragments from the translations as the segments in the target language. For this, translated sentences are split into clauses based on punctuation characters, and the procedure finds the globally optimal alignment that minimises the costs based on the length correlation factor. Additional heuristics can be used to constrain the maximum length of a subtitle block and to penalise subtitle breaks within a running sentence. The template can be either in XML or in SRT format, and the input should be plain text with one sentence per line. All data should be encoded in UTF-8. More details on subtitle block alignment are provided in Koponen et al. (2020b).

The MeMAD subtitle translation pipeline that makes use of these tools was released through the `subtitle-translation` repository[16] in the MeMAD Github space. We provide further details on the structure of the pipeline in the repository's documentation, as well as instructions for using the pipeline with pre-trained models, as discussed earlier in Section 2.1. In addition to the automatic reference-based metrics covered in this deliverable, we report our findings from further use case based evaluations of subtitling productivity and end user reception of translated subtitles in deliverable D6.9.

[16] https://github.com/MeMAD-project/subtitle-translation

3.2 Subtitle-optimised translation models

Subtitle translation requires further adjustments that impacts translation models as well. General-purpose models do not necessarily capture the specific properties and language style even if the training data contains large portions of in-domain training data. Furthermore, the alignment of translation to the audiovisual content requires further information to be provided by the translation engine. Therefore, we also produce models that are optimised for the use of this specific task using the following two steps:

Cross-lingual alignment: We train translation models that incorporate unsupervised word alignment into the training procedure in order to guide one of the cross-lingual attention heads to follow the links between tokens in source and target language.

Fine-tuning: We fine-tune models for the subtitle translation domain using a second step of additional training after creating a general-purpose translation model.

For the guided alignment feature, we use the efficient statistical word aligner `eflomal`,[17] which has been developed in the Language Technology research group in the University of Helsinki. We apply the software on the parallel training data segmented on the subword level using SentencePiece, in order to produce links that can directly be used by the neural MT model as a correspondence to cross-lingual token-level attention. `eflomal` is run on equally-sized partitions of the data, each with 5 million sentence pairs, to enable efficient and reliable alignments even on the big datasets that we are working with. We run word alignment for each language pair in both directions, and symmetrise the two directions using *grow-diag-final* heuristics (Koehn, 2009), a common strategy that emphasises recall and data coverage.

Besides bilingual translation models, we also train multilingual models in order to provide compact models that cover all the focus languages of the project. Multilingual models offer the possibility to deploy a single translation service that can be used for multiple translation directions and, hence, decrease resource requirements when running extensive services. Related work also reports positive effects through transfer learning when training multilingual models. However, in our case of high-resource languages we cannot see that effect and rather see slight drops in performance when using multilingual settings instead of strong bilingual models. Nevertheless, the remaining advantage of larger language coverage in a single model can still outweigh the minor decrease in translation quality depending on the task and its performance requirements.

The training set for the multilingual models is balanced between the individual language pairs, using a simple sampling strategy on shuffled data. In our case, we use one million sentence pairs per language pair to create a sufficiently large but still manageable dataset to train the system. Target language tokens are added to the system to enable the translation into various languages in the otherwise completely shared model among all translation directions. Language labels are given as ISO 639-3 codes enclosed between double inequality signs to distinguish them from regular words. For example, Swedish for the target language is encoded by adding a token >>*swe*<< to the beginning of the input string.

In summary, we include the following models:

[17] https://github.com/robertostling/eflomal

- Bilingual models: All combinations and translation directions for the six MeMAD focus languages, using all data available from the Tatoeba-MT challenge release.[18]
- A many-to-English translation model from all focus languages to English (without language labels), and sampled training data from the Tatoeba-MT challenge. Translations into English is a common use case and, therefore, we include this specific setup in our list of supported models.
- An English-to-many translation model that translates from English to any of the focus languages using unique language labels (same sampled data as above). Similar to above, translations from English are common in real-world use cases and, hence, this particular multilingual setup is important.
- A many-to-many translation model that covers all focus languages in all directions trained on data sampled from the same source as the other models. In contrast to the models above, this instance covers all translation directions between the focus languages of the MeMAD project.

For fine-tuning, we use the domain labels given by the Tatoeba-MT challenge release, and extract a sample of the subtitle section of each of the datasets (one million sentence pairs). Furthermore, we fine-tune on MeMAD internal data coming from Yle to contrast the models with the performance of clearly in-domain tuning. For the latter we reserved 10,000 sentences each for validation and kept the rest for training the fine-tuned models. The final models are then tested with an independent benchmark test set also provided by Yle with no overlap with the training and development data. This data set contains 1,254 sentence pairs for Finnish-Swedish general-purpose subtitles (FIN-SWE), 2,837 sentences that translate Finnish subtitles to Swedish subtitles for the hearing impaired (FIN-SWH) and 625 sentences that translate Finnish subtitles for the hearing impaired to general purpose subtitles in Swedish (FIH-SWE).

tune / test	FIN-SWE	FIH-SWE	FIN-SWH	SWE-FIN	SWH-FIN	SWE-FIH
baseline	22.3	17.0	18.2	20.8	15.9	12.2
OpenSubtitles (1M)	22.0	16.8	17.9	20.9	15.7	12.5
Yle-all (2M)	24.7	19.6	19.5	22.7	17.4	13.6
Yle-FIN-SWE (1.1M)	24.9	18.9	19.5	23.1	17.3	13.9
Yle-FIH-SWE (47k)	23.6	19.7	18.4	21.5	16.0	14.8
Yle-FIN-SWH (850k)	23.8	18.5	19.5	23.0	17.7	13.9

Table 3: Fine-tuning general-purpose NMT models (*baseline*) for the subtitle domain. Test sets include translations between subtitles for Finnish (FIN) and Swedish (SWE) with variants of subtitles for the hearing impaired (FIH and SWH, respectively). The rows refer to different tuning sets including one million parallel sentences sampled from OpenSubtitles (OPUS) and MeMAD internal training data provided by Yle with subsets for the specific language variants (general subtitles or for the hearing impaired).

Table 3 shows the results when evaluating on the internal YLE benchmarks. They demonstrate the importance of fine-tuning and the appropriateness of the data used for that purpose. The metrics show that generic subtitle data (from OpenSubtitles) is not good enough for domain adaptation as this material represents a wide variety of genres mostly (probably even exclusively) with a source language other than Finnish or Swedish. The performance actually deteriorates slightly when continuously optimising the pre-trained model for that data set. On

[18] https://github.com/MeMAD-project/Tatoeba-Challenge

the other hand, in-domain training data leads to clear improvements with a significant impact on the style match that refers to subtitles for the hearing impaired or the general audience. This is even more remarkable considering the small data set that we have available for the translation from and to Finnish for the hearing impaired showing the importance of examples that teach the model to take care of the inherent style difference.

We release all our models using our project community space on Zenodo[19] with links from the `subtitle-translation` repository[20] on Github. Unfortunately, the in-domain training and test data cannot be released together with the models due to copyright issues and license agreements. We are currently negotiating the release of the test set in order to provide a benchmark for comparing and replicating our results.

4 Image caption translation

The best-performing image caption translation model implemented within MeMAD was released through our Zenodo community[21], and the documentation is available from the `image-caption-translation` repository[22] on Github. The system constitutes our submission to the shared task at WMT 2018, and implements the winning system in that competition (Barrault et al., 2018). The system is carefully described in Grönroos et al. (2018) and we will not repeat the details here. This report focuses on the release details and provides links and installation instructions in order to deploy and run the model. Below we list the essential steps for setting up the system and refer the reader to the original publication to understand the architecture of the model.

Downloading the model

Fetch the model from Zenodo using the following command

```
curl https://zenodo.org/record/4038444/files/opennmt.transformer.
   multiling.mscoco%2Bmulti30k%2Bsubs3M.domainprefix.mmod.imgw.meanfeat
   .detectron.mask_surface.bpe50k_acc_80.57_ppl_2.43_e23.pt?download=1
   --output models/opennmt.transformer.multiling.mscoco+multi30k+subs3M
   .domainprefix.mmod.imgw.meanfeat.detectron.mask_surface.
   bpe50k_acc_80.57_ppl_2.43_e23.pt
```

Installing the software

Start by cloning the github repository at https://github.com/MeMAD-project/image-caption-translation.git.

We recommend a conda-based installation and provide the corresponding commands for installing the codebase below. The software used CUDA by default but can also run on CPU.

[19] https://zenodo.org/record/4556121
[20] https://github.com/MeMAD-project/subtitle-translation
[21] https://zenodo.org/record/4038444
[22] https://github.com/MeMAD-project/image-caption-translation

Feature extraction software can be installed in the following way:

```
conda create --name memaddetectron2 --file env/detectron2.cuda.conda -c pytorch
source activate memaddetectron2
pip install -r env/detectron2.cuda.pip
source deactivate
```

The installation of the translation system is described below. Note that this codebase uses specific versions of some libraries, which is specified in the environment files included in the repository used by the installation commands below. CUDA is disabled.

```
conda create --name memadmmt --file env/mmt.nocuda.conda -c pytorch
source activate memadmmt
pip install -r env/mmt.nocuda.pip
git clone https://github.com/Waino/OpenNMT-py.git
pushd OpenNMT-py
git checkout develop_mmod
python setup.py install
popd
source deactivate
```

Using the model

First of all, we need to extract `detectron2` features and store them in `img_feat.npy`:

```
source activate memaddetectron2
tools/image-features.py --imglist data/imglist
source deactivate
```

Note the following settings: `img_feat_dim=80`, `dtype=torch.float32`, saved as an $(N, 80)$ matrix in NumPy `.npy` format, where N is the number of lines to translate.

The next step is to apply BPE segmentation to previously tokenised and lowercased text:

```
source activate memadmmt
tools/apply_bpe.py --codes models/bpe.50k.multiling < data/input > data/segmented
```

After that, prepend the target language tag (either TO_de or TO_fr) and the domain tag:

```
sed -e "s/^/<TO_de> <DOMAIN_caption> /" < data/segmented > data/prefixed.de
sed -e "s/^/<TO_fr> <DOMAIN_caption> /" < data/segmented > data/prefixed.fr
```

Finally, we are able to perform translations:

```
OpenNMT-py/translate_mmod_finetune.py \
    -model models/opennmt.transformer.multiling.mscoco+multi30k+subs3M.domainprefix.mmod.imgw.meanfeat.detectron.mask_surface.bpe50k_acc_80.57_ppl_2.43_e23.pt \
    -src data/prefixed.de \
    -path_to_test_img_feats img_feat.npy \
    -output data/translated.de \
    --multimodal_model_type imgw
```

The translations still need to be post-processed in order to join BPE subwords and to recase the output.

As described in the paper (Grönroos et al., 2018), it is also possible to feed zero vectors as dummy features by replacing -path_to_test_img_feats img_feat.npy with -path_to_test_img_feats dummy.zeros.npy in the above command.

5 Spoken language translation

Whenever we needed to translate spoken language in MeMAD, we opted for a cascaded approach. This approach realises translation from audio containing spoken language by pipelining (1) automatic speech recognition (ASR), and (2) text-based machine translation (MT) stages. Essentially, this pipeline breaks the task of multimodal translation down into modality conversion followed by unimodal translation.

Our experiments on subtitling productivity and the user reception of automatically-generated subtitles, as part of the Use Case UC4 evaluations, both featured such pipelines. However, these have been cross-work-package efforts, integrated together under WP6. While the MT components were developed in the University of Helsinki and released within WP4, the ASR components were provided by Lingsoft and Aalto University as part of WP2. The scope of this deliverable only includes the MT components, however, the full speech translation pipeline can be reproduced by first generating a transcript of the audio using Lingsoft ASR, and continuing with the MT pipeline from the sentence segmentation step. We report further details of the pipeline as utilised for subtitle translation in deliverables D6.6 and D6.9.

We have also experimented with end-to-end speech translation, where a monolithic system undertakes the translation of source language audio to target language text in a single stage. Our release includes two sets of translation models that are able to translate between English and German in either direction—one which only processes source language audio, and another which can also make use of transcripts of the audio as auxiliary text input. Although the system performs multimodal translation end-to-end, the translation process still involves offline preprocessing and postprocessing steps. The preprocessing step is required when translating with transcripts, and performs normalisation, truecasing, and subword segmentation on the text input. The postprocessing step converts the translation output to a human-readable plain text format, and is required for both audio-to-text and audio+text-to-text translations.

Our end-to-end speech translation models were developed, along with the subtitle translation pipelines, in preparation for the MeMAD submissions to the IWSLT offline speech translation task in 2020 (Vázquez et al., 2020). Unlike the subtitle translation pipelines, the end-to-end speech translation models remained unchanged since our 2020 submission. The final version of the system has been released via the `speech-translation` repository[23] in the MeMAD Github space, including download links for the pre-trained models archived on Zenodo, and detailed instructions for setting up the system.

6 Discourse-aware translation

This deliverable includes two released packages in relation to discourse-aware machine translation: (1) Pre-trained concatenation models for three language pairs and (2) a dataset for benchmarking document-level approaches to machine translation.

6.1 Document-level models

We release six pre-trained models for concatenation-based document-level translation. In particular, we provide models that translate between Finnish and three other languages, English, French and Swedish in both directions for each language pair. The models can be downloaded from https://doi.org/10.5281/zenodo.4287562.

The models were trained using Marian (v1.8.2), implementing state-of-the-art transformer architectures, using 6 layers in both the encoder and the decoder, with 8 self-attention heads per layer. We use SentencePiece for tokenisation and subword segmentation and train each model on large collections of human translations provided by OPUS, the open parallel corpus.

The document-level models apply a so-called concatenation approach in which larger chunks are concatenated in order to enable cross-sentence dependencies to be covered by the model. In our case, we use text windows of up to 100 tokens on both sides and apply a simple greedy segmentation approach to create those chunks. Table 4 summarises the size of each data collection we use for training. We can see, that we have substantial amounts of data in each language pair ranging from 30 million to 44 million sentence pairs that constitute between 5.7 and 8.5 million document-level segment pairs to train on. This means that the window of 100 tokens creates segments with 5–6 sentences on average.

We also provide the scripts to pre- and post-process data that need to be translated with those models, and the procedures are straightforward (assuming a specific benchmark dataset and model in this case):

[23] https://github.com/MeMAD-project/speech-translation

language pair	segment pairs	sentence pairs
fi → sv	5,712,031	30,604,442
fi → en	8,494,683	43,942,504
fi → fr	6,087,869	30,138,134

Table 4: Sizes of the training data used for document-level MT models.

```
1 spm_encode --model models/fi-en/opus.src.spm32k-model < data/newstest2019-fien-src.fi.txt | scripts/split-text.pl -l 100 > data/data/newstest2019-fien-src.fi.doc100
2 marian-decoder -c models/fi-en/decoder.yml < data/newstest2019-fien-src.fi.doc100 > data/newstest2019-fien-sys.en.doc100
3 scripts/post-process.sh < data/newstest2019-fien-sys.en.doc100 > data/newstest2019-fien-sys.en.txt
```

The commands above perform the essential pre-processing steps (line 1) including subword segmentation using a pre-trained SentencePiece model provided in the release followed by a script that splits the text into chunks of maximum 100 tokens. The size needs to match the model that is applied in the subsequent step, which calls the NMT decoder (line 2) with the input created in the first step. The final step performs simple post-processing steps mainly referring to merging subword units produced by the NMT decoder. The prerequisites for running the translation with the steps above are a successfully compiled Marian decoder and the SentencePiece subword segmentation software installed in the path of your your system.

A detailed analysis of document-level models and their impact on the work of professional translators is provided in Koponen et al. (2020a).

6.2 Benchmarks

We release the datasets used in our study on concatenation approaches to document-level machine translation published at DiscoMT 2019 (Scherrer et al., 2019). They are taken from the English–German news translation task at WMT 2019 and the English–German bitext in the OpenSubtitles 2016 collection from OPUS. All datasets are sentence-aligned with corresponding lines being aligned to each other. Document boundaries are marked with empty lines (on both sides of the parallel corpus). The release contains a dedicated split into development, test and training data in the two domains considered in the study. The package can be downloaded from https://zenodo.org/record/3525366.

The following list shows the essential content of the release along with individual file sizes:

```
    dev
        ost.tok.de              6.2 MB
        ost.tok.en              5.8 MB
        wmt.tok.de            663.1 kB
        wmt.tok.en            592.5 kB
    test
        ost.tok.de            188.2 kB
        ost.tok.en            203.3 kB
        ost.nocontext.tok.de  193.2 kB
        ost.nocontext.tok.en  208.3 kB
        ost.shuffled.tok.de   188.3 kB
        ost.shuffled.tok.en   203.3 kB
        wmt.tok.de            397.1 kB
        wmt.tok.en            360.6 kB
        wmt.nocontext.tok.de  400.0 kB
        wmt.nocontext.tok.en  363.5 kB
        wmt.shuffled.tok.de   397.1 kB
        wmt.shuffled.tok.en   360.6 kB
    train
        ost.tok.de            516.4 MB
        ost.tok.en            479.7 MB
        wmt.de-en.tok.de      504.1 MB
        wmt.de-en.tok.en      435.2 MB
        wmt.en-de.tok.de        1.7 GB
        wmt.en-de.tok.en        1.5 GB
    unshuffle.py
```

The interesting part of the benchmark is the inclusion of shuffled and non-contextualised variants of the test data that makes it possible to evaluate systems with respect to their use of discourse-level dependencies. The assumption is that shuffled context or no context at all should show up in the evaluation scores hurting systems that rely on discourse-level features as part of their decision process.

7 Conclusion

This deliverable has presented all of our releases of tools and models developed in connection with the main tasks of MeMAD WP4. Our general-purpose machine translation models and spoken language translation systems have been widely-used catalysts for project-wide collaboration, and likely the most valuable technology among the contributions of WP4 in the larger scheme of MeMAD. As a consequence, our corresponding model and software releases make up a large portion of our public releases. Our efforts in developing translation systems that involve multimodality (image caption translation, spoken language translation) and discourse-awareness (document-level machine translation) have demonstrated that they serve relatively small niches. Our initial work on these systems has been in accordance each with a corresponding public shared task, so that we would become familiar with the state of the art, and avail ourselves of training and test datasets, as well as targeted evaluation methods. While our image caption translation and document-level machine translation releases correspond to the outcomes from our tasks for 2018 (as described in D4.1) and 2019 (as described in D4.2) respectively, we have expanded on both in the final year of MeMAD. In regard to multimodality, we have finalised our pipeline formula for semi- or fully-automatic interlingual subtitling of media containing spoken language. In the context of discourse-aware machine translation, we have introduced a benchmarking set tailored for document-level machine translation in order to facilitate further research, as the culmination of our work on the subject.

All WP4 releases have been made open for public use under permissive licences. In general, we have made our model releases as open access deposits via the MeMAD Zenodo community. Each deposit includes a summary description, and links to further resources and documentation as required. Our software releases have been made through code repositories registered under the MeMAD Github organisation. Each such repository contains the full source code as well as documentation that lists software dependencies, installation/usage instructions and examples, and/or external links to pre-trained models and datasets as necessary. We trust that our releases of tools and models will make life easier for others wishing to reproduce our results, use our outputs, and hopefully build upon them to advance the state of the art.

References

Loïc Barrault, Fethi Bougares, Lucia Specia, Chiraag Lala, Desmond Elliott, and Stella Frank. 2018. Findings of the third shared task on multimodal machine translation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 304–323, Belgium, Brussels. Association for Computational Linguistics.

