MTCUE: Learning Zero-Shot Control of Extra-Textual Attributes by Leveraging Unstructured Context in Neural Machine Translation

Sebastian Vincent and Robert Flynn and Carolina Scarton
Department of Computer Science, University of Sheffield, UK
{stvincent1,rjflynn2,c.scarton}@sheffield.ac.uk

Abstract
Efficient utilisation of both intra- and extra-textual context remains one of the critical gaps between machine and human translation. Existing research has primarily focused on providing individual, well-defined types of context in translation, such as the surrounding text or discrete external variables like the speaker’s gender. This work introduces MTCUE, a novel neural machine translation (NMT) framework that interprets all context (including discrete variables) as text. MTCUE learns an abstract representation of context, enabling transferability across different data settings and leveraging similar attributes in low-resource scenarios. With a focus on a dialogue domain with access to document and metadata context, we extensively evaluate MTCUE in four language pairs in both translation directions. Our framework demonstrates significant improvements in translation quality over a parameter-matched non-contextual baseline, as measured by BLEU (+0.88) and COMET (+1.58). Moreover, MTCUE significantly outperforms a “tagging” baseline at translating English text. Analysis reveals that the context encoder of MTCUE learns a representation space that organises context based on specific attributes, such as formality, enabling effective zero-shot control. Pre-training on context embeddings also improves MTCUE’s few-shot performance compared to the “tagging” baseline. Finally, an ablation study conducted on model components and contextual variables further supports the robustness of MTCUE for context-based NMT.

1 Introduction
Research in neural machine translation (NMT) has advanced considerably in recent years, much owing to the release of the Transformer architecture (Vaswani et al., 2017), subword segmentation (Sennrich et al., 2016c) and back-translation (Sennrich et al., 2016b). This resulted in claims of human parity in machine translation (Hassan et al., 2018), which in turn prompted researchers to look beyond the sentence level: at how a translation still needs to be compatible with the context it arises in.

Figure 1: A high-level overview of MTCUE (EN→PL).

The task of contextual adaptation to more nuanced extra-textual variables like the description of the discourse situation has been largely overlooked, in spite of earlier work suggesting that conversational machine translation may benefit from such fine-grained adaptations (van der Wees et al., 2016). Most existing work on contextual NMT has focused on document-level context instead, aiming to improve the coherence and cohesion of the translated document (e.g. Tiedemann and Scherrer, 2017). Some research has successfully adapted NMT to extra-textual context variables using supervised learning frameworks on labelled datasets, targeting aspects such as gender (Vanmassenhove et al., 2018; Moryossef et al., 2019; Vincent et al., 2022b), formality (Sennrich et al., 2016a; Nadejde et al., 2022), translators’ or speakers’ style (Michel and Neubig, 2018a; Wang et al., 2021b) and translation length (Lakew et al., 2019), sometimes controlling multiple attributes simultaneously (Schioppa et al., 2021; Vincent et al., 2022b). However, to our knowledge, no prior work has attempted to model the impact of continuous extra-textual contexts in translation or combined the intra- and extra-textual contexts in a robust framework. This is
problematic since translating sentences without or with incomplete context is akin to a human translator working with incomplete information. Similarly, only a handful of earlier research has contemplated the idea of controlling these extra-textual attributes in a zero-shot or few-shot fashion (Moryossef et al., 2019; Anastasopoulos et al., 2022); such approaches are essential given the difficulty of obtaining the labels required for training fully supervised models.

In some domains, extra-textual context is paramount and NMT systems oblivious to this information are expected to under-perform. For instance, for the dubbing and subtitling domain, where translated shows can span different decades, genres, countries of origin, etc., a one-size-fits-all model is limited by treating all input sentences alike. In this domain, there is an abundance of various metadata (not just document-level data) that could be used to overcome this limitation. However, such adaptation is not trivial: (i) the metadata often comes in quantities too small for training and with missing labels; (ii) it is expressed in various formats and types, being difficult to use in a standard pipeline; (iii) it is difficult to quantify its exact (positive) effect.