Stig-Arne Grönroos, Benoit Huet, Mikko Kurimo, Jorma Laaksonen, Bernard Merialdo, Phu Pham, Mats Sjöberg, Umut Sulubacak, Jörg Tiedemann, Raphael Troncy, and Raúl Vázquez. 2018. The MeMAD submission to the WMT18 multimodal translation task. In *Proceedings of the Third Conference on Machine Translation*. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Philipp Koehn. 2009. *Statistical Machine Translation*. Cambridge University Press.

Maarit Koponen, Umut Sulubacak, Kaisa Vitikainen, and Jörg Tiedemann. 2020a. Mt for subtitling: Investigating professional translators' user experience and feedback. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (AMTA), 1st Workshop on Post-Editing in Modern-Day Translation*, pages 79–92.

Maarit Koponen, Umut Sulubacak, Kaisa Vitikainen, and Jörg Tiedemann. 2020b. MT for subtitling: User evaluation of post-editing productivity. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 115–124, Lisboa, Portugal. European Association for Machine Translation.

Yves Scherrer, Jörg Tiedemann, and Sharid Loáiciga. 2019. Analysing concatenation approaches to document-level NMT in two different domains. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, Hong-Kong. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal. European Association for Machine Translation.

Jörg Tiedemann. 2008. Synchronizing translated movie subtitles. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA). Http://www.lrec-conf.org/proceedings/lrec2008/.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Raúl Vázquez, Mikko Aulamo, Umut Sulubacak, and Jörg Tiedemann. 2020. The University of Helsinki submission to the IWSLT2020 offline SpeechTranslation task. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 95–102, Online. Association for Computational Linguistics.

A Appendices

Below we attach a number of recent papers that demonstrate the use of our models and tools in professional workflows (papers A.1 and A.2) and our efforts related to the development of speech-to-text translation systems and resources for open machine translation development (papers A.3, A.4 and A.5).

Appendix A.1: A paper that discusses the experience and feedback of professional translators when working with MT integrated in the workflow of subtitle translation. Published at AMTA 2020

Appendix A.2: An evaluation of post-editing productivity in subtitling with machine translation. Published EAMT 2020.

Appendix A.3: A short paper on the development of open NMT models and tools that cover a large range of languages. Published at EAMT 2020.

Appendix A.4: A system paper describing the MeMAD submission to the offline speech translation task at IWSLT 2020.

Appendix A.5: A research paper presenting a new large scale resource and benchmark for machine translation with a large coverage of languages and language combinations. Published at WMT 2020.

A.1 AMTA paper on MT for professional subtitling

MT for Subtitling: Investigating professional translators' user experience and feedback

Maarit Koponen maarit.koponen@helsinki.fi
Department of Digital Humanities, HELDIG, University of Helsinki, Helsinki, Finland

Umut Sulubacak umut.sulubacak@helsinki.fi
Department of Digital Humanities, HELDIG, University of Helsinki, Helsinki, Finland

Kaisa Vitikainen kaisa.vitikainen@yle.fi
Yleisradio Oy, Helsinki, Finland

Jörg Tiedemann jorg.tiedemann@helsinki.fi
Department of Digital Humanities, HELDIG, University of Helsinki, Helsinki, Finland

Abstract

This paper presents a study of machine translation and post-editing in the field of audiovisual translation. We analyse user experience data collected from post-editing tasks completed by twelve translators in four language pairs. We also present feedback provided by the translators in semi-structured interviews. The results of the user experience survey and thematic analysis of interviews shows that the translators' impression of post-editing subtitles was on average neutral to somewhat negative, with segmentation and timing of subtitles identified as a key factor. Finally, we discuss the implications of the issues arising from the user experience survey and interviews for the future development of automatic subtitle translation.

1 Introduction

Developments in translation technology and machine translation (MT), particularly the quality improvements achieved by neural machine translation (NMT) in recent years, have led to MT increasingly becoming part of the modern-day translators' toolkit. Although post-editing (PE), where MT is used to produce a raw translation output which is then checked and corrected by a translator, has increased in many areas of translation, its use remains uncommon in audiovisual translation (AVT). AVT approaches include dubbing, voice-overs and subtitling for the purpose of making AV content accessible to audiences with no or limited understanding of the language of the original content. Different approaches are used to varying degrees depending on the type of content (e.g. voice-overs are common for documentaries) and region (e.g. subtitling is the predominant practice in Northern European countries).

As studies and practical experience have shown potential for PE in increasing productivity in other forms of translation, interest in implementing MT tools and PE workflows has also grown in the AV field. Studies have explored the use of MTPE subtitle translation with some promising although mixed results regarding effect on productivity (e.g. Bywood et al., 2017). When exploring the usability of such tools, however, productivity measurement is only one aspect. As Etchegoyhen et al. (2014) argue, subjective feedback from translators is equally important, as it provides insight into the actual user experience and necessary improvements.

This paper presents a pilot study investigating the usability of MT and PE in the subtitling workflow from the perspective of the prospective users. Twelve professional subtitle translators working in four language pairs (Finnish↔Swedish and Finnish↔English) subtitled short video clips by post-editing MT output. We analyse feedback collected with a user experience questionnaire and semi-structured interviews for positive and negative evaluations of the PE experience and improvement suggestions. We start with an overview of related work on MT and PE in the subtitling context and work on user feedback (Section 2). After describing our approach to automatic subtitle translation (Section 3), and the subtitle PE experiment (Section 4), we present the questionnaire and interview analyses (Section 5), followed by discussion of the observations and our ongoing work based on these analyses.

2 Related work

2.1 Subtitling, MT and PE

Subtitle translation differs from translating purely textual material in that the source text consists of the spoken audio, together with the visual mode, while the target text is a written representation of translated speech. Due to technical limitations like the number of characters within a subtitle frame and the time each subtitle remains visible, paraphrasing and condensation are typical features of subtitle translation (see e.g. Pedersen, 2017). The work of subtitle translators may involve "first translation", where they translate from the source audio and determine the segmentation and timing of the subtitle frames ("spotting"), or translation with subtitle templates, where the source text consists of pre-existing intralingual subtitles in the source language or sometimes interlingual subtitles in a pivot language (often English) with set subtitle segmentation and timing (Nikolić, 2015).

To date, the use of MT and PE for subtitling has been less common in AVT than other translation fields. Explanations for this may include the characteristics of subtitle translation, which pose challenges for MT, and also the difficulty of integrating current NMT systems to subtitle translation workflows (Matusov et al., 2019). MT for movie and TV subtitling has been tested in some language pairs since the early 2000s (Melero et al., 2006; Volk et al., 2010; de Sousa et al., 2011) with suggestions that PE may increase productivity also in this context.

A subtitle-oriented statistical MT system and PE platform was developed by the SUMAT project, and tested in a user evaluation involving several language pairs and 19 professional subtitle translators (Etchegoyhen et al., 2014; Bywood et al., 2017). In a study comparing task time for translation from scratch and MTPE, Bywood et al. (2017) report that MTPE increased the translators productivity; however, the results varied for different translators, language pairs and content types. More recently, Matusov et al. (2019) tested an NMT system customised for subtitles using parallel subtitle corpora from OpenSubtitles, GlobalVoices and TED talks and reported productivity increases for MTPE in a study involving two translators.

So far, work has focused on the use of intralingual subtitles as the source text for MT, but a recent paper by Karakanta et al. (2020a) explores an end-to-end spoken language translation system for subtitling. No user evaluation of the system is reported, although Karakanta et al. (2020a) note that based on automatic evaluation against "gold standard" human subtitles the MT quality appears satisfactory. Karakanta et al. (2020b) also investigate annotating subtitle corpora for segment breaks and propose an approach for segmenting sentences into subtitles conforming to length constraints.

2.2 Studies on user experience/feedback from translators

Subjective feedback is invaluable for providing insight into tools and workflows that affect the actual work of the prospective users, and revealing issues that would not be evident from the translations or process data (see Bundgaard, 2017). Various studies have investigated professional translators' experience with and perceptions of MT and PE with questionnaires and interviews. Analyses have reported mixed experiences: while translators sometimes find MT helpful, for example by providing useful terminology and making their work faster, other times PE may be even slower than translation from scratch. Whether working with technical or literary texts, translators often express concerns about MT affecting the final translation quality as well as their (cognitive) processes because the output can potentially mislead the translator or limit their creativity (e.g. Guerberof Arenas, 2013; Bundgaard, 2017; Moorkens et al., 2018).

Translator feedback on MT and PE in the context of AV translation was collected and analysed in the user evaluations of the SUMAT project (Etchegoyhen et al., 2014; Bywood et al., 2017). Etchegoyhen et al. (2014) describe a questionnaire used in the second evaluation round, where 19 translators carried out PE tasks in several language pairs and rated their impression of the PE process rather negatively overall (average 2.37 on a 5-point scale). Based on translator feedback, Etchegoyhen et al. (2014) identified improving MT quality to reduce cognitive load, improving quality estimation and filtering MT segments, and improving user interfaces for PE of MT subtitles as key issues for increasing usability.

Matusov et al. (2019) report a user experiment with two translators who both subtitled two programmes (a documentary and a sitcom) partly from scratch and partly with two different MT outputs. The translators rated their impression of the PE experience on average "fair" (3 on a 5-point scale) for the subtitle optimised system. The translator feedback noted useful terminology as one of the main reasons they would consider using MT in their work, but also expressed concerns about incorrect or unusual translations in the MT affecting the quality of the final translation (Matusov et al., 2019).

The study reported in this paper builds upon these analyses by collecting feedback on MT and PE for subtitling from professional subtitle translators. We aim to investigate the translators' impressions of PE more closely by introducing a more detailed user experience questionnaire where they rate different aspects of the process (see Section 4.3).

3 Automatic subtitle translation

Machine translation for subtitles requires some special treatment that we will discuss in this section. In particular, we consider models with extended context, which we will call *document-level translation models* and special tools that align translations with subtitle frames to be shown on screen. First, we briefly present the datasets and models before discussing frame alignment as a post-processing step.

3.1 Datasets and MT models

Our MT models are trained on a mix of diverse data sets[1] taken from OPUS.[2] Altogether, this includes over 30 million translation units for Finnish↔Swedish and about 44 million units for Finnish↔English. We follow the common practice in MT development to include as much data as possible even when coming from very different domains. However, the largest proportion of the training examples comes from a large collection of movie and TV show subtitles (the OpenSubtitles v2018 dataset) constituting almost half of the Finnish↔Swedish data and over 65% of the Finnish↔English data. This is certainly an advantage for our task and, hence, we expect a rather good domain-fit of our models.

We train both sentence-level and document-level models based on the Transformer architecture (Vaswani et al., 2017), the current state of the art in NMT. In particular, we apply the implementation from the MarianNMT toolkit (Junczys-Dowmunt et al., 2018), a production-ready software with fast training and decoding tools. The architecture refers to a 6-layered

[1] OPUS corpora used: bible-uedin, DGT, EMEA, EUbookshop, EUconst, Europarl, Finlex, fiskmo, GNOME, infopankki, JRC-Acquis, KDE4, MultiParaCrawl, OpenSubtitles, PHP, QED, Tatoeba, TildeMODEL, Ubuntu, wikimedia

[2] http://opus.nlpl.eu

network in both the encoder and decoder with 8 self-attention heads per layer. Recommended features like label smoothing and dropout are enabled and we use tied embeddings and a shared vocabulary. SentencePiece (Kudo and Richardson, 2018) is used for tokenisation and subword segmentation with models independently trained for source and target language. The shared vocabulary is set to a size of 65,000 with equal proportions in each language.

The document-level models refer to *concatenative models* proposed by Tiedemann and Scherrer (2017) and Junczys-Dowmunt (2019) using units of a maximum length of 100 tokens and special tokens for marking sentence boundaries. We observed that 100 tokens typically covers a substantial amount of contextual information in subtitles where sentences and sentence fragments are often very short. About 3.3 million pseudo-documents are created in a sequential way without overlaps for Finnish↔Swedish and 4.7 million pseudo-documents for Finnish↔English, corresponding to roughly 9 sentences per document on average.

The same kind of chunking needs to be done during test time. Sentence-level models are translated in the usual way. Note, however, that subtitles need to be pre-processed in a proper way in order to extract proper textual units that correspond to complete sentences to be translated. This involves merging fragments that run across subtitle frames and splitting frames in other cases.

We apply all our models to a dedicated test set taken from a larger set of subtitles from public broadcasts with audio in Finnish, Swedish or English. For this, intralingual subtitles (subtitles in the language of the original audio) are aligned with interlingual subtitles of the same programme in another language. The test set was carefully checked and non-corresponding segments are removed. Note that interlingual subtitles are produced independently from intralingual ones and, therefore, do not refer to direct translations of one another. Subtitles for the hard-of-hearing are also included but in a separate subset.

benchmark	**sentence-level**		**document-level**	
	BLEU	**chrF$_2$**	**BLEU**	**chrF$_2$**
fi→sv	18.8	0.443	19.3	0.451
sv→fi	15.7	0.449	16.8	0.462
fi→en	21.5	0.458	23.6	0.472
en→fi	16.0	0.444	17.1	0.454

Table 1: Comparison of BLEU and chrF$_2$ scores on the benchmark test set for the sentence-level and document-level systems in the language pairs Finnish→Swedish, Swedish→Finnish, Finnish→English, and English→Finnish.

Translation results for different subsets (scores calculated for spans of all subtitles within a video) are listed in Table 1. Evaluation of document-level translation required one additional step of aligning the automatically generated translations with corresponding reference translations. For this, we apply standard sentence alignment algorithms implemented in hunalign (Varga et al., 2005) using the re-alignment flag to enable lexical matching that ought to be very beneficial in this monolingual alignment task. Note that the automatic alignment may have negative effects on the final BLEU scores further supporting the strong result achieved by the document-level models compared to sentence-level ones according to the automatic evaluation. The scores indicate a consistent gain in using document-level information in both language pairs and all translation directions. Later, in Section 5, we will see, however, that the encouraging result does not hold in the manual assessment, which is most probably due to problems in segmentation and time frame alignment that we will discuss in the section below.

3.2 Subtitle frame alignment

One of the crucial steps in subtitle translation is the assignment to appropriate time slots. Our approach is to map translations back into the frames defined in the original source language subtitles assuming that they can fit in a similar way as the source language text was segmented. Those subtitle frames may include multiple sentences and sentences may stretch over several frames. Sentence extraction from the original subtitles is done with the techniques proposed by Tiedemann (2008). Time allocation of the translated sentences is implemented as yet another alignment algorithm.

Subtitles converted to sentence-level segments in XML:

```
  <s id="13">
    <time id="T16S" value="00:01:05,960" />
We have to make readmission agreements with
other countries, -
    <time id="T16E" value="00:01:12,360" />
    <time id="T17S" value="00:01:12,440" />
so that they would be willing.
  </s>
  <s id="14">
We have to cooperate closely.
    <time id="T17E" value="00:01:17,440" />
  </s>
```

Mapped back to subtitle frames after translation:

```
16
00:01:05,960 --> 00:01:12,360
Meidän on tehtävä
takaisinottosopimuksia muiden maiden kanssa,

17
00:01:12,440 --> 00:01:17,440
jotta ne olisivat halukkaita.
Meidän on tehtävä tiivistä yhteistyötä.
```

Figure 1: Pre- and post-processing of subtitle data before and after translation. Sentences may run over several subtitle frames and multiple sentences and sentence fragments can also appear in the same time frame. The translation comes from a document-level model.

Once again, we apply a length-based sentence alignment model to map translations to the given time slots in the source language frames. In contrast to standard bitext alignment we are now interested in 1-to-n alignments only in which each existing subtitle frame needs to be filled with one or more segments coming from the automatically generated translations. For the target language segmentation we consider simple heuristics for splitting sentences into clauses by breaking strings that are separated by punctuation plus space characters. Resulting sequences that exceed a certain length threshold are further split on space characters closest to the center of the string. After that, we apply the famous Gale & Church algorithm (Gale and Church, 1993) to optimise the global alignment between source segments (original subtitle frame data) and target segments with adjusted parameters referring to our specific task: (1) We apply a uniform prior over alignment types as there is no strong preference for frame-to-clause alignment in our case. (2) We define alignment types to include one-to-x units only with x ranging from one to four. (3) We introduce extra costs to discourage frame boundaries within running sentences and assignments that violate length constraints. Figure 1 shows an example outcome of the procedure.

Finally, we also apply simple heuristics to insert line breaks making them conform to length and formatting constraints. During the manual assessment, we found out that this segmentation and the introduction of length violation costs caused severe damage to the time slot assignment pointing out the importance of proper optimisations of those steps. The implementation of our frame alignment algorithm is available as an open source package.[3]

4 Subtitle post-editing experiment and collecting user feedback

The study described in this paper is a part of a research project involving MT and other technologies as a tool for managing and processing AV material. The purpose of this study was

[3] https://github.com/Helsinki-NLP/subalign

to investigate the usability of MT for interlingual subtitling in an experiment carried out in November–December 2019 with professional subtitle translators using MT and PE to subtitle short video clips. The experiment involved recording process data with keylogging software (Inputlog, see Leijten and Van Waes, 2013). The translators' subjective evaluations of the usability of MTPE for subtitling were collected with a questionnaire focused on the user experience and semi-structured interviews. In this paper, we focus on these subjective evaluations and the professional translators' user experience of MT and PE for subtitling.

4.1 Participants and the subtitling workflow context

The experiments were carried out at the Finnish public broadcasting company Yle which produces and broadcasts AV content on television and an online streaming service. The company employs in-house translators and outsources some translation work. Subtitling is the most common approach to AVT in this company and Finland in general. Subtitle templates are generally not used by Yle, rather, the translators' normal workflow involves first translation from audio and spotting the subtitles manually. The translators follow quality recommendations which specify, for example, technical recommendations like number of characters in subtitle frames, minimum and maximum duration of subtitle frames on screen and maximum reading speed, as well as linguistic features. National guidelines for subtitle translation published in 2020[4] reflect the practices already in place at the broadcasting company.

The subtitling tasks were carried out in four language pairs: Finnish→Swedish, Swedish→Finnish, Finnish→English and English→Finnish. Twelve translators (three per language pair) participated in the experiment: eight in-house translators and four freelancers with between 4 and 30 years of professional experience as subtitle translators in the relevant language pair. Two participants stated they had experimented with using MT for subtitling prior to this test, and seven others had used MT for other purposes.

4.2 Materials and subtitling tasks

Video clips to be subtitled were selected from datasets representing two content types: EU election debates (unscripted dialogue between multiple participants) and lifestyle or cultural programmes (semi-scripted dialogue or monologue by programme hosts on various topics e.g. movies, food and drink). Each clip was selected so that it (1) formed a coherent, self-contained section of the program as a whole; (2) was approximately 3 minutes long; and (3) contained approximately 30–35 intralingual subtitles. The length and number of clips was limited due to the limited availability of participants for the experiments. Some clips consisted of complete programmes of suitable length, while others were cut from longer programmes ensuring that they formed a coherent, self-contained section. Human-generated intralingual subtitles in the source language were translated with two different MT models, and aligned to SRT files using the subtitle segmentation and timing from the intralingual subtitles as detailed in Section 3.1.

The subtitling tasks were carried out using the subtitlers' preferred subtitling software (Wincaps Q4 or Spot). An external monitor and keyboard were provided, and the subtitlers had access to the internet as well as terminology and other resources normally used in their work. The participants were instructed to create subtitles that would be acceptable for broadcasting, and to follow their normal working processes, but to not spend excessive time on "polishing" any given wording or on researching information. No explicit time limit was given, rather, the participants were instructed to work at their own pace.

Each participant carried out six tasks: MTPE for four clips (two clips with sentence-level MT output and two clips with document-level MT output), and translation from scratch for two clips, with spoken audio as source and manual spotting. To mitigate potential differences related

[4] Currently available in Finnish and Swedish: `http://www.av-kaantajat.fi/Laatusuositukset/`.

to difficulty of each clip and facilitation effect, the clips and MT outputs were rotated so that each clip was subtitled with no MT output, with sentence-level MT output and with document-level MT output by a different participant, and task order was varied. An experimenter was present to set up each task, assist with any potential technical issues and conduct the post-task interview, but did not interact with the participants during the tasks.

4.3 User Experience Questionnaire

An online form was used to collect subjective evaluations of the usability of the MT output for PE. The questionnaire was based on the User Experience Questionnaire (UEQ) developed by Laugwitz et al. (2008) for end-user evaluation of software products. The objective of the UEQ is to provide users with a "simple and immediate way to express feelings, impressions and attitudes" toward the product and thereby elicit quick but comprehensive assessments of user experience (Laugwitz et al., 2008, 64). It consists of scalar ratings of opposing adjective pairs (e.g. *practical/impractical*) intended to measure both classic usability aspects and user experience aspects. The adjective pairs are shown on an scale of 1–7, with positive and negative adjectives alternating on the left/right.

Because the focus of our study was on the participants' experience of MTPE rather than subtitling software, a modified version of the UEQ was created to focus on the participant's experience of PE as a process. Adjective pairs focusing on the attractiveness or usability of the software interface were omitted, and some adjective pairs were added to elicit responses more focused on PE. The final questionnaire was provided to the participants in Finnish, and contained the following 13 adjective pairs[5], preceded by the words *Post-editing was...*: *difficult/easy*, *unpleasant/pleasant*, *stressful/relaxed*, *labourious/effortless*, *slow/fast*, *inefficient/efficient*, *boring/exciting*, *tedious/fun*, *complicated/simple*, *annoying/enjoyable*, *limiting/creative*, *demotivating/motivating*, *impractical/practical*. For analysis, we processed the scores using the formulae in the UEQ Data Analysis Tools (version 7)[6] to convert them to a scale of -3 to +3, with 0 representing a neutral mid-point. In the UEQ Data Analysis Tool, average scores between -0.8 and +0.8 are defined as neutral evaluations. Values below -0.8 correspond to negative and values above 0.8 to positive evaluations.

In addition to the PE experience, we included Likert-scale assessments for the automatic spotting and segmentation of subtitle frames (1 poor – 7 good) and the effort needed to correct them (1 easy – 7 a lot of effort). Short open questions were included for more specific comments regarding the MT output, subtitle spotting and segmentation.

4.4 Semi-structured interviews

After completing all PE tasks, a brief semi-structured interview was also carried out to collect more detailed feedback on each participant's experience, features affecting the process and usability, and possible suggestions for future development and improvements. In the interview, the participants were first asked for their overall impression of the PE tasks was, what features of the MT output affected that impression experience and whether they observed differences between the outputs. They were then asked to describe their normal subtitling process and how the MT output affected that process. Finally, the participants were asked whether they would consider using MT as a tool in their own work and what kind of improvements would be needed.

The interviews were transcribed and anonymised, and thematic analysis (see e.g. Matthews and Ross, 2010) was carried out using the analysis software Atlas.ti. The interview responses were analysed for positive and negative comments and specific issues raised by the participants,

[5] The Finnish translations provided in version 8 of UEQ were not yet available at the time of our experiment. The Finnish adjective pairs were created by the authors, and we provide here our back-translations into English.

[6] `https://www.ueq-online.org/`

such as features impacting quality and usability or suggestions for improvement. In some cases, the participant's statement was not explicitly positive or negative, but rather consisted of a neutral, generic observation or a mixed evaluation, such as a comment (sv-fi, participant A) that the MT was "sometimes surprisingly good but sometimes surprisingly bad". Such cases were labelled as mixed/neutral.

5 Results

5.1 Evaluations of User Experience

Figure 2 shows the UEQ scores for each adjective pair averaged over all participants and all clips in each language pair comparing the two MT outputs (sentence-level model and document-level model). On average, the participants appeared to describe their MTPE experience in neutral terms (values between -0.8 and +0.8). Averaged across all participants, the most negative reactions were seen for *labourious/effortless* and *limiting/creative*, although these did not cross the -0.8 threshold. No clear differences emerged between the two different MT outputs, although the participants appeared to have a slight preference for the sentence-level MT output. Similarly, no clear difference was observed for the two programme types. Overall scores for lifestyle/cultural clips were slightly higher, except in Swedish→Finnish, where the election debate clips received slightly higher scores.

Interestingly, the participants' experiences appeared to differ in different language pairs. In particular, the participants working with English→Finnish evaluated nearly all adjective pairs negatively; only *stressful/relaxed* and *complicated/simple* show neutral averages in this language pair. Responses for Swedish→Finnish were more neutral, although tending toward negative. For Finnish→Swedish, evaluations were generally neutral, except *difficult/easy* and *complicated/simple*, where averages for the document-level output crossed the 0.8 threshold to positive evaluation. Finally, for Finnish→English clearly negative scores were seen only for the adjective pair *limiting/creative*, and the sentence-level output reaches positive averages for *difficult/easy*, *stressful/relaxed*, *inefficient/efficient*, *complicated/simple* and *demotivating/motivating*.

Spotting and segmentation of subtitle frames was generally assessed as poor, and problems appeared to have been more common in the document-level output. Correcting spotting and segmentation, however, was mostly characterised as neutral or easy. The participants working with English→Finnish assessed spotting/segmentation as particularly poor and difficult to correct, which may have affected their general impression of the PE process as a whole.

5.2 Analysis of positive and negative statements in user interviews

Table 2 shows the numbers of positive, negative and mixed/neutral statements identified. Of the total 143 statements, 55% (79) were classified as negative. Positive statements accounted for 29% (42) and mixed/neutral statements for 15% (22). No differences were observed between the two MT outputs. most statements characterised MT in general without reference to specific output. In cases where a specific output was identifiable, the numbers of positive, negative and mixed statements were roughly equal between the two outputs. However, some differences can be seen between language pairs. The proportion of negative statements is higher in Swedish→Finnish and English→Finnish than in the other two language pairs. Finnish→English translators have the highest number of positive statements, and Finnish→Swedish translators the highest proportion of mixed/neutral statements.

A more detailed analysis was also conducted to identify the specific issue discussed in negative and positive statements. Most common issues involved spotting/segmentation of the subtitles, MT output quality, and the effect of MT and PE on the translator's workflow and processes. Some statements also concerned other issues like the clips and their subject matter.

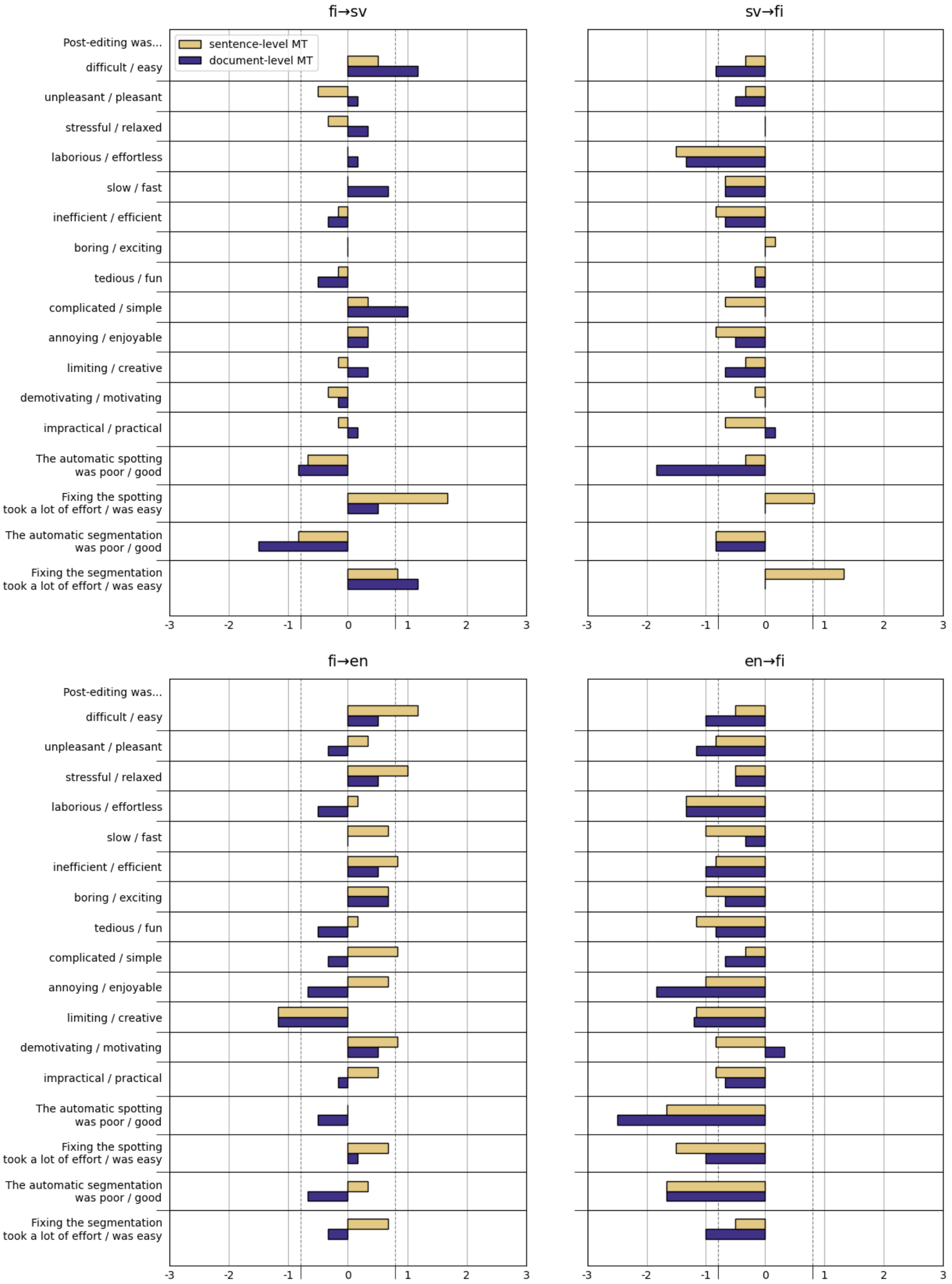


Figure 2: Average user experience scores comparing post-editing of sentence-level and document-level MT output in the language pairs Finnish→Swedish, Swedish→Finnish, Finnish→English and English→Finnish. Dashed lines represent the range [-0.8, +0.8], defined as neutral evaluations in the UEQ Data Analysis Tool.

Statement type	**en→fi**	**fi→en**	**fi→sv**	**sv→fi**	**Total**
Positive	13	16	8	5	42
Negative	23	19	14	23	79
Mixed/neutral	2	3	10	7	22
Total	38	38	32	35	143

Table 2: Positive, negative and mixed/neutral statements in the translator interviews

Most negative statements (33 out of 79) concerned the spotting or segmentation of subtitle frames, which all 12 participants commented on negatively. Specific problems involved subtitles being out of sync with the audio, and cases where a sentence had been incorrectly split into two (or more) segments. Two participants felt that the MT output tried to pack “too much” into a subtitle frame and that the machine was not able to condense the translation. Although the translations were created based on intralingual subtitles, which often already involve some condensation compared to the audio, this may suggest further differences between source and target languages. Three statements regarding the spotting were mixed/neutral, and the only two positive statements qualified spotting as “better” in some clips.

MT output quality received 30 negative mentions. Specific issues included lexical errors like mistranslated words or “odd” word choices (8 statements) and accuracy errors involving longer passages (5 statements), as well as fluency issues like ungrammatical or unidiomatic structures (6 statements). Two participants also noted omissions (words or longer passages) in the MT output. The remaining negative statements referred to MT output in general, without naming specific issues. On the other hand, the participants made 23 positive statements concerning MT quality. Specific comments referred to useful terminology and other lexical choices (9 statements) and fluency of the output (3 statements), while 11 positive statements involved general characterisations of the output as good or useful. Additionally, 13 mixed/neutral statements were made involving MT output quality in general terms.

The effect of MT and PE on the subtitling process was mentioned in 42 statements, which were mostly negative. In 15 statements, the participants commented that using MT and PE seemed to involve more effort than translation from scratch and to reduce productivity. A positive effect on productivity was mentioned in 3 statements, and 9 statements characterised the effect as mixed, sometimes reducing but sometimes increasing effort. Negative comments regarding effects also included an impression of being limited by the MT (8 statements) and potentially lower quality of the final translation (5 statements). Finally, 12 statements were made characterising the overall PE experience positively, while 5 statements described the experience negatively, and 8 in mixed/neutral terms.

5.3 User feedback for improvements

In total 28 suggestions involving development and improvement were identified in the transcripts. The most commonly mentioned improvement need was spotting/segmentation of the subtitles (8 statements). Two participants mentioned segmentation according to speaker changes as particularly useful. On the other hand, two participants would have preferred to see the MT output separately without segmentation, and one wished to see automatic speech recognition output of the original audio. Other specific issues mentioned involved need for condensing the MT output for subtitles (2 statements), improving cohesion, genre adaptation and punctuation in subtitles. Two participants mentioned the multimodal nature of AVT, one remarking that the machine is not able to take the visual aspect into account and the other wondering whether MT could use visual information.

Integration of functionalities other than MT into the subtitling software was also mentioned by some participants. Some type of terminology tool integration was mentioned in 4 statements. Some wished for a tool like a translation memory (4 statements), where one or more translators could add their own material and see how things were translated previously.

Of the participants, four would consider using MT for subtitling, although all would like to see some improvements in quality, while two participants stated they could not see themselves using MT as a tool at all. The other six gave a more mixed answer, stating that they could see MT and PE suitable for some situations but not others, for example, depending on the type of programme and subject matter. Some considered MT most useful for unfamiliar content as a terminology aid. In contrast, others would only use MT with subject matter they were already familiar with, to make sure to notice possible errors. With regard to genre, some stated MT seemed more useful for the election debates, with more formal speech, while others considered it more suitable for “simpler”, less formal language in some of the lifestyle clips.

6 Discussion and ongoing work

The subjective evaluations offer valuable insight into the user experience of MTPE for subtitling. Our participants did not find PE particularly difficult or complicated, but they tended to characterise it negatively as limiting and annoying in the questionnaire, and these themes are further present in the interviews. The translators’ feelings of MT limiting their creativity are similar to findings in studies addressing literary translation (Moorkens et al., 2018) as well as localisation (Guerberof Arenas, 2013). Translators in other studies have similarly referred to being “trapped by MT” (Bundgaard, 2017) and expressed concerns of a detrimental effect on the quality of the final translation (Moorkens et al., 2018; Matusov et al., 2019).

Both the questionnaire and the interviews point to problems in the MT subtitle alignment. Although the frame alignment (see Section 3.2) produces subtitles conforming to technical length constraints, the translators did not always find *way* the content was segmented acceptable. In some files, omissions or repetition of content in the MT also caused misalignments. The overall assessment of user experience also appears more negative in the language pairs where the participants rated the timing and segmentation poorest. This suggests that alignment problems may have affected the overall experience in addition to the MT output quality. One participant explicitly stated that dealing with off-sync subtitles probably led them to make also linguistic changes that may have been unnecessary.

In the interviews, most participants did not think MTPE increased productivity, some even felt the opposite. Similar observations have again been made in other studies; Etchegoyhen et al. (2014), for example, discuss how the increased cognitive load of dealing with MT output is a significant part of productivity. Although a detailed discussion of the process data is not within the scope of this paper, some parallels can be seen in our productivity measurements. On average, task times for MTPE were slightly faster than for translation from scratch, although considerable variation was observed between different files and participants. Five out of twelve participants were in fact slower when post-editing, concurring with the translators’ mixed experience. For a more detailed analysis of the productivity metrics, see Koponen et al. (2020).

Some care is needed when interpreting the results. Firstly, it is important to note that the participants in this study did not have prior experience with MT for subtitling (only two had previously tested it). Their responses may therefore be affected by the unfamiliarity of the task, which some participants mentioned in the interviews (see also similar observations by Bywood et al., 2017). Secondly, the participants are used to doing first translation from audio instead of working with subtitle templates. Since the MT outputs were created using intralingual subtitles as the source text, rather than the spoken language, the participants may additionally have been affected by the intralingual subtitler’s choices regarding spotting, paraphrasing and condensation. These may then have been perceived as issues in the MT output, such as omissions. More detailed communication regarding how the subtitles had been automatically generated could have clarified this issue for the participants. Finally, to allow the participants to follow their normal subtitling processes, they used their preferred subtitling software in the PE tasks. However, these tools (and AVT tools in general) are not designed for MTPE, and may therefore not be optimal for the task. This may also have affected the participants' perception of PE, and exploring AVT tools with more effective support for MTPE would be needed.

In view of our observations, it is clear that more work is needed to address the issues pointed out by the participants. It seems relevant to also compare experiences in a contrastive MTPE setting based on automatic AV transcriptions, in order to neutralise creative constraints imposed by subtitling choices carried over from intralingual subtitles. Following the interviews, we have made an effort to improve our MT pipeline in response to the segmentation and time frame alignment issues, and added support for machine-generated transcripts and time frames via automatic speech recognition and spotting. We have fixed some errors in our segmentation procedure for subtitle translations, and updated our heuristics to be less strict in enforcing length limits and clause breaks. Currently, our pipeline also makes use of an additional *restoration* stage as an endcap for MT pre-processing, implemented in practice as "intralingual translation" going from case- and punctuation-stripped input to fully-formatted output within the same language. The goal of this stage is to boost translation performance on automatic transcripts (where the MT is sensitive to differences in input formatting), and also of segmentation heuristics for post-processing (which are heavily dependent on punctuation in determining clause boundaries). Improvements to the MT and segmentation have been evaluated in further MTPE user tests during summer/fall 2020 with most of the same participants, and analysis of the data is still underway. Preliminary observations suggest somewhat more positive views of MT quality and segmentation, but the use of automatic transcriptions was received more negatively.

7 Conclusion

In this paper, we have presented a user evaluation of MTPE for subtitle translation based on experiments carried out by twelve professional subtitle translators in four language pairs (Finnish↔Swedish and Finnish↔English). Our analysis of data collected with a user experience questionnaire showed that, on average, translators' impression of MTPE varied from negative to neutral or mildly positive depending on language pair. Thematic analysis of interviews provided further information of the translators' experience. While translators did not consider PE particularly difficult, they tended to characterise it as limiting and somewhat annoying. Most, however, were open to using MT for at least some subtitling content. Further work on the quality of the outputs and tools is needed, and the translators' feedback provided valuable insight for this work. In both the questionnaire and interviews, the segmentation and timing of MT subtitles were identified as major issues, in addition to overall MT quality. As this paper reports our first user evaluations of MT for subtitling in specific language pairs and in a specific AVT context, definitive conclusions regarding the ultimate applicability of MTPE for subtitling naturally cannot yet be made. As work in this area continues, further studies on the user experience of subtitle translators are essential to investigate this question.

Acknowledgements

This work is part of the MeMAD project, funded by the European Union's Horizon 2020 Research and Innovation Programme (Grant Agreement No 780069).

References

Bundgaard, K. (2017). Translator attitudes towards translator-computer interaction - Findings from a workplace study. *Hermes– Journal of Language and Communication in Business*, 56:125–144.

Bywood, L., Georgakopoulou, P., and Etchegoyhen, T. (2017). Embracing the threat: machine translation as a solution for subtitling. *Perspectives: Studies in Translatology*, 25(3):492–508.

de Sousa, S. C., Aziz, W., and Specia, L. (2011). Assessing the post-editing effort for automatic and semi-automatic translations of DVD subtitles. In *Proceedings of RANLP 2011*, pages 97–103.