In this paper, we address (i) and (ii) by proposing MTCUE (Machine Translation with Contextual universal embeddings), a novel NMT framework that bridges the gap between training on discrete control variables and intra-textual context as well as allows the user to utilise metadata of various lengths in training, easing the need for laborious data editing and manual annotation (Figure 1). During inference, when context is provided verbatim, MTCUE falls back to a code-controlled translation model; by vectorising the inputs, it exhibits competitive performance for noisy phrases and learns transferrability across contextual tasks. While (iii) is not directly addressed, our evaluation encompasses two translation quality metrics and two external test sets of attribute control, showing the impact on both translation quality and capturing relevant contextual attributes.

MTCUE can generalise to unseen context variables, achieving 100% accuracy at a zero-shot formality controlling task; it learns to map embeddings of input contexts to discrete phenomena (e.g. formality), increasing explainability; and it exhibits more robust few-shot performance at multi-attribute control tasks than a “tagging” baseline. The main contributions of this work are:

1. MTCUE (§2): a novel framework for combining (un)structured intra- and extra-textual context in NMT that significantly improves translation quality for four language pairs in both directions: English (EN) to/from German (DE), French (FR), Polish (PL) and Russian (RU).
2. A comprehensive evaluation, showing that MTCUE can be primed to exhibit excellent zero-shot and few-shot performance at downstream contextual translation tasks (§4 and §5).
3. Pre-trained models, code, and an organised version of the OpenSubtitles18 (Lison et al., 2018) dataset with the annotation of six metadata are made available.

This paper also presents the experimental settings (§3), related work (§6) and conclusions (§7).

2 Proposed Architecture: MTCUE

MTCUE is an encoder-decoder Transformer architecture with two encoders: one dedicated for contextual signals and one for inputting the source text. The signals from both encoders are combined using parallel cross-attention in the decoder. Below we describe how context inputs are treated in detail, and later in §2.2 and §2.3 we describe the context encoder and context incorporation, respectively.

2.1 Vectorising Contexts

Context comes in various formats: for example, the speaker’s gender or the genre of a film are often supplied in corpora as belonging to sets of pre-determined discrete classes, whereas plot descriptions are usually provided as plain text (and could not be treated as discrete without significant loss of information). To leverage discrete variables as well as short and long textual contexts in a unified framework, we define a vectorisation function that maps each context to a single meaningful vector, yielding a matrix $E_{c \times r}$, where $c$ is the number of contexts and $r$ is the embedding dimension. The function is deterministic (the same input is always embedded in the same way) and semantically coherent (semantically similar inputs receive similar embeddings). We use a sentence embedding model (Reimers and Gurevych, 2019) for vectorisation, which produces embeddings both deterministic and semantically coherent. Motivated by Khandelwal et al. (2018) and O’Connor and Andreas (2021) who report that generation models mostly use general topical information from past context, ignor-
ing manipulations such as shuffling or removing non-noun words, we hypothesise that sentence embeddings can effectively compress the relevant context information into a set of vectors, which, when processed together within a framework, will formulate an abstract representation of the dialogue context. We select the MiniLMv2 sentence embedding model (Wang et al., 2021a), which we access via the sentence-transformers library; a similar choice was made concurrently in Vincent et al. (2023). In the experiments, we also refer to DISTILBERT (Sanh et al., 2019) which is used by one of our baselines, and a discrete embedding function which maps unique contexts to the same embeddings but has no built-in similarity feature.

For any sample, given a set of its $k$ textual contexts $C = [c_1, ..., c_k]$, we vectorise each one separately using the method described above. The resulting array of vectors is the input we supply to the context encoder in MTCUE.