Etchegoyhen, T., Bywood, L., Fishel, M., Georgakopoulou, P., Jiang, J., Van Loenhout, G., Del Pozo, A., Maučec, M. S., Turner, A., and Volk, M. (2014). Machine translation for subtitling: A large-scale evaluation. In *Proceedings of the 9th International Conference on Language Resources and Evaluation, LREC 2014*, pages 46–53.

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

Guerberof Arenas, A. (2013). What do professional translators think about post-editing. *The Journal of Specialised Translation*, 19(19):75–95.

Junczys-Dowmunt, M. (2019). Microsoft Translator at WMT 2019: Towards large-scale document-level neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation*, pages 225–233.

Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., Seide, F., Germann, U., Aji, A. F., Bogoychev, N., Martins, A. F. T., and Birch, A. (2018). Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121.

Karakanta, A., Negri, M., and Turchi, M. (2020a). Is 42 the answer to everything in subtitling-oriented speech translation? In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 209–219, Online. Association for Computational Linguistics.

Karakanta, A., Negri, M., and Turchi, M. (2020b). MuST-cinema: a speech-to-subtitles corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3727–3734, Marseille, France. European Language Resources Association.

Koponen, M., Sulubacak, U., Vitikainen, K., and Tiedemann, J. (2020). MT for subtitling: User evaluation of post-editing productivity. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 115–124, Lisboa, Portugal. European Association for Machine Translation.

Kudo, T. and Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 EMNLP*, pages 66–71.

Laugwitz, B., Held, T., and Schrepp, M. (2008). Construction and evaluation of a user experience questionnaire. In Holzinger, A., editor, *HCI and Usability for Education and Work. USAB 2008*, volume 5298 of *Lecture Notes in Computer Science*, pages 63–76, Berlin/Heidelberg. Springer.

Leijten, M. and Van Waes, L. (2013). Keystroke logging in writing research: Using Inputlog to analyze and visualize writing processes. *Written Communication*, 30(3):358–392.

Matthews, B. and Ross, L. (2010). *Research Methods: A Practical Guide for the Social Sciences*. Pearson Education Ltd, Edinburgh.

Matusov, E., Wilken, P., and Georgakopoulou, Y. (2019). Customizing neural machine translation for subtitling. In *Proceedings of the Fourth Conference on Machine Translation*, pages 82–93.

Melero, M., Oliver, A., and Badia, T. (2006). Automatic multilingual subtitling in the eTITLE project. In *Proceedings of Translating and the Computer 28*, pages 1–18.

Moorkens, J., Toral, A., Castilho, S., and Way, A. (2018). Translators' perceptions of literary post-editing using statistical and neural machine translation. *Translation Spaces*, 7(2):240–262.

Nikolić, K. (2015). The pros and cons of using templates in subtitling. In *Audiovisual Translation in a Global Context: Mapping an Ever-Changing Landscape*, pages 192–202.

Pedersen, J. (2017). The FAR model: assessing quality in interlingual subtitling. *The Journal of Specialised Translation*, 28:210–229.

Tiedemann, J. (2008). Synchronizing translated movie subtitles. In *Proceedings of LREC'08*.

Tiedemann, J. and Scherrer, Y. (2017). Neural machine translation with extended context. In *Proceedings of the Third DiscoMT*, pages 82–92.

Varga, D., Németh, L., Halácsy, P., Kornai, A., Trón, V., and Nagy, V. (2005). Parallel corpora for medium density languages. In *Proceedings of RANLP 2005*, pages 590–596.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Volk, M., Sennrich, R., Hardmeier, C., and Tidström, F. (2010). Machine translation of TV subtitles for large scale production. In *Proceedings of the Second Joint EM+/CNGL Workshop*, pages 53–62.

A.2 EAMT paper on post-editing of subtitle MT

MT for subtitling: User evaluation of post-editing productivity

Maarit Koponen [*] **Umut Sulubacak** [*] **Kaisa Vitikainen** [†*] **Jörg Tiedemann** [*]

[*] University of Helsinki
{name.surname}@helsinki.fi

[†] Yle
{name.surname}@yle.fi

Abstract

This paper presents a user evaluation of machine translation and post-editing for TV subtitles. Based on a process study where 12 professional subtitlers translated and post-edited subtitles, we compare effort in terms of task time and number of keystrokes. We also discuss examples of specific subtitling features like condensation, and how these features may have affected the post-editing results. In addition to overall MT quality, segmentation and timing of the subtitles are found to be important issues to be addressed in future work.

1 Introduction

Developments in machine translation (MT) in the last two decades have led to significant improvements in translation quality. The success and popularity of statistical machine translation (SMT) systems were matched and eventually surpassed by neural machine translation (NMT). As quality has improved, the use of MT and post-editing (PE) has also increased in professional translation workflows. Broadly, PE refers to the practice of using MT output as a raw version checked and corrected by the translator. The use of MT and PE has been found to increase productivity in various translation scenarios (e.g. Plitt and Masselot, 2010). However, this workflow appears less common in the field of audiovisual translation (AVT). For example, Bywood et al. (2017) note that while specialised subtitling software with various functionalities are used, technologies like translation memory (TM) or MT have not been widely adopted in AVT. Matusov et al. (2019) suggest that a reason for the lower rate of MT adoption in the AVT field may be that current NMT systems are not suited for the particular features of subtitle translation.

This paper presents a pilot study carried out in November 2019 examining how the use of MT and PE in the subtitling workflow affects the work and productivity of subtitlers. In the study, 12 professional subtitle translators worked on a series of tasks in four language pairs (Finnish→English, Finnish→Swedish, English→Finnish, and Swedish→Finnish). They created interlingual (translated) subtitles for short video clips both with and without MT output. To assess productivity and effort, keylogging data were recorded during these tasks. Task time and technical effort represented by keystrokes were compared between post-editing and translation from scratch.

We first discuss related work on MT for subtitling and approaches to user evaluation of MTPE in Section 2. The MT models and subtitle alignment are presented in Section 3. Section 4 outlines the user data collection, and Section 5 presents the analysis of productivity measures. Section 6 discusses observations on PE changes, followed by future work and conclusions.

2 Related work

2.1 Machine translation for subtitling

Interlingual translated subtitles are a solution (along with dubbing and voice-overs) for bringing movies, television series, documentaries and other video material to audiences who do not understand the original language of the video. Whether dubbing or subtitling is used varies in different countries and also contexts. Finland, where this study was carried out, is one of the countries where subtitling is predominant for most content types (only children's programming tends to be dubbed).

Subtitling has some features which differentiate it from text translation. Firstly, the source text in subtitling is spoken language, or written representation of spoken language when intralingual subtitles in the language of the original video are used as the source in so-called template translation (e.g. Bywood et al., 2017). The translated subtitles represent source language speech in written target language. Secondly, subtitles have certain technical restrictions related to the number of characters and lines in one subtitle frame, and the length of time the frame is shown on the screen. For example, at the broadcasting company where this study was carried out, subtitle frames contain a maximum of two lines consisting of a maximum of 37 characters, and each frame is on screen from 2 seconds up to 6 seconds. Therefore, subtitle translation commonly involves condensation through solutions like omissions and paraphrases (Pedersen, 2017). Burchardt et al. (2016) also note that issues such as wide variation in subject matter, disfluencies and lack of context in the spoken language as well as the effect of the visual context may present additional challenges for MT.

On the other hand, some authors have suggested that the generally short and relatively simple sentences typical of subtitles would be well-suited for MT. For example, Volk et al. (2010) discuss an SMT system for Swedish→Danish MT of subtitles. In a PE experiment with 6 translators, they report relatively little was edited (average BLEU score between MT and PE for three different TV genres 65.8), with 22% of segments not changed at all. However, no process-based effort measures are reported in that study.

The eTITLE project (Melero et al., 2006) developed a web-based subtitling platform (for English, Spanish, Catalan and Czech) which offered translation memories and MT output from third-party MT engines as a tool for subtitlers. Their tool contains modules for condensation of the machine-translated subtitles and for subtitle placement. Melero et al. (2006) present a user evaluation where one translator translated parts of a movie (English→Czech) either based on the English source text or using MTPE, and report that subtitling the parts with MT was approximately 17% faster than the parts without.

In another study, de Sousa et al. (2011) experimented with MT and TM for DVD subtitling (English→Portuguese). Based on an experiment where 11 volunteers (described as "native speakers of Brazilian Portuguese and fluent speakers of English" with "some experience with translation tasks") alternately translated and post-edited 250 source sentences, de Sousa et al. (2011) report that MTPE was on average 40% faster than translation from scratch.

The SUMAT project (Bywood et al., 2017) developed a cloud-based platform for subtitle translation using MT and post-editing in multiple language pairs, and involved a large-scale user evaluation of productivity and usability of MTPE for subtitling. They collected time data and subjective feedback from 19 professional subtitle translators who translated two files using a source language template, and post-edited MT with and without quality estimation filtering. Bywood et al. (2017) found that MTPE improved productivity (in terms of task time) on average by nearly 40%, although considerable variation was observed in different language pairs and content types. They report the highest increase in English→Dutch (86%) whereas in Spanish→English, a 3.4% decrease of productivity was observed. On average, productivity increased by approximately 14% for scripted vs 50% for unscripted content (Bywood et al., 2017).

Matusov et al. (2019) customised an English→Spanish NMT system for subtitle translation using OpenSubtitles parallel data and other "conversational corpora" like GlobalVoices and TED talks. They report a user experiment where two professional translators subtitled a documentary and a sitcom episode partly from scratch and partly using a source language template and by post-editing two different MT outputs. Based on the experiments, Matusov et al. (2019) estimate average time savings by the translators to be approximately 25% with the customised MT and 5% with the baseline system.

2.2 User evaluation of MT and PE effort

Common approaches to evaluating MT quality include automatic MT metrics such as BLEU (Papineni et al., 2002) or (H)TER (Snover et al., 2006), which calculate similarity scores or edit rates based on the overlap of words or n-grams between an MT hypothesis and one or more reference translations. These metrics are sometimes used to compare MT output and post-edited versions of the MT as representation of PE effort in terms of the number of words changed during PE (e.g. Volk et al., 2010). However, this product-based approach cannot fully capture the actual effort involved in the PE process. For a more accurate picture of the feasibility of using MTPE, evaluations need to address PE effort in terms of time, technical effort required carrying out for corrections, as well as cognitive effort required for identifying errors and deciding what actions are needed (see Krings, 2001).

Temporal effort can be measured by recording task times (e.g. to the nearest minute) and comparing different types of tasks, such as MTPE versus translation "from scratch" (without MT output), or PE of different MT outputs. More fine-grained time data can be collected using keystroke logging tools like Inputlog (Leijten and Van Waes, 2013), which also provide information about the technical effort involved. Cognitive effort is the most difficult of the three to capture. Approaches to measuring cognitive effort include examining pauses in keylogging, introspective methods, and eyetracking. For an overview of process methodologies, see e.g. Saldanha and O'Brien (2013).

Like the previous studies on MT for subtitling in Section 2.1, the user evaluation reported in this paper addresses productivity in MTPE compared to translation from scratch. However, where prior work has mainly focused on task time or throughput (words or subtitles translated per time unit), we also examine technical effort through keylogging. Effort measures (task time, number of keystrokes) were analysed comparing subtitling from scratch and MT post-editing (see Section 4).

3 Automatic subtitle translation

3.1 Datasets and MT models

For the assessment of MT in subtitle translation, we created sentence-level and document-level translation models from all the parallel data available in OPUS.[1] For Finnish↔Swedish, this includes a bit over 30 million training examples,[2] and for Finnish↔English, roughly 44 million.[3] The training data comes from diverse backgrounds, with sources ranging from Bible translations to software localisation data, official EU publications, and data mined from unrestricted web crawls.

The largest portion of training data is a collection of movie and TV show subtitles derived from the OpenSubtitles (v2018) dataset. For Finnish↔Swedish, this collection contains over 15 million translation units, and for Finnish↔English, it contains almost 30 million translation units. Even though this sub-corpus is quite noisy as well, it fits the task rather well, and we can therefore expect that our models should have a decent performance in the subtitle translation task even without further fine-tuning.

The models we trained rely on the Transformer architecture (Vaswani et al., 2017), the current state of the art in NMT. We apply the implementation from the MarianNMT toolkit (Junczys-Dowmunt et al., 2018), which offers fast training and decoding with the latest features of production-ready NMT. We use the common settings of a multi-layer transformer, with 6 layers on both the encoder and the decoder, and 8 attention heads in each layer. We enable label smoothing and dropout, and use tied embeddings with a shared vocabulary, basically following the recommendations for training transformer models in the MarianNMT documentation. For text segmentation, we apply SentencePiece (Kudo and Richardson, 2018) with models that are trained independently for source and target languages for a vocabulary size of 32,000 in each language. We do not apply any further pre-processing to keep the setup as general as possible, apart from some basic normalisation of Unicode punctuation characters, and parallel corpus filtering using standard scripts from the Moses SMT package (Koehn et al., 2007).

For the document-level models, we apply the concatenative models proposed by Tiedemann and Scherrer (2017) and Junczys-Dowmunt (2019) using units of a maximum length of 100 tokens. Note that sentences and sentence fragments in subtitles are typically very short, and 100 tokens typically cover substantial amounts of context beyond sentence boundaries. We mark sentence bound-

[1] `http://opus.nlpl.eu`

[2] OPUS corpora used: bible-uedin, DGT, EMEA, EUbookshop, EUconst, Europarl, Finlex, fiskmo, GNOME, infopankki, JRC-Acquis, KDE4, MultiParaCrawl, OpenSubtitles, PHP, QED, Tatoeba, TildeMODEL, Ubuntu, wikimedia

[3] OPUS corpora used: bible-uedin, Books, DGT, ECB, EMEA, EUbookshop, EUconst, Europarl, GNOME, infopankki, JRC-Acquis, KDE4, OpenSubtitles, ParaCrawl, PHP, QED, Tatoeba, TildeMODEL, Ubuntu

aries with special tokens, chunking the training and test data sequentially from the beginning to the end without any overlaps. This procedure creates roughly 3.3 million pseudo-documents for Finnish↔Swedish and 4.7 million documents for Finnish↔English. This means that we have on average about 9 sentences per document, which are concatenated into one long string with boundary markers between sentences.

During test time, we proceed in the same way, creating pseudo-documents from the original input by concatenating subsequent sentences and splitting when a segment exceeds 100 tokens. Sentence-level models are translated in the usual way. In order to examine the translation quality, we applied our models to a dedicated test set taken from a larger set of subtitles from public broadcasts with audio in Finnish, Swedish or English. Intralingual subtitles in the language of the original audio were aligned with interlingual subtitles of the same programme in one of the other two languages. However, it should be noted that the interlingual subtitles are not direct translations of the intralingual subtitles as such. The alignment of subtitle segments in the test set was manually checked and non-corresponding segments were removed. The Finnish and Swedish parts of the dataset also contain intralingual subtitles for the deaf or hard-of-hearing, which were separated in the test set as their own subsets.

The translation results are shown in Table 1, where scores are listed separately for different subsets. Note that the document-level results need to be treated in a special way as they do not automatically match the sentence-level reference translations even when splitting on generated sentence boundary markers. To ensure that the reference and the system output correspond to each other, we apply a standard sentence alignment algorithm implemented in the hunalign package (Varga et al., 2005). We use the re-alignment flag to enable lexical matching as well, which is very beneficial in this monolingual alignment task. BLEU scores may have been negatively affected by this procedure as this alignment is not perfect.

Overall, the results indicate that document-level models seem to be beneficial in the subtitle translation case. The automatic evaluation scores consistently show an improvement over the corresponding sentence-level models for both language pairs and in all directions. However, this encouraging result unfortunately does not carry over to the manual assessment (see Section 5). A reason for this may be at least partially related to the problem of segmentation and time frame alignment, which we introduce below.

benchmark	**sentence-level**		**document-level**	
	BLEU	**chrF$_2$**	**BLEU**	**chrF$_2$**
fi→sv	18.8	0.443	19.3	0.451
sv→fi	15.7	0.449	16.8	0.462
fi→en	21.5	0.458	23.6	0.472
en→fi	16.0	0.444	17.1	0.454

Table 1: Comparison of BLEU and chrF$_2$ scores on the benchmark test set for the sentence-level and document-level systems in the language pairs Finnish→Swedish, Swedish→Finnish, Finnish→English, and English→Finnish.

3.2 Subtitle frame alignment

In both sentence-level and document-level translation, we have to treat the results in a way that maps the translations back into the time slots allocated for the original subtitles. Those time slots may include more than one sentence, and sentences may stretch over multiple time slots. Because our translation models are trained on sentence-aligned data, we need to extract sentences first from subtitles, too. We do this using the techniques proposed by Tiedemann (2008), which were also applied to the OpenSubtitles corpus in our training data.

Subtitles converted to sentence-level segments in XML:

```
  <s id="13">
    <time id="T16S" value="00:01:05,960" />
We have to make readmission agreements with other countries, -
    <time id="T16E" value="00:01:12,360" />
    <time id="T17S" value="00:01:12,440" />
so that they would be willing.
  </s>
  <s id="14">
We have to cooperate closely.
    <time id="T17E" value="00:01:17,440" />
  </s>
```

Mapped back to subtitle frames after translation:

```
16
00:01:05,960 --> 00:01:12,360
Meidän on tehtävä
takaisinottosopimuksia muiden maiden kanssa,

17
00:01:12,440 --> 00:01:17,440
jotta ne olisivat halukkaita.
Meidän on tehtävä tiivistä yhteistyötä.
```

Figure 1: Pre- and post-processing of subtitle data before and after translation. Sentences may run over several subtitle frames and multiple sentences and sentence fragments can also appear in the same time frame. The translation comes from a document-level model.

Mapping back to subtitle frames and their time allocations is implemented as another alignment algorithm. We apply a simple length-based algorithm for this, assuming that there is a strong length correlation between the source- and target-language subtitles. The difference to traditional sentence alignment is that we are now only interested in 1-to-n alignments, meaning that each existing subtitle frame in the original input should be filled with one or more segments from the translation. The segments on the target side that we consider are clauses from the generated sentences. For simplicity, we split on any punctuation in the output that is followed by space to approximate the structural segmentation. We then apply the traditional Gale & Church algorithm (Gale and Church, 1993) to optimise the global alignment between source segments (original subtitle frame data) and target segments. For this, we adjust the parameters of the algorithm in two ways: (i) we remove priors and apply a uniform distribution over possible alignment types, and (ii) we change the set of alignment types to include all possible mappings from one source segment to a maximum of four target segments. The mapping between source and target is then created using the original algorithm that ensures a globally optimal mapping according to the model (see Figure 1 for an example). Furthermore, we apply simple heuristics to insert line breaks in order to make subtitles conform to length and formatting constraints. The implementation of the entire procedure is available as an open source package[4].

4 User PE data collection

The subtitling tasks for productivity data collection were carried out in November 2019 at the premises of the Finnish Broadcasting Company Yle. In total 12 translators (3 per language pair) participated in the tasks: 8 in-house translators and 4 freelancers with experience of working for Yle. The participants have between 4 and 30 years of professional subtitling experience in their language pair. Only 2 stated they had previously used MT for subtitling, and 7 others had used MT for other purposes.

The subtitling tasks were carried out using the subtitlers' preferred software (Wincaps Q4 or Spot). To replicate their normal working environment, an external monitor and keyboard were provided, and they had access to the internet as well as terminology and other resources normally used in their work. Process data were logged using Inputlog (Leijten and Van Waes, 2013), which records all keyboard and mouse activity. Windows 10 screen recording software was used to capture video to support the analysis. Pre- and post-task questionnaires were used to collect background information and participants' subjective assessment of the MT output and PE experience. After the tasks, a brief semi-structured interview was also carried out to collect more detailed feedback regarding problems in the workflow and the participants' views on potential improvements. In this paper, we focus on an analysis of the process data.

Subtitling tasks were carried out in 4 language pairs: Finnish→English, Finnish→Swedish, English→Finnish, and Swedish→Finnish. For each source language, six clips were selected from a dataset provided by Yle. Three clips were selected from unscripted European election debates, and three clips from semi-scripted lifestyle or cultural programmes. The individual clips were selected so that each clip (i) forms a coherent, self-contained section of the programme, (ii) is approximately 3 minutes long, and (iii) contains 30–35 subtitle segments.

Each participant completed a total of six tasks where they subtitled two clips "from scratch" without MT output, two clips using output from a sentence-level MT system, and two clips using output from a document-level MT system. The clips and MT outputs were rotated in a round-robin format so that each clip was subtitled once in each condition (no MT output, sentence-level MT output, document-level MT output) by a different participant. Task order was also varied to minimise facilitation effect. The participants were instructed to produce subtitles that would be acceptable for broadcasting, and to use the resources they normally would for their work, but to not spend excessive time in "polishing" any given wording or researching information. No explicit time limit was given for each task, rather, the participants were instructed to work at their own pace.

In the from scratch condition, the participants also created the segmentation and timing of the subtitles following their normal work process. Subtitling templates are not used by Yle for these content types. In the MTPE condition, the participants worked with output that was pre-segmented and timed based on the intralingual subtitles used as source text for the MT (see Section 3.2).

To assess productivity, the process logs were analysed using Inputlog's analysis functions. The

[4] https://github.com/Helsinki-NLP/subalign

task time and the number of keystrokes logged were used as productivity measures. Using Inputlog filters, we focused only on task time and keystrokes in the subtitling software, excluding other activity such as internet searches for terminology or other information. Based on the final subtitles produced, edit rate between the MT output and the final versions were calculated using HTER (Snover et al., 2006) and characTER (Wang et al., 2016). As PE of the subtitles involved also changes to the segmentation, e.g. adding or deleting frames and moving words between frames, subtitle segmentation was ignored and edit rates were calculated as document-level scores to focus on edits affecting the textual content. These measures were then compared between the tasks of creating interlingual subtitles from scratch and MTPE, as well as between PE of the sentence-level and document-level MT outputs described in Section 3.1.

5 Comparison of subtitling productivity

Figure 2 shows a comparison of the average subtitling task time for subtitling from scratch and subtitling with MTPE. The topmost three bars show averages for post-editing the sentence- and document-level MT output and for translation from scratch across all language pairs, while the bottom pairs of bars show averages for PE (either MT output) compared to from scratch. On average, post-editing machine-translated subtitles (regardless of MT output) was slightly faster than creating subtitles from scratch. Some differences can be seen between the language pairs: the largest difference in task times is seen in Swedish→Finnish, while the task times for Finnish→English and Finnish→Swedish are nearly equal. No clear difference could be observed between the two different MT outputs, although on average post-editing the sentence-level MT output appeared to be slightly faster.

Figure 3 shows a comparison of technical effort in terms of the average number of keystrokes used when producing subtitles. The topmost three bars show averages for post-editing the sentence- and document-level MT output and for translation from scratch across all language pairs, while the bottom pairs of bars show averages for PE (either MT output) compared to from scratch. On average, post-editing machine-translated subtitles (regardless of MT output) involved fewer keystrokes than creating subtitles from scratch. The reduction in the number of keystrokes is more pronounced than in the case of task times, and seen in all language pairs. Again, no clear difference could be observed between the two different MT outputs, although on average post-editing the sentence-level MT output appeared to involve slightly less technical effort.

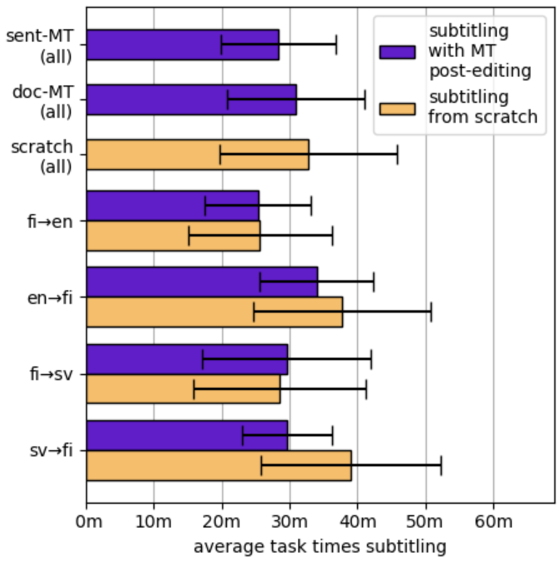


Figure 2: Average task times subtitling through post-editing and from scratch. The top three bars show averages for post-editing sentence- and document-level MT, and subtitling from scratch. The bottom pairs of bars are averages for each language pair. Error bars indicate standard deviation.

Although a detailed analysis of the types of keystrokes is not within the scope of this paper, some observations can be made regarding the distribution of keystroke types. Intuitively, PE reduced the need for text producing keystrokes on average by 54% compared to from scratch, as the MT output provides some of the text needed. However, the number of text deleting keystrokes was 24% higher in PE, as correcting the output also involves removing words or characters. In the from scratch case, the participants needed to create and set the timing for each subtitle frame themselves, which requires keystrokes and/or mouse clicks. In MTPE, the MT output was already segmented and timed based on the intralingual subtitles used as source text, which reduced the associated keystrokes by approximately 32%, but the number of keystrokes shows that the participants found it necessary to change both the segmentation and timing. Changes to subtitle segmentation are discussed in more detail below.

To examine the number of changes between

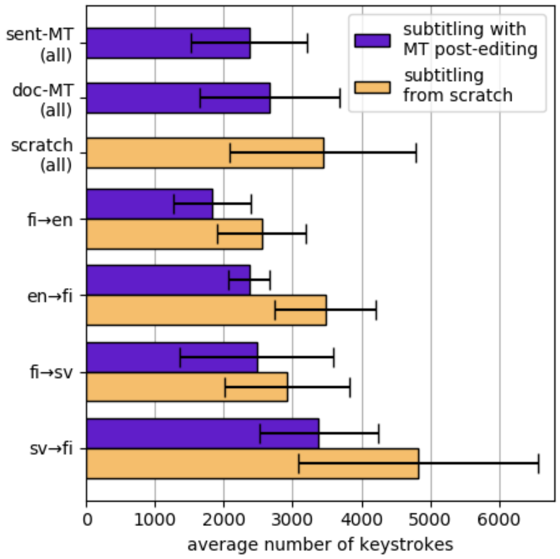


Figure 3: Average numbers of keystrokes subtitling through post-editing and from scratch. The top three bars show averages for post-editing sentence- and document-level MT, and subtitling from scratch. The bottom pairs of bars are averages for each language pair. Error bars indicate standard deviation.

the MT outputs and final PE versions, edit rates were calculated using word-based HTER and character-level characTER. Table 2 shows the HTER and characTER scores for the sentence-level and document-level MT across all four language pairs and for each language pair. The high edit rates (overall average HTER 57.7 and characTER 46.8) indicate considerable rewriting during PE, particularly in the case of English→Finnish. The high HTER score in this language pair may be due to the fact that word-based metrics do not distinguish changed words and changed word forms, which are common in morphologically-rich target languages like Finnish. The considerable difference in the characTER and HTER scores in English→Finnish suggests word form edits are indeed more common in this language pair. However, a similar effect is not seen in Swedish→Finnish. A preliminary analysis of the edits indicates that the participants working on this language pair have added words more frequently than participants in other language pairs. Corresponding to the process metrics, average edit rate for the sentence-level MT output is slightly lower than for the document-level MT. At least partly, this may be explained by the observation that repetition of words or phrases was more common in the document-level MT output.

In addition to the textual content of the MT subtitles, the participants edited both the segmentation of that content into subtitle frames and timing of the frames. On average, the participants increased the number of subtitle frames in the clips by 7% by splitting or adding frames. This tendency was particularly noticeable in Swedish→Finnish (+19%). English→Finnish was the only language pair where the participants reduced the number of subtitle frames (–4%) for example by joining and condensing the textual content of the frames. Comparing the timestamps of the original subtitle frames used for the MT output and the frames in the post-edited files, we observed that only 24% of the original timed frames had been retained in PE. For 27% of frames, either the in or out time had been changed, and for 49% both in and out time were changed.

	HTER	**characTER**
sent-level	55.1 ± 17.7	45.0 ± 12.3
doc-level	60.3 ± 16.1	48.7 ± 11.1
fi→en	45.6 ± 17.7	39.3 ± 13.5
en→fi	74.1 ± 12.7	48.9 ± 6.4
fi→sv	52.7 ± 13.2	44.1 ± 11.4
sv→fi	58.4 ± 9.5	55.1 ± 9.2
overall	**57.7 ± 16.9**	**46.8 ± 11.8**

Table 2: Comparison of word-level (HTER) and character-level (characTER) edit rates divided by MT system (sentence-level vs document-level) and language pair (Finnish→English, English→Finnish, Finnish→Swedish, Swedish→Finnish).

The intralingual subtitles used as source text were not translated as isolated subtitle frames but rather as sentences or longer passages and then aligned back to the frames (see Section 3.2). However, the heuristics used for alignment were not always successful. In some cases, splitting a segment due to punctuation caused the next segment to become too long and started to push content into the following frames, causing the subtitles to fall out of sync with the audio. Similar issues were also observed due to repetition in the MT output. It is also possible that the sync issues arising from incorrect segmentation may have lead the participants to also change the timing of subtitle frames.

6 Discussion of PE changes

Considerable variation in task times and numbers of keystrokes was observed between different participants. Productivity gains were most evident for participants with the longest average task times overall. However, 5 out of the 12 participants were in fact slower in PE. Two of them also used slightly more keystrokes, but three were slower despite using fewer keystrokes in PE. These findings are similar to other process studies both on subtitling and other text types (e.g. Plitt and Masselot, 2010; Bywood et al., 2017) showing that potential productivity gains from MTPE vary, and that participants who are already fast benefit less. Fewer keystrokes not necessarily leading to time saving has also been observed in other studies. While the number of keystrokes reflects the technical effort needed, it does not capture the amount of cognitive effort involved in recognising potential errors and deciding on necessary changes.

The edit rates of different participants also vary. At the level of individual subtitlers, average HTER scores range from 31.9 (Finnish→English, participant C) to 84.8 (English→Finnish, participant C). These edit rates are comparable to the HTER scores reported by Matusov et al. (2019) for different MT system outputs, genres and post-editors, which range from 27.8 to 82.7. In our study, the two participants with the highest average edit rates both worked on English→Finnish, and the two with the lowest average edit rates on Finnish→English, but differences are also evident within the same language pair. Since the participants post-edited different MT versions, some variation may be explained by different output quality, but to some extent these differences may also reflect individual preferences. Qualitative observations suggest that while some edits relate to clear MT errors, many are also caused by what appear to be preferential edits; for example, in the Finnish→English clips, one participant accepts the translation “financial discipline” for the Finnish *talouskuri* while another replaces it with “austerity”.

A possible factor affecting both productivity and number of changes is PE experience. The participants in this study had little prior experience with MT specifically for subtitling. The subtitlers’ productivity and approach to the task may therefore have been affected by the fact that PE was unfamiliar and different from their normal work processes. As Bywood et al. (2017) also note, psychological factors such as unfamiliarity and irritation with MT errors influence productivity. These factors may have also led to preferential and possibly unnecessary changes. More practice working with MT output and pre-segmented subtitles may affect their approaches, e.g. by reducing preferential changes, and increase productivity in this task.

As noted in Section 2.1, the spoken content of the videos and subtitles as a written representation of spoken language differ from each other. Due to technical restrictions, condensation is common in subtitle translation, and may affect the edit rate to some extent. On the other hand, because the source text for the subtitlers consists of not only the written subtitles, but also the audiovisual context, they may make changes based on information in the audio or video of the clip being subtitled.

An example of condensation through omission and paraphrasing can be seen in Table 3, where the participant has combined two subtitle frames (0001 and 0002) in the intralingual subtitles and the MT. This type of condensation was observed particularly in English→Finnish, where the participants reduced the number of subtitle frames.

In contrast to condensation, the participants sometimes added content to subtitles. While some additions correspond to missing words in the MT output, others in fact involve content not present in the intralingual subtitles used as source text for MT. The intralingual subtitles themselves already involve some condensation and paraphrasing, and therefore do not match exactly the spoken audio. Particularly in the Swedish “lifestyle” clips, the intralingual subtitles appear to have been very condensed, and the participants post-editing Swedish→Finnish added both textual content and new subtitle frames. These additions show one effect of the multimodal context: having the omitted information present in the audio led the participants to make additions that would have been unlikely or impossible if only the written subtitles had been available.

Subtitle translators are also affected by the visual context of the video. Changes related to the visual context occur, for example, when the subtitler chooses to replace a pronoun with the referent seen in the video. An example of this appears in one of the Swedish→Finnish clips involving cooking. The expression *de ska kokas mjuka* ‘they should be cooked soft’ in the dialogue is correctly translated in both MT outputs using the Finnish pronoun *ne* ‘they’. However, both participants post-editing MT output for this clip replaced the pronoun with *hedelmät* ‘fruit’, referring to the fruit being cooked.

Source	MT output (doc)	Post-edited
0001 00:00:00:00 00:00:02:24 Viikonloppuna on vaalitarkkailijoita -	0001 00:00:00:00 00:00:02:24 There will be election observers this weekend -	0001 00:00:00:00 00:00:04:17 There are more election observers there than ever before.
0002 00:00:00:00 00:00:02:24 enemmän kuin ehkä missään muissa vaaleissa	0002 00:00:00:00 00:00:02:24 more than there may be in any other election.	

Table 3: An example of condensation of subtitle content by a post-editor.

These observations suggest that not all changes during PE correspond to MT errors. However, a detailed analysis of the MT outputs and changes carried out during PE would be needed to establish to what extent changes relate to MT errors, subtitling features like condensation, or preferential edits.

7 Future work

Based on the experiment and user feedback, segmentation of the interlingual subtitle content into appropriate chunks is an important issue to be addressed, although using subtitle timing from pre-existing intralingual subtitles was to some extent useful. Potential directions for improving segmentation and timing could involve the use of time information to split the data into coherent blocks separated by significant breaks, and the integration of speaker information into the translation engines to segment subtitles into dialogue turns by leveraging speaker labels or diarisation output. Multimodality can also play a crucial role in segmentation as visual and auditory cues may help in improving the division of verbal content into discourse units. We plan to implement an end-to-end system for subtitle translation and segmentation after Matusov et al. (2019), and investigate how well such a system could generate organic subtitles.

Multimodality may also be useful in optimising translation quality. Augmenting subtitles with information from the visual and auditory modalities could help improve translation accuracy in general. For example, visual information could be helpful in resolving ambiguity. In future work, we will explore incorporating multimodal features in translation in connection with non-linguistic context for language grounding and disambiguation.

A more detailed manual analysis of the types of PE changes made by the participants and their potential explanations (MT errors, subtitling conventions, or preferential changes) is currently underway. Feedback collected from the participants is also being analysed for information regarding the user experience. A second round of user evaluations is also planned for 2020 to collect further data and assess the effect of the new developments of our MT approaches, and to give the participants more experience with post-editing subtitles.

8 Conclusion

This paper presented a user evaluation pilot study of MT and post-editing for subtitles. Based on an analysis of process data collected from 12 professional subtitlers in four language pairs, we presented a comparison of productivity in terms of task time and number of keystrokes when post-editing MT subtitles vs translating from scratch. On average, our results indicate MTPE to be slightly faster and to involve fewer keystrokes than subtitling from scratch. However, considerable variation was observed between different language pairs and participants. We also discussed examples of specific subtitling features like condensation, and how these features may have affected the post-editing results. In addition to overall MT quality, the segmentation and the timing of the subtitles were found to be important issues to be addressed in future work.

Acknowledgments

This work is part of the MeMAD project, funded by the European Union's Horizon 2020 Research and Innovation Programme (Grant Agreement No 780069).

References

Burchardt, A., Lommel, A., Bywood, L., Harris, K., and Popović, M. (2016). Machine translation quality in an audiovisual context. *Target*, 28(2):206–221.

Bywood, L., Georgakopoulou, P., and Etchegoyhen, T. (2017). Embracing the threat: machine translation as a solution for subtitling. *Perspectives: Studies in Translatology*, 25(3):492–508.

de Sousa, S. C., Aziz, W., and Specia, L. (2011). Assessing the post-editing effort for automatic and semi-automatic translations of DVD subtitles. In *Proceedings of RANLP 2011*, pages 97–103.

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

Junczys-Dowmunt, M. (2019). Microsoft Translator at WMT 2019: Towards large-scale document-level neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation*, pages 225–233.

Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., Seide, F., Germann, U., Aji, A. F., Bogoychev, N., Martins, A. F. T., and Birch, A. (2018). Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121.

Koehn, P., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., and Moran, C. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th ACL*.

Krings, H. P. (2001). *Repairing texts: Empirical investigations of machine translation post-editing process*. The Kent State University Press, Kent, OH.

Kudo, T. and Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 EMNLP*, pages 66–71.

Leijten, M. and Van Waes, L. (2013). Keystroke logging in writing research: Using Inputlog to analyze and visualize writing processes. *Written Communication*, 30(3):358–392.

Matusov, E., Wilken, P., and Georgakopoulou, Y. (2019). Customizing neural machine translation for subtitling. In *Proceedings of the Fourth Conference on Machine Translation*, pages 82–93.

Melero, M., Oliver, A., and Badia, T. (2006). Automatic multilingual subtitling in the eTITLE project. In *Proceedings of Translating and the Computer 28*, pages 1–18.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th ACL*, pages 311–318.

Pedersen, J. (2017). The FAR model: assessing quality in interlingual subtitling. *The Journal of Specialised Translation*, 28:210–229.

Plitt, M. and Masselot, F. (2010). A productivity test of statistical machine translation post-editing in a typical localisation context. *The Prague Bulletin of Mathematical Linguistics*, 93:7–16.

Saldanha, G. and O'Brien, S. (2013). *Research Methodologies in Translation Studies*. Routledge, London and New York.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of AMTA 2006*.

Tiedemann, J. (2008). Synchronizing translated movie subtitles. In *Proceedings of LREC'08*.

Tiedemann, J. and Scherrer, Y. (2017). Neural machine translation with extended context. In *Proceedings of the Third DiscoMT*, pages 82–92.

Varga, D., Németh, L., Halácsy, P., Kornai, A., Trón, V., and Nagy, V. (2005). Parallel corpora for medium density languages. In *Proceedings of RANLP 2005*, pages 590–596.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Volk, M., Sennrich, R., Hardmeier, C., and Tidström, F. (2010). Machine translation of TV subtitles for large scale production. In *Proceedings of the Second Joint EM+/CNGL Workshop*, pages 53–62.

Wang, W., Peter, J.-T., Rosendahl, H., and Ney, H. (2016). CharacTer: Translation edit rate on character level. In *Proceedings of the First Conference on Machine Translation*, pages 505–510.