### 2.2 Context Encoder

**Processing vectorised contexts** The context encoder of MTCUE is a standard self-attention encoder with a custom input initialisation. Its inputs are sentence embeddings of context ($\S$2.1) projected to the model’s dimensions with a linear layer ($384 \rightarrow d_{\text{model}}$). In preliminary experiments, we observe that the first layer of the context encoder receives abnormally large input values, which sometimes leads to the explosion of the query ($Q$) and key ($K$) dot product $QK^T$. We prevent this by replacing the scaled dot product attention with query-key normalisation (Henry et al., 2020): applying L2 normalisation of $Q$ and $K$ before the dot product, and replacing the scaling parameter $\sqrt{d}$ with a learned one, initialised to a value based on training data lengths.\(^2\)

**Positional embeddings** We use positional context embeddings to (a) indicate the distance of a past utterance to the source sentence and (b) to distinguish metadata inputs from document information. In particular, when translating the source sentence $s_i$ at position $i$ in the document, a sentence distance positional embedding ($POS$) is added to the embedding representations of each past sentence $s_{i-j}$, with $j \in [0, t]$ where $t$ is the maximum allowed context distance: $e'(s_{i-j}, j) = e(s_{i-j}) + POS(j)$. Metadata contexts ($m_0, \ldots, m_n$) do not receive positional embeddings.

\(^2\)An alternative solution applies layer normalisation to the input of the first layer, but we found that this degraded performance w.r.t. QK-NORM.
beddings since their order is irrelevant. The final vectorised input of the context encoder is:
\( e'(s_i,0), e'(s_{i-1},1), \ldots, e'(s_{i-t}, t), e(m_0), \ldots, e(m_n) \).

2.3 Context incorporation

The outputs of the context and source encoders (respectively \( C \) and \( S \)) are combined in the decoder using parallel attention (Libovický et al., 2018). Let the output of the decoder self-attention be \( T \). Let \( T_{out} = FFN(T') + T' \), where \( T' \) is the multi-head attention output; i.e. \( T_{out} = T' \) with the feed-forward layer and the residual connection applied. In a non-contextual Transformer, source and target representations are combined with cross-attention:
\[
T' = \text{mAttn}(kv = S, q = T)
\]
In contrast, parallel attention computes individual cross-attention of \( T \) with \( S \) and \( C \) and then adds them together:
\[
S' = \text{mAttn}(kv = S, q = T)
C' = \text{mAttn}(kv = C, q = T)
T' = C' + S'
\]
Parallel attention is only one of many combination strategies which can be used, and in preliminary experiments we found the choice of the strategy to have a minor impact on performance.

3 Experimental Setup

3.1 Data: the OpenSubtitles18 Corpus

<table>
<thead>
<tr>
<th>Data type</th>
<th>EN→DE</th>
<th>EN→FR</th>
<th>EN→PL</th>
<th>EN→RU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source &amp; target</td>
<td>5.3M</td>
<td>14.7M</td>
<td>12.9M</td>
<td>12.4M</td>
</tr>
</tbody>
</table>

### metadata

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genre</td>
<td>45.3% 57.8% 60.5% 73.4%</td>
</tr>
<tr>
<td>PG rating</td>
<td>35.0% 46.9% 48.8% 62.3%</td>
</tr>
<tr>
<td>Writer(s)</td>
<td>45.3% 57.1% 58.0% 71.7%</td>
</tr>
<tr>
<td>Year</td>
<td>45.3% 57.8% 60.5% 73.7%</td>
</tr>
<tr>
<td>Country</td>
<td>37.7% 42.9% 45.7% 42.7%</td>
</tr>
<tr>
<td>Plot description</td>
<td>43.4% 57.1% 59.7% 72.6%</td>
</tr>
</tbody>
</table>

### previous dialogue

<table>
<thead>
<tr>
<th>n</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>n–1</td>
<td>60.1%</td>
<td>68.0%</td>
<td>63.7%</td>
<td>73.6%</td>
<td></td>
</tr>
<tr>
<td>n–2</td>
<td>42.0%</td>
<td>51.2%</td>
<td>46.4%</td>
<td>57.9%</td>
<td></td>
</tr>
<tr>
<td>n–3</td>
<td>31.2%</td>
<td>40.1%</td>
<td>35.5%</td>
<td>46.9%</td>
<td></td>
</tr>
<tr>
<td>n–4</td>
<td>23.9%</td>
<td>32.2%</td>
<td>28.0%</td>
<td>38.6%</td>
<td></td>
</tr>
<tr>
<td>n–5</td>
<td>18.7%</td>
<td>26.2%</td>
<td>22.4%</td>
<td>32.2%</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Data quantities for the extracted OpenSubtitles18 corpus. An average of 81% samples has at least one context input.