A.3 EAMT paper on OPUS-MT

OPUS-MT – Building open translation services for the World

Jörg Tiedemann
Department of Digital Humanities
HELDIG
University of Helsinki

Santhosh Thottingal
Wikimedia Foundation

1 Introduction

Equality among people requires, among other things, the ability to access information in the same way as others independent of the linguistic background of the individual user. Achieving this goal becomes an even more important challenge in a globalized world with digital channels and information flows being the most decisive factor in our integration in modern societies. Language barriers can lead to severe disadvantages and discrimination not to mention conflicts caused by simple misunderstandings based on broken communication. Linguistic discrimination leads to frustration, isolation and racism and the lack of technological language support may also cause what is known as the *digital language death* (Kornai, 2013).

Machine translation (MT) has developed into a useful tool that diminishes and partially removes such language barriers. Modern MT engines enable people to communicate, to access information in foreign languages and to build efficient resources for new communities. The mission of OPUS-MT[1] is to provide open translation services and tools that are free from commercial interests and restrictions. The idea is to make automatic translation accessible for anyone in a transparent and secure way without exploitation plans and hidden agendas compromising privacy and placing marketing strategies. We also want to focus on the support of minority and low-resource languages with the aim to introduce a community effort for the benefit of all.

OPUS-MT has successfully launched its first pilot system and currently collaborates with the Wikimedia foundation in the setup of translation services for the production of Wikipedia content in new languages based on more elaborated resources available in, e.g. English. Currently, the project provides over 1,000 pre-trained translation models that are free to download and use. OPUS-MT also contains open-source software for launching translation services as web applications. The on-going effort focuses on the improvement of translation quality, language coverage and emphasizes specific test cases to study the applicability of the approach. More details about the implementation and current status of the project are given below.

2 OPUS-MT models

The models that we train are based on state-of-the-art transformer-based neural machine translation (NMT). We apply Marian-NMT[2] in our framework, a stable production-ready NMT toolbox with efficient training and decoding capabilities (Junczys-Dowmunt et al., 2018). Our models are trained on freely available parallel corpora collected in the large bitext repository OPUS[3] (Tiedemann, 2012). The architecture is based on a standard transformer setup with 6 self-attentive layers in both, the encoder and decoder network with 8 attention heads in each layer. The hyper-parameters follow the general recommendations given in the documentation of the software. All the details can be seen in the training procedures that we also release as open source in our GitHub repository.[4]

OPUS-MT supports both, bilingual as well as multilingual models. For the latter, we apply the language label approach proposed by (Johnson et al., 2017). Our package implements generic

[1] `https://github.com/Helsinki-NLP/Opus-MT`

[2] `https://marian-nmt.github.io`

[3] `http://opus.nlpl.eu`

[4] `https://github.com/Helsinki-NLP/Opus-MT-train`

model	BLEU	chrF$_2$
English–Finnish	22.9	0.548
+ back-translation	23.7	0.562
+ fine-tuning	25.7	0.578

Table 1: Test results for the English–Finnish OPUS-MT model based on the news translation task from WMT 2019. Fine-tuning was done using the English–Finnish news translation test sets from earlier years.

procedures that make it easy to train a large number of translation models from the existing data in the OPUS collection. The procedures take care of proper pre-processing and training setups to enable batch-processes without the immediate need for further adjustments. We try to reduce the burden of time-consuming optimization and focus on rather generic models for the time being in order to quickly achieve a good language coverage without significantly compromising translation quality that can be achieved.

We use common benchmarks and test sets that are extracted on the fly from held-out data to monitor the quality of the NMT models. Test sets and results are released together with the models, pre- and post-processing scripts and basic information about their usage. The table of currently supported language pairs can be accessed on-line.[5]

We also develop generic fine-tuning and data augmentation procedures that can be used to further improve the translation models. We implemented a pipeline for backtranslation of Wikimedia content (coming from Wikipedia, Wikibooks, Wikisource, etc.) to augment existing training data. Backtranslation is known to significantly boost performance and to enable simple domain adaptation based on in-domain target language data. Furthermore, we also provide procedures for fine-tuning that can adjust model parameters according to some small in-domain data set, another successful strategy for domain adaptation. The impact of fine-tuning and backtranslation can be seen on the example of the English–Finnish OPUS-MT model listed in Table 1.

3 OPUS-MT servers

Finally, we also provide simple web applications that can be used to launch translation services based on the pre-trained models. The most straightforward setup is implemented as a dockerized Tornado-besed web application that can be set up with a few simple commands. The configuration can be adjusted and extended to serve any bilingual translation model that we provide. Each service can accommodate several language pairs and may connect multiple servers. The current implementation is based on CPU-based decoding as a cost-efficient setup for every-day users but it should be adjustable to a GPU-based setup without major changes. A running service demonstrating the app is hosted by the Wikimedia foundation at `https://opusmt.wmflabs.org`.

Another websocket based application is also provided, which enables the support of multilingual models, a simple translation cache and the retrieval of token alignment information, which is supported by most models that we train with the guided alignment feature of Marian-NMT. Further improvements of the web applications are planned once we have finished our tests of the current implementation in a production environment for selected test cases.

4 Conclusions

This paper presents OPUS-MT, a project that focuses on the development of free resources and tools for machine translation. The current status is a repository of over 1,000 pre-trained neural MT models that are ready to be launched in on-line translation services. For this we also provide open-source implementations of web applications that can run efficiently on average desktop hardware with a straightforward setup and installation.

The work is supported by the European Union's Horizon 2020 projects FoTran and MeMAD as well as by the Finnish IT Center for Science, CSC.

References

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., et al. (2017). Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association of Computational Linguistics*, 5(1):339–351.

Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., Seide, F., Germann, U., Aji, A. F., Bogoychev, N., Martins, A. F. T., and Birch, A. (2018). Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121.

Kornai, A. (2013). Digital language death. *PloS one*, 8(10). :e77056.

Tiedemann, J. (2012). Parallel data, tools and interfaces in OPUS. In *Proceedings of LREC*, Istanbul, Turkey.

[5] `http://opus.nlpl.eu/Opus-MT/`

A.4 MeMAD submission to the IWSLT shared task

The University of Helsinki submission to the IWSLT2020 Offline Speech Translation Task

Raúl Vázquez, Mikko Aulamo, Umut Sulubacak, Jörg Tiedemann

University of Helsinki
{name.surname}@helsinki.fi

Abstract

This paper describes the University of Helsinki Language Technology group's participation in the IWSLT 2020 offline speech translation task, addressing the translation of English audio into German text. In line with this year's task objective, we train both cascade and end-to-end systems for spoken language translation. We opt for an end-to-end multitasking architecture with shared internal representations and a cascade approach that follows a standard procedure consisting of ASR, correction, and MT stages. We also describe the experiments that served as a basis for the submitted systems. Our experiments reveal that multitasking training with shared internal representations is not only possible but allows for knowledge-transfer across modalities.

1 Introduction

An effective solution for performing spoken language translation (SLT) must deal with the evident challenge of transferring the implicit semantics between audio and text modalities. An end-to-end SLT system must hence appropriately address this problem while simultaneously performing accurate machine translation (MT) (Sulubacak et al., 2018).

In last year's IWSLT challenge, both end-to-end and cascade systems yielded similar results (Niehues et al., 2019). It follows that this year's IWSLT offline speech translation challenge focuses on whether *"the cascaded solution is still the dominant technology in spoken language translation"* (Ansari et al., 2020). For our participation on this task, we train both cascade and end-to-end systems for SLT. For the end-to-end system, we use a multimodal approach trained in a multitask fashion, which maps the internal representations of different encoders into a shared space before decoding. For the cascade approach, we use a pipeline of three stages: (i) automatic speech recognition (ASR), (ii) punctuation and letter-case restoration, and (iii) MT.

We focus on exploiting the knowledge-transfer capabilities of a multitasking architecture based on language-specific encoders-decoders (Lu et al., 2018; Schwenk and Douze, 2017; Luong et al., 2016). This idea has been proposed and studied in the multilingual scenario (Vázquez et al., 2020; Subramanian et al., 2018; Firat et al., 2017), however, we adapt it to be used in a multimodal scenario. Regarding different modalities (in this case, audio and text) as different languages when training the model, allows us to employ a cross-modal intermediate shared layer for performing SLT in an end-to-end fashion. By jointly training this layer, we aim for the the model to combine the semantic information provided in the text-to-text MT tasks with the ability to generate text from audio in the ASR tasks.

2 Proposed Systems

End-to-end SLT

We use an inner-attention based architecture proposed by Vázquez et al. (2020). In a nutshell, it follows the conventional structure of an encoder-decoder model of MT (Bahdanau et al., 2015; Luong et al., 2016) enabled with multilingual training by incorporating language-specific encoders and decoders trainable with a language-rotating scheduler (Dong et al., 2015; Schwenk and Douze, 2017), and an intermediate shared inner-attention layer (Cífka and Bojar, 2018; Lu et al., 2018). We implement our model on top of an OpenNMT-py (Klein et al., 2017) fork, which we make available for reproducibility purposes.[1]

The text encoders and the decoders (always text output) are transformers (Vaswani et al., 2017).

[1] https://github.com/Helsinki-NLP/OpenNMT-py/tree/iwslt2020

Proceedings of the 17th International Conference on Spoken Language Translation (IWSLT), pages 95–102
July 9-10, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

We implement the transformer-based audio encoders inspired by the SLT architecture with tied layer structure from Tu et al. (2019) and the R-Transformer from Di Gangi et al. (2019b). It consists of n CNN layers; the first one taking k stacked Mel filterbank features as input channels, and the following ones 32 input channels. Afterwards, a linear layer corrects the shape of the embeddings and is concatenated with the positional embeddings to be fed as input to m transformer layers.

Given the multimodal nature of the task, we modified the source-target rotating scheduler. Instead of a uniform distribution over the language pairs, we propose using a weighted sampling scheme based on the inverse of the batch size of the modalities. This modification allows us to have a more balanced training because audio inputs tend to be considerably longer than text inputs, and a transformer-based encoder could not possibly handle the 4096 tokens conventionally used as the ad-hoc choice of batch size for a text-based transformer.

Cascade approach

The ASR stage of our pipeline is trained with an S-Transformer (Di Gangi et al., 2019b); an adaptation of the transformer architecture to end-to-end SLT. The encoder in this architecture makes it possible to process audio features. It consists of two 2-dimensional CNN-blocks meant to downsample the input, followed by two 2-dimensional self-attention layers to model the long-range context, an attention layer that concatenates its output with the positional encodings of the input, and six transformer-based layers.

The output of the ASR stage is followed by the **restoration stage** for punctuation and letter case restoration. Since the training data for the ASR model mixes different training sets with different formatting, the raw output from the ASR block can have stylistic differences from the input seen during the training of the translation stage. The restoration stage involves the use of an auxiliary transformer-based MT model to perform "intralingual translation" from lowercased text without punctuation into fully-cased and punctuated text. Stripping punctuation on the ASR output, converting the text to lowercase, and processing the result through the restoration stage ensures that the output conforms to the same format that the translation stage was optimized for.

As the last step, **the translation stage** uses another transformer to translate the processed ASR output to German. Both this transformer model and the one used in the restoration stage are based on the freely available Marian NMT implementation (Junczys-Dowmunt et al., 2018). Our configuration uses a learning rate of 0.0003 with linear warmup through the first 16 000 batches, decaying afterwards. The decoder normalizes scores by translation length (normalization exponent of 1.0) during beam search. All other options use the default values.

3 Data Preprocessing

The MT, ASR and end-to-end SLT systems have been trained on different subsets of the allowed training corpora. For the cascade approach SLT system

Corpora	# utterances	Length
Europarl-ST	40,141	89 hrs
IWSLT2018	166,214	271 hrs
How2	189,366	297 hrs
MuST-C	264,036	400 hrs
Mozilla Common Voice	854,430	1,118 hrs

Table 1: Size of audio data used.

Data for the end-to-end SLT system. We use Europarl-ST (Iranzo-Sánchez et al.), IWSLT2018 (Niehues et al., 2019) and MuST-C (Di Gangi et al., 2019a), a total of 433k utterances after cleaning some corrupt files or with other problems in the sampling. We extracted 80-dimensional Mel filterbank features for each sentence-like segment using our own implementation.

Text data for the end-to-end SLT system. For the text data of the multimodal end-to-end SLT system, we use a total of ~51M sentence pairs from corpora specified in Table 2. Instead of using all of this data, we first filter out noisy translations. OpenSubtitles2018, which consists of subtitle translations, and corpora gathered by crawling the internet, Common Crawl and ParaCrawl, are especially likely to contain noisy data. For filtering the corpora, we utilize OpusFilter (Aulamo et al., 2020), a toolbox for creating clean parallel corpora.

First, we extract six feature values for each of the sentence pairs. In particular, we apply the following features: CharacterScore, CrossEntropy, LanguageID, NonZeroNumeral, TerminalPunctuation and WordAlign, each of which is defined in Aulamo et al. (2020). Secondly, we train a logistic regression classifier based on those features. The classifier is trained only on WIT^3, MuST-C, Europarl-ST and IWSLT18, which are multimodal datasets with speech-to-text and text-to-text data. This allows the system to adapt to text translations that are associated with speech translations. Finally, we use the classifier to assign a cleanness score ranging from 0 to 1 for all sentence pairs in all corpora. The data is then ranked based on the cleanness score, after which a portion of noisy pairs is removed from the tail. Our preliminary translation experiments showed that removing up to 40% of the data improves the translation quality, leaving us ∼30.5M sentence pairs of training data, which are then used in all our end-to-end experiments.

Corpora	**# sentences**
WIT^3	196,112
MuST-C train	229,703
Rapid 2019	1,480,789
Europarl v9	1,817,763
OpenSubtitles2018	11,621,073
News Commentary v14	365,340
Common Crawl	2,399,123
Europarl-ST	32,628
WikiTitles	1,305,078
IWSLT2018	171,025
ParaCrawl v3	31,360,203
Total	50,978,837
Filtered	30,540,267

Table 2: Text training data used for end-to-end systems.

Audio for the cascade system. We have extracted 40-dimensional Filterbank features with speaker normalization for each sentence-like segment of the MuST-C, How2 (Sanabria et al., 2018) and Mozilla Common Voice (Ardila et al., 2019) corpora using XNMT (Neubig et al., 2018). After getting rid of audio files that were too short (less than 0.4 seconds), corrupted, or no longer available for download from YouTube, some 1.2M clean utterances remained for training the ASR system, and 30k for validation.

On the target side, we use two contrastive preprocessing pipelines:

i) the same subword segmentation used for the MT system

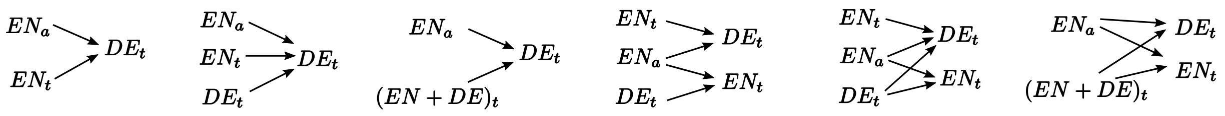
```
_it _& apos ; s _a _lobster _made
_of _play d ough _that _& apos ; s
_afraid _of _the _dark _.
```

ii) character level segmentation

```
I t <space> ' s <space> a <space>
l o b s t e r <space> m a d e <space>
o f <space> p l a y d o u g h <space>
t h a t <space> ' s <space> a f r a i d
<space> o f <space> t h e <space>
d a r k <space> .
```

Text data for the cascade system. In our SLT pipeline, the data we applied for our restoration and translation models have some overlap and some differences. For training, both models use the text data from the IWSLT 2018 speech translation corpus, the MuST-C training set, News Commentary v14, Europarl v9, and Rapid 2019. The translation model also uses data from the OpenSubtitles2018 dataset, which the restoration model does not since this dataset is particularly noisy in terms of punctuation and letter cases. Conversely, the restoration model also uses data from the How2 and Mozilla Common Voice datasets, which the translation model does not use as they do not contain German text. The translation model uses the IWSLT development set from 2010 and test sets from 2011–2015 as validation data, while the restoration model uses them as supplementary training data in order to reinforce domain bias, using only the MuST-C development set for validation.

Initially, we "clean" the output of our ASR model to remove segments containing musical note characters (♫♪), and repeating phrases that were consistently hallucinated during silence, applause, laughter or noise in the audio (e.g. in our case, "Shake. Fold."), as well as parts of segments that designate the speaker (e.g. "Audience: ..."). Subsequently, we use the same preprocessing pipeline for the cleaned ASR output as we do for all of our text data. For this, we start by removing non-printing characters, normalizing punctuation, and retokenizing the text using the corresponding utilities from the Moses toolkit (Koehn et al., 2007). Afterwards, we apply subword segmentation via SentencePiece (Kudo and Richardson, 2018), using a joint English–German BPE model with a vocabulary size of 32 000 for all of our translation models,



(a) opt1 (b) opt2 (c) opt2ShareEnc (d) opt3 (e) opt4 (f) opt5ShareEnc

Figure 1: Configurations tested for multitask training.

and an English unigram model with a vocabulary size of 24 000 for the restoration stage of our cascade SLT, both trained on all of the data used for the translation and restoration models combined.

Before the training of the restoration model, the training data was run through a Moses truecaser model (trained on the same selection of training data as the restoration model) as an additional step before segmentation. This step removes sentence-initial capitalization for words that would not be capitalized otherwise, ensuring that differences in distributions of words appearing in sentence-initial positions does not influence case restoration for the model. Once truecased and segmented, we assign the processed data as the target for the restoration model, and continue to strip punctuation and lowercase the target to generate the source. This configuration comes with the useful side effect of the model learning to generate truecased output, which may be beneficial for MT.

4 Experiments

In this section we report on the experiments that lead up to our final submissions. The experiments on this section have been trained, validated and tested on the respective splits of the MuST-C dataset.

As a first stage, we focused on selecting the multitask training strategy that performed better. Having the three modalities ENAUDIO, ENTEXT and DETEXT as possible inputs, and both text modalities as possible outputs, there can be up to 64 combinations where audio is an input[2] without taking into account the cases where the text encoder is shared between German and English. We considered the 5 scenarios depicted in Figure 1 and present its results in Table 3 together with the number of steps it took for them to converge.

All the models were trained using the same set of hyperparameters. At the time we ran these experiments, the final version of the audio encoder was not ready for deployment, so we used a 4-layered pyramidal CNN+RNN encoder adaptation from Amodei et al. (2016) with 512 hidden units and pooling factors of (1,1,2,2) after each layer, respectively. For the text encoders, we applied embedding layers of 512 dimensions, four stacked bidirectional LSTM layers with 512 hidden units (256 per direction). We use attentive text decoders composed of two unidirectional LSTM layers with 512 units. Regarding the shared attention bridge layer, we used 100 *attention heads* with 1024 hidden units each. Training is performed using the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.0002 and batch size 32 for all source-target pairs, for at most 100,000 steps per language pair[3]. At this stage, we apply a uniform language-rotating scheduler. Isolating the effect of multitasking from the effect of weighting the scheduling distribution helped us understand the importance of weighting it with respect to the batch size.

Configuration	BLEU	Steps
opt3	5.00	330K
opt5shareEnc	4.94	250K
opt2shareEnc	4.84	250K
opt4	4.50	300K
opt1	4.30	220K
opt2	3.62	190K

Table 3: Training steps and best BLEU scores obtained with end-to-end systems on the German part fo the MuST-C test set.

Our preliminary BLEU scores[4] for these models are low. We, however, justify our choice to include them given the low performance of other experiments in similar scenarios reported in the literature. Namely, Tu et al. (2019) reported 9.55 BLEU training on the same set with a transformer based architecture, the only paper that trains and tests on the same set, and thus the only truly com-

[2] 64 is the total number of bipartite graphs that can be defined on sets of three and two vertices.