The publicly available OpenSubtitles18 corpus (Lison et al., 2018), hereinafter OpenSubtitles, is a subtitle dataset in .xml format with IMDb ID attribution and timestamps. It is a mix of original and user-submitted subtitles for movies and TV content. Focusing on four language pairs (EN→{DE,FR,PL,RU}), we extract parallel sentence-level data with source and target document-level features (up to 5 previous sentences) using the timestamps (see Appendix A). We also extract a range of metadata by matching the IMDb ID against the Open Movie Database (OMDb) API. Table 1 shows training data quantities and portions of annotated samples per context while Table 2 shows an example of the extracted data. We select six metadata types that we hypothesise to convey useful extra-textual information: plot description (which may contain useful topical information), genre (which can have an impact on the language used), year of release (to account for the temporal dimension of language), country of release (to account for regional differences in expression of English), writers (to consider writers’ style), PG rating (which may be associated with e.g. the use of adult language). For validation and testing, we randomly sample 10K sentence pairs each from the corpus, based on held-out IMDb IDs.

### Preprocessing

The corpus is first detokenised and has punctuation normalised (using Moses scripts (Koehn et al., 2007)). Then a custom cleaning script is applied, which removes trailing dashes, unmatched brackets and quotation marks, and fixes common OCR spelling errors. Finally, we perform subword tokenisation via the BPE algorithm with Sentencepiece (Kudo and Richardson, 2018).

Film metadata (which comes from OMDb) is left intact except when the fields contain non-values such as “N/A”, “Not rated”, or if a particular field is not sufficiently descriptive (e.g. a PG rating field represented as a single letter “R”), in which case
we enrich it with a disambiguating prefix (e.g. “R” → “PG rating: R”). Regardless of the trained language pair, metadata context is provided in English (which here is either the source or target language). Document-level context is limited to source-side context. Since for *→EN language pairs the context input comes in two languages (e.g. English metadata and French dialogue), we use multilingual models to embed the context in these pairs.

3.2 Evaluation

We evaluate the presented approach with the general in-domain test set as well as two external contextual tasks described in this section.

Translation quality The approaches are evaluated against an in-domain held-out test set of 10K sentence pairs taken from OPENSUBTITLES. As metrics, we use BLEU5 (Papineni et al., 2002) and COMET6 (Rei et al., 2020).

Control of multiple attributes about dialogue participants (EAMT22) The EAMT22 task, introduced by Vincent et al. (2022b), evaluates a model’s capability to control information about dialogue participants in English-to-Polish translation. The task requires generating hypotheses that align with four attributes: gender of the speaker and interlocutor(s) (masculine/feminine/mixed), number of interlocutors (one/many), and formality (formal/informal). These attributes can occur in a total of 38 unique combinations. We investigate whether MTCUE can learn this task through zero-shot learning (pre-training on other contexts) or through few-shot learning (when additionally fine-tuned on a constrained number of samples).

To prepare the dataset, we use scripts provided by Vincent et al. (2022b) to annotate OPENSUBTITLES with the relevant attributes, resulting in a corpus of 5.1M annotated samples. To leverage the context representation in MTCUE, we transcribe the discrete attributes to natural language by creating three sentences that represent the context. For example, if the annotation indicates that the speaker is male, the interlocutor is a mixed-gender group, and the register is formal, we create the following context: (1) “I am a man”, (2) “I’m talking to a group of people” and (3) “Formal”.

We train seven separate instances of MTCUE using different artificial data settings. Each setting contains the same number of samples (5.1M) but a varying number of annotated samples. To address class imbalances in the dataset (e.g. masculine speaker occurring more often than feminine speaker) and ensure equal representation of the 38 attribute combinations, we collect multiples of these combinations. We select sample numbers to achieve roughly equal logarithmic distances: 1, 5, 30, 300, 3K and 30K supervised samples per each of 38 combinations, yielding exactly 38, 180, 1, 127, 10, 261, 81, 953 and 510, 683 samples respectively. Including the zero-shot and full supervision (5.1M cases), this results in a total of eight settings. Each model is trained with the same hyperparameters as MTCUE, and on the same set of 5.1M samples, with only the relevant number of samples annotated (non-annotated samples are given as source-target pairs without contexts). We compare our results against our re-implementation of the TAGGING approach which achieved the best performance in the original paper (i.e. Vincent et al., 2022b). We train the TAGGING model in replicas of the eight settings above.

Zero-shot control of formality (IWSLT22) We experiment with the generalisation of MTCUE to an unseen type of context: formality. In the IWSLT22 formality control task (Anastasopoulos et al., 2022), the model’s challenge is to produce hypotheses agreeing with the desired formality (formal/informal). For the English-to-German language pair, the task provides a set of paired examples (each source sentence is paired with a formal reference and an informal one), to a total of 400 validation and 600 test examples; for the English-to-Russian pair, only the 600 test examples are provided. We test the capacity of MTCUE to control formality zero-shot, given a textual cue as context input.7

3.3 Baselines

In our experiments, we compare MTCUE with three types of baselines:

1. Base and Base-PM. These are pre-trained translation models that match MTCUE either in the shape of the encoder-decoder architecture (BASE) or in terms of the total number of parameters (BASE-PM). For BASE-PM, the extra parameters are obtained from enhancing the source encoder, increasing the number

5Computed with SacreBLEU (Post, 2018).
6Computed using the wmt20-comet-da model.
7We describe the process of choosing the context input for evaluation in Appendix D.
Table 3: Model details for MTCUE and baselines. Timings and epochs are averaged across all language directions.

3.4 Implementation and hyperparameters

We implement MTCUE and all its components in FAIRSEQ, and use HuggingFace (Wolf et al., 2020) for vectorising contexts. We use hyperparameters recommended by FAIRSEQ, plus optimise the learning rate and the batch size in a grid search. We found that a learning rate of 0.0003 and a batch size of simulated 200K tokens worked best globally. Table 3 presents the architecture details and runtimes for the models. All training is done on a single A100 80GB GPU, one run per model. We use early stopping based on validation loss with a patience of 5.

4 Results

Translation quality Results in Table 4 show that MTCUE beats all non-contextual baselines in translation quality, achieving an average improvement of \(+1.51\) BLEU/+3.04 COMET over BASE and \(+0.88/+1.58\) over BASE-PM. It is also significantly better than NOVOTNEY-CUE (+0.46/+0.66). MTCUE achieves comparable results to the parameter-matched TAGGING model, consistently outperforming it on all language directions from English, and being outperformed by it on directions into English. Since the primary difference between the two models is that MTCUE sacrifices more parameters to process context, and TAGGING uses these parameters for additional processing of source text, we hypothesise that the difference in scores is due to the extent to which context is a valuable signal for the given language pair: it is less important in translation into English. This is supported by findings from literature: English is a language that does not grammatically mark phenomena such as gender (Stahlberg et al., 2007).

The largest quality improvements with MTCUE are obtained on EN-DE (+1.66/+4.14 vs BASE-PM and +1.14/+1.70 vs TAGGING) and EN-FR (+2.23/+3.32 vs BASE-PM and +0.80/+0.62 vs TAGGING) language pairs. Contrastively, the smallest improvements against BASE-PM are obtained on the RU-EN pair. MTCUE is outperformed by
Table 4: Translation quality results on the OPENSUBTITLES test set. *Model trained without access to any context. We highlight the best result in each column and underline all statistically indistinguishable results, \( p \leq 0.05 \) (except the Average column).

Table 5: Evaluation on the IWSLT22 formality control evaluation campaign. Baseline systems were trained on different corpora.