[3] Model configuration 3, for instance, has 4 language pairs was trained for at most 400K steps

[4] We use the `multi-bleu-detok.pl+` Moses script, that uses sentence smoothing for detokenized input.

System	status	de BLEU	en BLEU	WER	Steps
end-to-end opt6	submission time	12.90	56.65	36	172K
	converged	14.38	59.22	33	294K
end-to-end opt3	submission time	9.47	44.12	48	32K
	converged	11.71	52.91	40	72K
cascade bpe37k		22.20	60.87	29	-
cascade char-level		20.90	54.49	55	-

Table 4: Scores of our primary and contrastive submissions on on the MuST-C test set.

parable results. In addition, Di Gangi et al. (2019a) reported 12.25 BLEU training MuST-C together with IWSLT18 and initialized their system with the ASR system.

The well-known sensitivity to hyperperparameter choice of the transformer architecture is also visible in our transformer-based audio encoders. We performed hyperparameter tuning on opt3 multitask training configuration (Figure 1 (d)). This resulted in a performance of a 9.53 BLEU score on German translations and 47.63 on the English, a clear increase from the untuned models that got at most 1 BLEU point in any of them. The final hyperparameter setup consists of:

- text encoders and decoders using 3 layered transformer architecture with 8 heads, 512 dimensional embeddings, 2048 feedforward hidden dimensions, and a batch size of 4096 tokens;

- audio encoders as described in Section 2 with 2 CNN layers with stride of 2 and kernel width of, the first of which takes a single input channel, three 8-headed transformer layers, positional embeddings of size 512 concatenated to the output of a linear layer for being passed to the transformer layers, a batch size of 32 utterances; and

- an attention bridge of size 100 with a hidden dimension of 1024.

Training was done with 8,000 warmup steps, using an Adam optimizer with learning rate 2 and Noam decay method, accumulation count of 8 to have an approximate effective batch size of 256 for the audio utterances, dropping utterances above the length of 5500, and a language rotating scheduler that uses the inverse of the batch size as weights [5].

We also tried other strategies such as (i) using 3, 4 and 6 stacked filterbanks as different channel inputs for the CNNs to reduce the input size instead of dropping utterances, (ii) using SpecAugment (Park et al., 2019) layers (2 frequency masks of width 20 and 2 time masks of width 50) to produce a data augmentation effect while training, (iii) including layer normalization after the attention bridge, (iv) using the positional embeddings of our transformer-based audio encoder in other places of the encoder or not using them at all. Unfortunately, none of them produced as effective improvements as what we describe above. We note that it is probable that using milder hyperparameters for SpecAugment could be beneficial.

5 Results

From the insights gained out of our experiments on the MuST-C dataset, for our submission, we train a system using the data as described in section 3 with the training configuration opt3 (see Figure 1 (d)) and the hyperparameters that yielded the best results. Further, we decided to try out an additional training configuration we had not previously tried out: ENAUDIO as input and DETEXT and ENTEXT as output, which we refer to as opt6. Configurations from Figure 1 use both modalities as input, whereas opt6 separates them by using only-audio input and only-text output. This might be the reason why opt6 outperformed them when tested on the MuST-C test set. Further experimentation would be required to make this statement conclusive. One of our main aims in participating in this task is to test our multitask architecture; for this reason we submit our best SLT system as primary system and the cascade approach with subword segmentation as contrastive baseline. We would like to

[5] In case of training opt3, the weights assigned to ENAUDIO → {DETEXT,ENTEXT} are 0.42 each and both text-to-text pairs get 0.08 because the average sentence length of MuST-C is around 24, which implies that 4096 tokens are about 170 sentences.

note that, unfortunately, at the time of submission, our end-to-end systems had not converged yet.

For the sake of consistency, these have been benchmarked with the MuST-C test set as well. The results are reported in Table 4, where we also report BLEU and WER for English, corresponding to the ASR task.

6 Conclusion

In this paper we present our work for the IWSLT2020 offline speech translation task, along with the set of experiments that led to our final systems. Our submission includes both a cascaded baseline and a multimodal system trainable in a multitask fashion. Our work shows that it is possible to train a system that shares internal representations for transferring the implicit semantics between audio and text modalities. The nature of the architecture enables end-to-end SLT, while at the same time providing a system capable of performing ASR and MT. Although this represents an important step in multimodal MT, there is still a lot of room for improvement in the proposed systems. In future work, we would like to implement more sophisticated audio encoders, such as the S-Transformer. This, along with using the same amount of data during training, will allow us to draw a truly fair comparison between both end-to-end and cascade approaches.

Acknowledgments

This work is part of the FoTran project, funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement № 771113), as well as the MeMAD project, funded by the European Union's Horizon 2020 Research and Innovation Programme (grant № 780069).

References

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, Jie Chen, Jingdong Chen, Zhijie Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Ke Ding, Niandong Du, Erich Elsen, Jesse Engel, Weiwei Fang, Linxi Fan, Christopher Fougner, Liang Gao, Caixia Gong, Awni Hannun, Tony Han, Lappi Johannes, Bing Jiang, Cai Ju, Billy Jun, Patrick LeGresley, Libby Lin, Junjie Liu, Yang Liu, Weigao Li, Xiangang Li, Dongpeng Ma, Sharan Narang, Andrew Ng, Sherjil Ozair, Yiping Peng, Ryan Prenger, Sheng Qian, Zongfeng Quan, Jonathan Raiman, Vinay Rao, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Kavya Srinet, Anuroop Sriram, Haiyuan Tang, Liliang Tang, Chong Wang, Jidong Wang, Kaifu Wang, Yi Wang, Zhijian Wang, Zhiqian Wang, Shuang Wu, Likai Wei, Bo Xiao, Wen Xie, Yan Xie, Dani Yogatama, Bin Yuan, Jun Zhan, and Zhenyao Zhu. 2016. Deep Speech 2: End-to-end speech recognition in English and Mandarin. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 173–182, New York, New York, USA. PMLR.

Ebrahim Ansari, Amittai Axelrod, Nguyen Bach, Ondrej Bojar, Roldano Cattoni, Fahim Dalvi, Nadir Durrani, Marcello Federico, Christian Federmann, Jiatao Gu, Fei Huang, Kevin Knight, Xutai Ma, Ajay Nagesh, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Xing Shi, Sebastian Stüker, Marco Turchi, and Changhan Wang. 2020. Findings of the IWSLT 2020 Evaluation Campaign. In *Proceedings of the 17th International Conference on Spoken Language Translation (IWSLT 2020)*, Seattle, USA.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber. 2019. Common Voice: A massively-multilingual speech corpus.

Mikko Aulamo, Sami Virpioja, and Jörg Tiedemann. 2020. OpusFilter: A configurable parallel corpus filtering toolbox. Accepted to ACL 2020, System Demonstrations.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*, San Diego, California, USA. Conference Track.

Ondřej Cífka and Ondřej Bojar. 2018. Are BLEU and meaning representation in opposition? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1362–1371, Melbourne, Australia. Association for Computational Linguistics.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019a. MuST-C: a multilingual speech translation corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, "Minneapolis, MN, USA".

Mattia A. Di Gangi, Matteo Negri, and Marco Turchi. 2019b. Adapting transformer to end-to-end spoken language translation. In *Proc. Interspeech 2019*, pages 1133–1137.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1723–1732, Beijing, China.

Orhan Firat, Kyunghyun Cho, Baskaran Sankaran, Fatos T Yarman Vural, and Yoshua Bengio. 2017. Multi-way, multilingual neural machine translation. *Computer Speech & Language*, 45:236–252.

Javier Iranzo-Sánchez, Joan Albert Silvestre-Cerdà, Javier Jorge, Nahuel Roselló, Adrià Giménez, Albert Sanchis, Jorge Civera, and Alfons Juan. Europarl-ST: A multilingual corpus for speech translation of parliamentary debates. Accepted to ICASSP2020.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *3rd International Conference for Learning Representations*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, Vancouver, Canada.

Philipp Koehn, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, Evan Herbst, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, and Christine Moran. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions - ACL '07*, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Yichao Lu, Phillip Keung, Faisal Ladhak, Vikas Bhardwaj, Shaonan Zhang, and Jason Sun. 2018. A neural interlingua for multilingual machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 84–92, Belgium, Brussels. Association for Computational Linguistics.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *4th International Conference on Learning Representations, ICLR 2016*, San Juan, Puerto Rico. Conference Track (Poster).

Graham Neubig, Matthias Sperber, Xinyi Wang, Matthieu Felix, Austin Matthews, Sarguna Padmanabhan, Ye Qi, Devendra Singh Sachan, Philip Arthur, Pierre Godard, John Hewitt, Rachid Riad, and Liming Wang. 2018. XNMT: The extensible neural machine translation toolkit. In *Conference of the Association for Machine Translation in the Americas (AMTA) Open Source Software Showcase*, Boston.

J. Niehues, R. Cattoni, S. Stüker, M. Negri, M. Turchi, E. Salesky, R. Sanabria, L. Barrault, L. Specia, and M. Federico. 2019. The IWSLT 2019 evaluation campaign. In *Proceedings of the 16th International Workshop on Spoken Language Translation (IWSLT 2019)*, Hong Kong, China.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin Dogus Cubuk, and Quoc V. Le. 2019. SpecAugment: A simple augmentation method for automatic speech recognition. In *INTERSPEECH*.

Ramon Sanabria, Ozan Caglayan, Shruti Palaskar, Desmond Elliott, Loïc Barrault, Lucia Specia, and Florian Metze. 2018. How2: A large-scale dataset for multimodal language understanding. In *Proceedings of the Workshop on Visually Grounded Interaction and Language (ViGIL)*. NeurIPS.

Holger Schwenk and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. In *ACL workshop on Representation Learning for NLP*, pages 157–167, Vancouver, Canada.

Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. 2018. Learning general purpose distributed sentence representations via large scale multi-task learning. In *6th International Conference on Learning Representations, ICLR 2018*, Vancouver, Canada. Conference Track (Poster).

Umut Sulubacak, Jörg Tiedemann, Aku Rouhe, Stig-Arne Grönroos, and Mikko Kurimo. 2018. The memad submission to the iwslt 2018 speech translation task. In *Proceedings of the 16th International Workshop on Spoken Language Translation (IWSLT 2019)*, pages 89–94, Brussels, Belgium.

Mei Tu, Wei Liu, Lijie Wang, Xiao Chen, and Xue Wen. 2019. End-to-end speech translation system description of LIT for IWSLT 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information*

Processing Systems, pages 5998–6008, Long Beach, California, USA.

Raúl Vázquez, Alessandro Raganato, Mathias Creutz, and Jörg Tiedemann. 2020. A systematic study of inner-attention-based sentence representations in multilingual neural machine translation. *Computational Linguistics*, 0(ja):1–53.

A.5 WMT paper about the Tatoeba MT Challenge

The Tatoeba Translation Challenge – Realistic Data Sets for Low Resource and Multilingual MT

Jörg Tiedemann
University of Helsinki
jorg.tiedemann@helsinki.fi
https://github.com/Helsinki-NLP/Tatoeba-Challenge

Abstract

This paper describes the development of a new benchmark for machine translation that provides training and test data for thousands of language pairs covering over 500 languages and tools for creating state-of-the-art translation models from that collection. The main goal is to trigger the development of open translation tools and models with a much broader coverage of the World's languages. Using the package it is possible to work on realistic low-resource scenarios avoiding artificially reduced setups that are common when demonstrating zero-shot or few-shot learning. For the first time, this package provides a comprehensive collection of diverse data sets in hundreds of languages with systematic language and script annotation and data splits to extend the narrow coverage of existing benchmarks. Together with the data release, we also provide a growing number of pre-trained baseline models for individual language pairs and selected language groups.

1 Introduction

The Tatoeba translation challenge includes shuffled training data taken from OPUS,[1] an open collection of parallel corpora (Tiedemann, 2012), and test data from Tatoeba,[2] a crowd-sourced collection of user-provided translations in a large number of languages. All data sets are labeled with ISO-639-3 language codes using macro-languages in case when available. Naturally, training data do not include sentences from Tatoeba and neither from the popular WMT testsets to allow a fair comparison to other models that have been evaluated using those data sets.

Here, we propose an open challenge and the idea is to encourage people to develop machine translation in real-world cases for many languages. The most important point is to get away from artificial setups that only simulate low-resource scenarios or zero-shot translations. A lot of research is tested with multi-parallel data sets and high resource languages using data sets such as WIT^3 (Cettolo et al., 2012) or Europarl (Koehn, 2005) simply reducing or taking away one language pair for arguing about the capabilities of learning translation with little or without explicit training data for the language pair in question (see, e.g., Firat et al. (2016a,b); Ha et al. (2016); Lakew et al. (2018)). Such a setup is, however, not realistic and most probably over-estimates the ability of transfer learning making claims that do not necessarily carry over towards real-world tasks.

In the set we provide here we, instead, include all available data from the collection without removing anything. In this way, the data refers to a diverse and skewed collection, which reflects the real situation we need to work with and many low-resource languages are only represented by noisy or very unrelated training data. Zero-shot scenarios are only tested if no data is available in any of the sub-corpora. More details about the data compilation and releases will be given below.

Tatoeba is, admittedly, a rather easy test set in general but it includes a wide variety of languages and makes it easy to get started with rather encouraging results even for lesser resourced languages. The release also includes medium and high resource settings and allows a wide range of experiments with all supported language pairs including studies of transfer learning and pivot-based methods.

2 Data releases

The current release includes over 500GB of compressed data for 2,961 language pairs covering 555 languages. The data sets are released per language

[1] http://opus.nlpl.eu/

[2] https://tatoeba.org/

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1174–1182
Online, November 19–20, 2020. ©2020 Association for Computational Linguistics

pair with the following structure, using deu-eng as an example (see Figure 1).

```
data/deu-eng/
data/deu-eng/train.src.gz
data/deu-eng/train.trg.gz
data/deu-eng/train.id.gz
data/deu-eng/dev.id
data/deu-eng/dev.src
data/deu-eng/dev.trg
data/deu-eng/test.src
data/deu-eng/test.trg
data/deu-eng/test.id
```

Figure 1: Released data packages: training data, development data and test data. Language labels are stored in ID files that also contain the name of the source corpus for the training data sets.

Files with the extension *.src* refer to sentences in the source language (*deu* in this case) and files with extension *.trg* contain sentences in the target language (*eng* here). File with extension *.id* include the ISO-639-3 language labels with possibly extensions about the orthographic script (more information below). In the *.id* file for the training data there are also labels for the OPUS corpus the sentences come from. We include the entire collection available from OPUS with data from the following corpora: ada83, Bianet, bible-uedin, Books, CAPES, DGT, DOGC, ECB, EhuHac, EiTB-ParCC, Elhuyar, EMEA, EUbookshop, EUconst, Europarl, Finlex, fiskmo, giga-fren, GlobalVoices, GNOME, hrenWaC, infopankki, JRC-Acquis, JW300, KDE4, KDEdoc, komi, MBS, memat, MontenegrinSubs, MultiParaCrawl, MultiUN, News-Commentary, OfisPublik, OpenOffice, OpenSubtitles, ParaCrawl, PHP, QED, RF, sardware, SciELO, SETIMES, SPC, Tanzil, TED2013, TedTalks, TEP, TildeMODEL, Ubuntu, UN, UNPC, wikimedia, Wikipedia, WikiSource, XhosaNavy.

The data sets are compiled from the pre-aligned bitexts but further cleaned in various ways. First of all, we remove non-printable characters and strings that violate Unicode encoding principles using regular expressions and a recoding trick using the forced encoding mode of *recode* (v3.7), a popular character conversion tool.[3] Furthermore, we also de-escape special characters (like ’&’ encoded as ’&amp;’) that may appear in some of the corpora. For that, we apply the tools from Moses (Koehn et al., 2007). Finally, we also apply automatic language identification to remove additional noise from the data. We use the compact language detect library (CLD2) through its Python bindings[4] and a Python library for converting between different ISO-639 standards.[5] CLD2 supports 172 languages and we use the options for ”best effort” and apply the assumed language from the original data as the ”hint language code”. For unsupported languages, we remove all examples that are detected to be English as this is a common problem in some corpora where English texts appear in various places (e.g. untranslated text in localization data of community efforts). In all cases, we only rely on the detected language if it is flagged as reliable by the software.

All corpus data and sub-languages are merged and shuffled using terashuf[6] that is capable to efficiently shuffle large data sets. But we keep track of the original data set and provide labels to recognize the origin. In this way, it is possible to restrict training to specific subsets of the data to improve domain match or to reduce noise. The entire procedure of compiling the Tatoeba Challenge data sets is available from the project repository at https://github.com/Helsinki-NLP/Tatoeba-Challenge.

The largest data set (English-French) contains over 180 million aligned sentence pairs and 173 language pairs are covered by over 10 million sentence pairs in our collection. Altogether, there are almost bilingual 3,000 data sets and we plan regular updates to improve the coverage. Below, we give some more details about the language labels, test sets and monolingual data sets that we include in the package as well.

2.1 Language labels and scripts

We label all data sets with standardized language codes using three-letter codes from ISO-639-3. The labels are converted from the original OPUS language IDs (which roughly follow ISO-639-1 codes but also include various non-standard IDs) and information about the writing system (or script) is automatically assigned using Unicode regular expressions and counting letters from specific script character properties. For the scripts we use four-letter codes from ISO-15924 and attach them to the three-letter language codes defined in ISO-639-3. Only the most frequently present script in a string is shown. Mixed content may appear but is not marked specifically. Note that the code Zyyy

[3] https://github.com/pinard/Recode

[4] https://pypi.org/project/pycld2/

[5] https://pypi.org/project/iso-639/

[6] https://github.com/alexandres/terashuf

refers to common characters that cannot be used to distinguish scripts. The information about the script is not added if there is only one script in that language and no other scripts are detected in any of the strings. If there is a default script among several alternatives then this particular script is not shown either. Note that the assignment is done fully automatically and no corrections have been made. Three example label sets are given below using the macro-languages Chinese (zho), Serbo-Croatian (hbs) and Japanese (jpn) that can use character from different scripts:

Chinese: cjy_Hans, cjy_Hant, cmn, cmn_Bopo, cmn_Hans, cmn_Hant, cmn_Latn, gan, lzh, lzh_Bopo, lzh_Hang, lzh_Hani, lzh_Hans, lzh_Hira, lzh_Kana, lzh_Yiii, nan_Hani, nan_Latn, wuu, wuu_Bopo, wuu_Hang, wuu_Hani, wuu_Hira, yue_Hans, yue_Hant, yue_Latn

Japanese: jpn, jpn_Hani, jpn_Hira, jpn_Kana, jpn_Latn

Serbo-Croatian: bos_Latn, hrv, srp_Cyrl, srp_Latn

This demonstrates that a data set may include examples from various sub-languages if they exist (e.g. Bosnian, Croatian and Serbian in the Serbo-Croatian case) or language IDs with script extensions that show the dominating script in the corresponding string (e.g. Cyrl for Cyrillic or Latn for Latin script). Those labels can be used to separate the data sets, to test sub-languages or specific scripts only or to remove some noise (like the examples that are tagged with the Latin script (Latn) in the Japanese data set. Note that script detection can also fail in which the corresponding code is missing or potentially wrong. For example, the detection of traditional (Hant) och simplified Chinese (Hans) can be ambiguous and encoding noise can have an effect on the detection.