**Control of multiple attributes about dialogue participants (EAMT22)**: MTCUE achieves 80.25 zero-shot accuracy at correctly translating the speaker and interlocutor attributes, an improvement of 12.08 over the non-contextual baseline, also expressed in increased translation quality (25.22 vs 23.36 BLEU). Furthermore, it boasts TAGGING at few-shot performance by 5 to 8 accuracy points, reaching above 90% accuracy with only 190 of the 5.1M annotated samples (Figure 4). Both TAGGING and MTCUE perform similarly with more supervised data. The TAGGING model achieves +2 to +3 accuracy points in the 1K to 100K range, while BLEU remains comparable. We hypothesise that this happens because MTCUE relies strongly on its pre-training prior when context is scarce: this proves useful with little data, but becomes less relevant as more explicitly labelled samples are added. Finally, with full supervision, both models achieve above 99% accuracy.

**Zero-shot control of formality (IWSLT22)**: MTCUE appears to successfully control the formality of translations in a zero-shot fashion, achieving nearly 100% accuracy on the IWSLT22 test sets across two language pairs, beating all zero-shot models on the EN-RU pair and performing on par with the best supervised model for EN-DE. Notably, both baselines presented in Table 5 were built to target formality specifically, unlike MTCUE which is a general-purpose model.

Following MTCUE’s success at controlling formality with sample contexts, we investigate the relationship between context embeddings and their corresponding formality control scores. We consider all 394 unique contexts from the OPENSUBTITLES validation data, and another 394 document contexts (individual past sentences) at random (in-domain). We also use an in-house dataset from a similar domain (dubbing of reality cooking shows with custom annotations of scene contents) and select another 394 metadata and 394 document contexts from there (out-of-domain). We run inference on the IWSLT22 test set with each context individually (1,576 runs), and use UMAP (McInnes et al., 2018) to visualise (i) the input embedding from MINILM-v2, (ii) the output vector of the context encoder and (iii) the corresponding formality score (Figure 3).

We invite the reader to pay attention to the separation of dark and light points in Figure 3b that is not present in Figure 3a. There is a spatial property that arises in the context encoder and is shown by Figure 3b, namely a relationship between the feature vectors from context encoder and formality scores across both domains: contexts yielding translations of the same register tend to be clustered together. This is true for both in-domain data (cir-
The robustness of MTCUE with an ablation study on the model components as well as a complementary ablation on types of context (metadata vs document). We evaluate three language pairs (EN→DE, FR, PL) and report results from single runs (Table 6): COMET score on the OpenSub...

Table 6: Ablation study on model components and data settings. *Corresponds to non-contextual Transformer.
titles18 data and zero-shot accuracy at the two contextual tasks (on the validation sets in all cases).

Removing the context encoder (output of the linear layer is combined with source straight away) or the position embeddings has only a minor effect on the COMET score; replacing MiniLM-v2 with a discrete embedding function hurts performance the most. Positional embeddings seem more important to the EAMT22 task than IWSLT22 - possibly because EAMT22 focuses on sentence-level phenomena, so the order of past context matters.

Replacing MiniLM-v2 with a discrete embedding function removes the zero-shot effect in both tasks. An interesting finding is that between metadata and document-level data, it is the latter that brings more improvements to contextual tasks; this means that our model potentially scales to domains without metadata. Finally, using random context degrades performance w.r.t. full model implying that the gains come from signals in data rather than an increase in parameters or training time.

6 Related Work

Although contextual adaptation has been discussed in other tasks (e.g. Keskar et al., 2019), in this section we focus on NMT, as well as set our work side by side with research that inspired our approach.

Existing studies on incorporating context into NMT have primarily focused on document-level context. These approaches include multi-encoder models (e.g. Miculicich et al., 2018), cache models (Kuang et al., 2018), automatic post-editing (Voita et al., 2019a), shallow fusion with a document-level language model (Sugiyama and Yoshinaga, 2021), data engineering techniques (Lupo et al., 2022) or simple concatenation models (Tiedemann and Scherrer, 2017). Another line of research aims to restrict hypotheses based on certain pre-determined conditions, and this includes formality (Sennrich et al., 2016a), interlocutors’ genders (e.g. Vanmassenhove et al., 2018; Moryossef et al., 2019), or a combination of both (Vincent et al., 2022b).

Other conditions include translation length and monotonicity (Lakew et al., 2019; Schioppa et al., 2021), vocabulary usage (Post and Vilar, 2018) or domain and genre (Matusov et al., 2020). While wider contextual adaptation in NMT has been discussed theoretically, most empirical research falls back to gender (Rabinovich et al., 2017) or formality control (Niu et al., 2017). One exception is Michel and Neubig (2018b) who adapt NMT for each of many speakers by adding a “speaker bias” vector to the decoder outputs.

Our work is motivated by the CUE vectors (Novotney et al., 2022) and their application in personalised language models for film and TV dialogue Vincent et al. (2023). CUE vectors represent context computed by passing sentence embeddings of the input context through a dedicated encoder. Novotney et al. show that incorporating CUE in language modelling improves perplexity, while Vincent et al. use them to personalise language models for on-screen characters. In contrast, we reformulate CUE for contextual machine translation, provide a detailed analysis of incorporating CUE into the model, emphasise the importance of vectorising the context prior to embedding it, and examine the benefits for zero-shot and few-shot performance in contextual NMT tasks.

7 Conclusions

We have presented MTCUE, a new NMT architecture that enables zero- and few-shot control of contextual variables, leading to superior translation quality compared to strong baselines across multiple language pairs (English to others, cf. Table 4). We demonstrated that using sentence embedding-based vectorisation functions over discrete embeddings and leveraging a context encoder significantly enhances zero- and few-shot performance on contextual translation tasks. MTCUE outperforms the winning submission to the IWSLT22 formality control task for two language pairs, with zero-shot accuracies of 100.0 and 99.7 accuracy respectively, without relying on any data or modelling procedures for formality specifically. It also improves by 12.08 accuracy points over the non-contextual baseline in zero-shot control of interlocutor attributes in translation at the EAMT22 English-to-Polish task. Our ablation study and experiments on formality in English-to-German demonstrated that the context encoder is an integral part of our solution. The context embeddings produced by the context encoder of the trained MTCUE can be mapped to specific effects in translation outputs, partially explaining the model’s improved translation quality. Our approach emphasises the potential of learning from diverse contexts to achieve desired effects in translation, as evidenced by successful improvements in formality and gender tasks using film metadata and document-level information in the dialogue domain.
Limitations

While we carried out our research in four language pairs (in both directions), we recognise that these are mainly European languages and each pair is from or into English. The choice of language pairs was limited by the data and evaluation tools we had access to, however as our methods are language-independent, this research could be expanded to other pairs in the future.

Another limitation is that the work was conducted in one domain (TV subtitles) and it remains for future work to investigate whether similar benefits can be achieved in other domains, though the findings within language modelling with CUE in Novotney et al. (2022) who used a different domain suggest so.

Ethics Statement

We do not foresee a direct use of our work in an unethical setting. However, as with all research using or relying on LMs, our work is also prone to the same unwanted biases that these models already contain (e.g. social biases). Therefore, when controlling contextual attributes, researchers should be aware of the biases in their data in order to understand the models’ behaviour.

Acknowledgements

This work was supported by the Centre for Doctoral Training in Speech and Language Technologies (SLT) and their Applications funded by UK Research and Innovation [grant number EP/S023062/1]. We acknowledge IT Services at The University of Sheffield for the provision of the High Performance Computing Service. This work was also supported by ZOO Digital.

References


A Data Preprocessing

Parsing OPENSUBTITLES To prepare OPENSUBTITLES (specifically the document-level part of the corpus), we follow the setup described in Voita et al. (2019b). There are timestamps and overlap values for each source-target sample in the corpus; we only take into account pairs with overlap $\geq 0.9$ and we use two criteria to build any continuous document: (1) no omitted pairs (due to poor overlap) and (2) no distance greater than seven seconds between any two consecutive pairs. To generate train/validation/test splits, we use generated lists of held-out IMDB IDs based on various published test splits (Müller et al., 2018; Lopes et al., 2020; Vincent et al., 2022b) to promote reproducibility. These lists can be found within the GitHub repository associated with this paper.