We also release the tools that we developed for converting and standardizing OPUS IDs and also the tools that detect scripts and variants of writing systems. The package is available from github[7] and can be installed from CPAN.[8]

2.2 Multiple reference translations

Test and development data are taken from a shuffled version of Tatoeba. All translation alternatives are included in the data set to obtain the best coverage of languages in the collection. Development and test sets are disjoint in the sense that they do not include identical source-target language sentence pairs. However, there can be identical source sentences or identical target sentences in both sets, which are not linked to the same translations. Similarly, there can be identical source or target sentences in one of the sets, for example the test set, with different translations. In Figure 2, you can see examples from the Esperanto-Ladino test set.

epo	lad_Latn
u vi estas en Berlino?	Estash en Berlin?
u vi estas en Berlino?	Vos estash en Berlin?
u vi estas en Berlino?	Vozotras estash en Berlin?
La hundo estas nigra.	El perro es preto.
La hundo nigras.	El perro es preto.

Figure 2: Examples of test sentences with multiple reference translations taken from the Esperanto-Ladino test set.

The test data could have been organized as multi-reference data sets but this would require to provide different sets in both translation directions. Removing alternative translations is also not a good option as this would take away a lot of relevant data. Hence, we decided to provide the data sets as they are, which implicitly creates multi-reference test sets but with the wrong normalization.

2.3 Monolingual data

In addition to the parallel data sets we also provide monolingual data that can be used for unsupervised methods or data augmentation approaches such as back-translation. For that purpose, we extract public data from Wikimedia including source from Wikpedia, Wikibooks, Wikinews, Wikiquote and Wikisource. We extract sentences from data dumps provided in JSON format[9] and process them with jq,[10] a lightweight JSON processing tool. We apply the same cleaning steps as we do for the OPUS bitexts including language identification and convert language IDs to ISO-639-3 as before. Sentence boundaries are detected using UDPipe (Straka et al., 2016) with models trained on universal dependency treebanks v 2.4 and the Moses sentence splitter with language-specific non-breaking prefixes if available. We preserve document boundaries and do not shuffle the data to enable experiments with discourse-aware models. The data sets are released along with the rest of the Tatoeba challenge data.

3 The translation challenge

The main challenge is to develop translation models and to test them with the given test data from

[7] https://github.com/Helsinki-NLP/LanguageCodes

[8] https://metacpan.org/pod/ISO::639::3 and https://metacpan.org/pod/ISO::639::5

[9] https://dumps.wikimedia.org/other/cirrussearch/current

[10] https://stedolan.github.io/jq/

Tatoeba. The focus is on low-resource languages and to push their coverage and translation quality. Resources for high-resource are also provided and can be used as well for translation modeling of those languages and for knowledge transfer to less resourced languages. Note that not all language pairs have sufficient data sets for test, development (dev) and training ($train$) data. Hence, we divided the Tatoeba challenge data into various subsets based on the size of the training data available.

high-resource settings: 298 language pairs with training data of at least one million training examples (aligned sentence pairs), we further split into language pairs with more than 10 million training examples (173 language pairs) and other language pairs with data sets below the size of 10 million examples

medium-sized resource settings: 97 language pairs with more than 100,000 and less than 1 million training examples

low-resource settings: 87 language pairs with less than 100,000 training examples, we further distinguish between language pairs with more than 10,000 training examples (63) and language pairs below 10,000 training examples (24)

zero-shot translation: language pairs with no training data (40 in the current data set)

For all those 522 selected language pairs, the data set provides at least 200 sentences per test set. 101 of them involves English as one of the languages. 288 test sets contain more than 1,000 sentence pairs of which only 68 include English. Note, that everything below 1,000 sentences is probably not very reliable as a proper test set but we decided to release smaller test sets as an initial benchmark to trigger further development even for extremely under-resourced language pairs. We also decided to use very low thresholds for the division into low-resource languages. Having 10,000 training examples or less is very realistic for many real-world examples and we want to encourage the work on such cases in particular.

The maximum size of test sets in our collection is 10,000 sentence pairs, which is available for 76 language pairs. The test size is reduced to 5,000 if there is less than 20,000 sentence pairs in Tatoeba (19 data sets). The remaining sentences are released as disjoint validation data. For 48 Tatoeba language pairs with less than 10,000 sentence pairs, we keep 2,500 for the test set and the rest for validation and for 78 Tateoba language pairs with less than 5,000 sentence pairs we keep 1,000 for validation and the rest for testing. Finally, for language pairs with less than 2,000 sentences in Tatoeba we skip validation data and use everything for test purposes.

Test and validation data are strictly disjoint and none of the examples from Tatoeba are explicitly included in the training data. However, as it is common in realistic cases, there is a natural chance for a certain overlap between those data sets. Figure 3 plots the percentage of sentence pairs in test and validation sets that can also be found in the corresponding training data we release. The average proportion is rather low around 5.5% for both with a median percentage of 2.3% and 2.9% for test and validation data, respectively. There is one clear outlier with a very high proportion of over 55% overlap and that is Danish–English for some reason that is not entirely clear to us. Otherwise, the values are well below that ratio.

4 The data challenge

The most important ingredient for improved translation quality is data. It is not only about training data but very much also about appropriate test data that can help to push the development of transfer models and other ideas of handling low-resource settings. Therefore, another challenge we want to open here is the increase of the coverage of test sets for low-resource languages. Our strategy is to organize the extension of the benchmarks directly through the Tatoeba initiative. Users who would like to contribute to further MT benchmark development are asked to register for the open service provided by Tatoeba and to upload new translations in the languages of interest. From our side, we will continuously update our challenge data set to include the latest data releases coming from Tatoeba including new language pairs and extended data sets for existing language pairs. We will make sure that the new test sets do not overlap with any released development data from previous revisions to enable fair comparisons of old models with new benchmarks. The extended test and validation data sets will be released as new packages and old revisions will be kept for replicability of existing

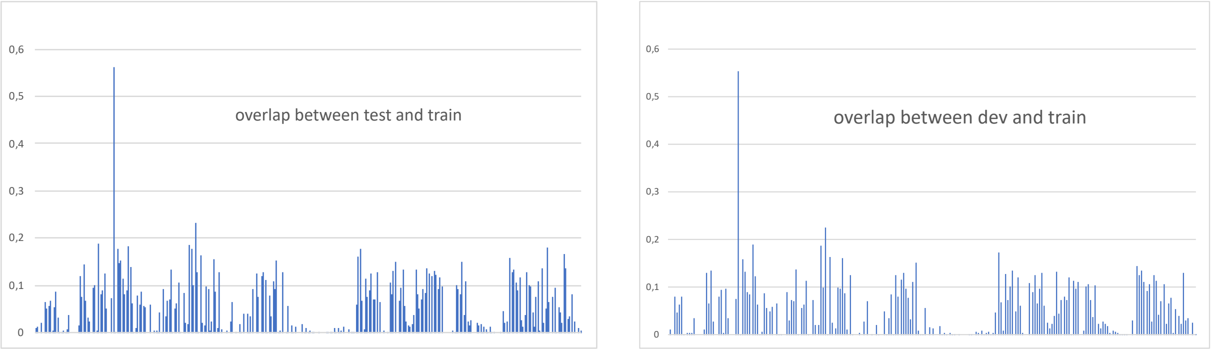


Figure 3: Overlap between test and validation (dev) data and the training data: Proportion of sentence pairs that exist in the training data for all data sets above 1,000 sentence pairs.

scores.

In order to provide information about language pairs in need, we provide a list of data sets with less than 1,000 examples per language pair. In the current release, this refers to 2,375 language pairs. 2,141 language pairs have less than 200 translation units and are, therefore, not included in the released benchmark test set. Furthermore, we also provide a list of languages for which we release training data coupled with English but no test data is available from Tatoeba. Currently, this relates to 246 languages.

We encourage users to especially contribute translations for those data sets in order to improve the language coverage even further. We hope to trigger a grass-root development that can significantly boost the availability of development and test sets as one of the crucial elements for pushing NMT development in the corresponding languages.

Finally, we also encourage to incorporate other test sets besides of the Tatoeba data. Currently, we also test with WMT news test sets for the language pairs that are covered by the released development and test sets over the years of the news translation campaign. Contributions and links can be provided through the repository management interface at github.

5 How to participate

The goal of the data release is to enable a straightforward setup for machine translation development. Everyone interested is free to use the data for their own development. A leader board for individual language pairs will be maintained. Furthermore, we also intend to make models available that are listed in the challenge. This does not only support replicability but also provides a new unique resource of pre-trained models that can be integrated in real-world applications or can be used in further research, unrelated downstream tasks or as a starting point for subsequent fine-tuning and domain adaptation. A large number of models is already available from our side providing baselines for a large portion of the data set. More details will be provided below.

For participation, there are certain rules that apply:

- Do not use any development or test data for training (*dev* can be used for validation during training as an early stopping criterion).
- Only use the provided training data for training models with comparable results in constrained settings. Any combination of language pairs is fine or backtranslation of sentences included in training data for any language pair is allowed, too. That means that additional data sets, parallel or monolingual, are not allowed for official models to be compared with others.
- Unconstrained models may also be trained and can be reported as a separate category. Using pre-trained language or translation models fall into the unconstrained category. Make sure that the pre-trained model does not include Tatoeba data that we reserve for testing.
- We encourage to release models openly to ensure replicability and re-use of pre-trained models. If you want to enter the official leader board you have to make your model available including instructions on how to use them.

6 Baseline Models

Along with the data, we also release baseline models that we train with state-of-the-art transformer models (Vaswani et al., 2017) using Marian-NMT,[11] a stable production-ready NMT toolbox with efficient training and decoding capabilities (Junczys-Dowmunt et al., 2018). We apply a common setup with 6 self-attentive layers in both, the encoder and decoder network using 8 attention heads in each layer. The hyper-parameters follow the general recommendations given in the documentation of the software.[12] The training procedures follow the strategy implemented in OPUS-MT (Tiedemann and Thottingal, 2020) and detailed instructions are available from github.[13]

We train a selection of models on v100 GPUs with early-stopping after 10 iterations of dropping validation perplexities. We use SentencePiece (Kudo and Richardson, 2018) for the segmentation into subword units and apply a shared vocabulary of a maximum of 65,000 items. Language label tokens in the spirit of Johnson et al. (2017) are used in case of multiple language variants or scripts in the target language. Models for over 400 language pairs are currently available and we refer the reader to the website with the latest results. For illustration, we provide some example scores below in Table 1 using automatic evaluation based on chrF2 and BLEU computed using sacrebleu (Post, 2018). The actual translations are also available for each model and the distribution comes along with the logfiles from the training process and all necessary data files such as the SentencePiece models and vocabularies.

language pair	chrF2	BLEU
aze-eng	0.490	31.9
bel-eng	0.268	10.0
cat-eng	0.668	50.2
eng-epo	0.577	35.6
eng-glg	0.593	37.8
eng-hye	0.404	16.6
eng-ilo	0.569	30.8
eng-run	0.436	10.4

Table 1: Translations scores from baseline models trained for a selection of medium-size language pairs (according to our classification) tested on the provided Tatoeba benchmark. We show here models that include English and score above 10 BLEU.

7 Multilingual Models

One of the most interesting questions is the ability of multilingual models to push the performance of low-resource machine translation. The Tatoeba translation challenge provides a perfect testbed for systematic studies on the effect of transfer learning across various subsets of language pairs. We already started various experiments with a number of multilingual translation models that we evaluate on the given benchmarks. In our current work, we focus on models that include languages in established groups and for that we facilitate the ISO-639-5 standard. This standard defines a hierarchy of language groups and we map our data sets accordingly to start new models that cover those sets. As an example, we look at the task of Belorussian-English translation that has been included in the previous section as well. Table 2 summarizes the results of our current models sorted by chrF2 scores.

model	chr-F2	BLEU
sla-eng/opus4m	0.610	42.7
sla-eng/opus2m	0.609	42.5
sla-eng/opus1m	0.599	41.7
ine-eng/opus2m	0.597	42.2
ine-eng/opus4m	0.597	41.7
ine-eng/opus1m	0.588	41.0
zle-eng/opus4m	0.573	38.7
zle-eng/opus2m	0.569	38.3
mul-eng/opus1m	0.550	37.0
mul-eng/opus2m	0.549	36.8
zle-eng/opus1m	0.543	35.4
ine-ine/opus1m	0.512	31.8
bel-eng/opus	0.268	10.0

Table 2: Translation results of the Belorussian-English test set using various multilingual translation models compared to the baseline bilingual model (shown at the bottom). opusXm refers to sampled data sets that include X million sentences per language pair.

The models focus on different levels of relatedness of the languages and range from East Slavic Languages (zle), Slavic languages (sla) to the language family of Indo-European languages (ine) and the set that contains all languages (mul). Each model is trained on sampled data set in order to balance between different languages. The smallest training sets are based on data that are sampled to include a maximum of one million sentence per language pair (opus1m). We use both, down-sampling and up-sampling. The latter is done by simply

[11] https://marian-nmt.github.io

[12] https://github.com/marian-nmt/marian-examples/tree/master/transformer

[13] https://github.com/Helsinki-NLP/OPUS-MT-train/blob/master/doc/TatoebaChallenge.md

multiplying the existing data until the threshold is reached. We also set a threshold of 50 for the maximum of repeating the same data in order to avoid over-representing small noisy data. The one-million models are trained first and form the basis of larger models. We continue training with data sets sampled to two million before increasing to four million sentence pairs.

The Table shows some interesting patterns. First of all, we can clearly see a big push in performance when adding related languages to the training data. This is certainly expected especially in the case of Belorussian that is closely related to higher-resource-languages such as Russian and Ukrainian. Interesting is that the East Slavic language group is not the best performing model even though it includes those two related languages. The additional information from other Slavic languages pushes the performance beyond their level quite significantly. Certainly, those models will see more data and this may cause the difference. The 'sla-eng' model covers 13 source languages whereas 'zle' only 5. Also interesting to see is that the Indo-European language model fairs quite well despite the enormous language coverage that this model has to cope with. On the other hand, the big 'mul' translation model does not manage to create the same performance and the limits of the standard model with such a massive setup become apparent. Training those models becomes also extremely expensive and slow and we did not manage to start the 4-million-sentence model.

Currently, we look into the various models we train and many other interesting patterns can be seen. We will leave a careful analyses to future work and also encourage the community to explore this field further using the given collection and benchmark. Updates about models and scores will be published on the website and we would also encourage more qualitative studies that we were not able to do yet.

8 Zero-shot and few-shot translation

Finally, we have a quick look at zero-shot and few-shot translation tasks. Table 3 shows results for Awadhi-English translation, one of the test sets for which no training data is available. Awadhi is an Eastern Hindi language in the Indo-Iranian branch of the Indo-European language family.[14]

model	chr-F2	BLEU
ine-eng/opus1m	0.285	10.0
mul-eng/opus1m	0.257	9.4
inc-eng/opus1m	0.217	6.8
iir-eng/opus1m	0.214	7.9
ine-ine/opus1m	0.201	2.4
tatoeba-zero/opus	0.042	0.1

Table 3: Translation results of the Awadhi-English test set using multilingual translation models.

The table shows that a naive approach of throwing all languages that are part of zero-shot language pairs into one global multilingual model (tatoeba-zero) does not work well. This is probably not very surprising. Another interesting observation is that a symmetric multilingual model with Indo-European languages on both sides (ine-ine) also underperforms compared to other multilingual models that only translate into English. Once again, the Indo-European-language-family to English model performs quite well. Note that the performance purely comes from overlaps with related languages as no Awadhi language data is available during training. The performance is still very poor and needs to be taken with a grain of salt. They demonstrate, however, the challenges one faces with realistic cases of zero-shot translation.

In Table 4, we illustrate another case that could be described as a realistic few-shot translation task. Our collection comes with 3,613 training examples for the translation between English and Faroese. The table shows our current results in this task using multilingual models that translate from English to language groups including the Scandinavian language in question.

model	chr-F2	BLEU
eng-gem/opus	0.318	9.4
gem-gem/opus	0.312	7.0
eng-gmq/opus	0.311	7.0
eng-ine/opus	0.281	6.3
eng-mul/opus	0.280	5.7
ine-ine/opus	0.276	5.9
tatoeba-zero/opus	0.042	0.1

Table 4: Translation results of the English-Faroese test set with different multilingual NMT models.

Again, we can see that the naive tatoeba-zero model is the worst. The symmetric Indo-European model performs better but the English-Germanic

[14] We use ISO639-3 and ISO639-5 standards for names and codes of languages and language groups.

model gives the best performance, which is still very low and not satisfactory for real-world applications. Once again, the example demonstrates the challenge that is posed by extremely low-resource scenarios and we hope that the data set we provide will trigger additional fascinating studies on a large variety of interesting cases.

9 Comparison to the WMT news task

Finally, we also include a quick comparison to the WMT news translation task, see Table 5. Note that we did not perform any optimization for that task, did not use any in-domain back-translations and did not run fine-tuning in the news domain. We only give results for English–German (in both directions) for the 2019 test data to give an impression about the released baseline models.

English – German		
model	BLEU	chr-F2
eng-deu	42.4	0.664
eng-gmw	35.9	0.616
eng-gem	35.0	0.613
eng-ine	26.6	0.554
eng-mul	21.0	0.512
WMT best	44.9	–

German – English		
model	BLEU	chr-F2
deu-eng	40.5	0.645
gmw-eng	36.6	0.615
gem-eng	37.2	0.618
ine-eng	31.7	0.571
mul-eng	27.0	0.529
WMT best	42.8	–

Table 5: Translation results of baseline models on English–German news translation from WMT 2019 using bilingual and multilingual Tatoeba baseline models. The BLEU scores are also compared to the best score that is currently available from http://matrix.statmt.org/matrix – retrieved on October 4, 2020.

The results demonstrate that the models can achieve high quality even on a domain they are not optimized for. The best scores in the German–English case are close to the top performing model registered for this task even though the comparison is not fair for various reasons. The purpose is anyway not to provide state-of-the-art models for the news translation task but baseline models for the Tatoeba case and in future work we will also explore the use of our models as the basis for systems that can be developed for other benchmarks and applications. In the example we can also see that multilingual models significantly lag behind bilingual ones in high-resource cases. Each increase of the language coverage (except for the move from West Germanic languages (gmw) to Germanic languages (gem) in the German–English case) leads to a drop in performance but note that those multilingual models are not fine-tuned for translating from and to German.

10 Conclusions

This paper presents a new comprehensive data set and benchmark for machine translation that covers roughly 3,000 language pairs and over 500 languages and language variants. We provide training and test data that can be used to explore realistic low-resource scenarios and zero-shot machine translation. The data set is carefully annotated with standardized language labels including variations in scripts and with information about the original source. We also release baseline models and results and encourage the community to contribute to the data set and machine translation development. All tools for data preparation and training bilingual as well as multilingual translation models are provided as open source packages on github. We are looking forward to new models, extended test sets and a better coverage of the World's languages.

Acknowledgements

This work is supported by the FoTran project (grant agreement No 771113), funded by the European Research Council (ERC) and the MeMAD project (grant agreement No 780069) under the European Union's Horizon 2020 research and innovation program. We would also like to acknowledge the support of the CSC IT Center for Science, Finland, for computational resources.

References

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016a. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Orhan Firat, Baskaran Sankaran, Yaser Al-onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. 2016b. Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 268–277, Austin, Texas. Association for Computational Linguistics.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association of Computational Linguistics*, 5(1):339–351.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 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, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Surafel M. Lakew, Marcello Federico, Matteo Negri, and Marco Turchi. 2018. Multilingual neural machine translation for low-resource languages. *IJCoL - Italian Journal of Computational Linguistics*, 4(1). Emerging Topics at the Fourth Italian Conference on Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: Trainable pipeline for processing CoNLL-u files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297, Portorož, Slovenia. European Language Resources Association (ELRA).

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of LREC*, Istanbul, Turkey.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.