Embedding contexts Since a lot of metadata is repeated, and models are trained for multiple epochs, we opt for the most efficient way of embedding and storing data which is to use a memory-mapped binary file with embeddings for unique contexts, and an index which maps each sample to its embedding. This saves more than 90% space w.r.t. storing a matrix of all embeddings, and trains over $3\times$ faster than embedding batches on-the-fly.

B Model details

MTCUE is trained from a pre-trained machine translation model (corresponding to the BASE model) which is the transformer NMT architecture within FAIRSEQ. We follow model specifications and training recommendations set out by FAIRSEQ in their examples for training a translation model\(^8\). We train a model for each of the eight language directions on the source-target pairs from OPENSUBTITLES. We train the model until a patience parameter of 5 is exhausted on the validation loss.

C Observations on training and hyperparameters

We shortly describe here our findings from seeking the optimal architecture for MTCUE and training settings in the hope that this helps save the time of researchers expanding on our work.

- Reducing the number of context encoder layers led to inferior performance.
- Freezing the source encoder when fine-tuning MTCUE from a translation model led to inferior performance,
- Training MTCUE from scratch — significantly increased training time while having a minor effect on performance.

• Other context combination strategies (sequential and flat attention in Libovický et al., 2018) led to similar results.
• Some alternatives to QK-NORM to combat the problem of the exploding dot-product were successful but had a negative impact on performance:
  – using layer normalisation after the linear layer is applied to vectorised contexts,
  – using SmallEmb\(^9\) which initialises the embedding layer (in our case, the linear proj. layer) to tiny numbers and adds layer normalisation on top.
• Zero-shot performance at the IWSLT22 task is generally consistent (at around 98.0—100.0 accuracy) though may vary depending on the selected checkpoint. We found that training MTC\(^{UE}\) for longer (i.e. more than 20 epochs) may improve translation quality but degrade the performance on e.g. this task.
• We found that MTC\(^{UE}\) is generally robust to some hyperparameter manipulation on the OPEN-SUBTITLES dataset, and recommend performing a hyperparameter search when training the model on new data. For simplicity, in this paper we use a single set of hyperparameters for all language directions, though for some pairs the results may improve by manipulating parameters such as batch size and context dropout.

D Formality
To evaluate the performance of any tested model on the formality task we had to come up with a fair method of choosing a context to condition on, since in a zero-shot setting the model organically learns the tested attributes from various contexts rather than specific cherry-picked sentences.

To do so, we sampled some metadata from the validation set of the OPEN-SUBTITLES data and picked eight contexts (four for the formal case and four for the informal case) which either used formal or informal language themselves or represented a domain where such language would be used. We also added two generic prompts: Formal conversation and Informal chit-chat. The full list of prompts was as follows:

1. Formal conversation
2. Hannah Larsen, meet Sonia Jimenez. One of my favourite nurses.
3. In case anything goes down we need all the manpower alert, not comfortably numb.
4. Biography, Drama,
5. A musician travels a great distance to return an instrument to his elderly teacher

• Informal:
  1. Informal chit-chat
  2. I’m gay for Jamie.
  3. What else can a pathetic loser do?
  4. Drama, Family, Romance
  5. Animation, Adventure, Comedy

We then ran the evaluation as normal with each context separately, and selected the highest returned score for each attribute.

E Examples of Model Outputs (Zero-Shot)
We include examples of translations produced zero-shot by MTC\(^{UE}\) in Table 7. We would like to draw attention particularly to the top example for the EAMT22 task (“I just didn’t want you to think you had to marry me”). The phrase to marry someone can be translated to Polish in several ways, indicating that the addressee is to be a wife (ożenić się z kimś), a husband (wyjść za kogoś [za męż]) or neutral (wziąć ślub). While the reference in this example uses a neutral version, both the baseline model and MTC\(^{UE}\) opted for feminine/masculine variants. However, the gender of the speaker is feminine, so the phrase “… had to marry me” should use either the neutral version (wziąć ślub) or the feminine one (ożenić się). The baseline model incorrectly picks the masculine version while MTC\(^{UE}\) is able to pick the correct one based on the context given. MTC\(^{UE}\) also correctly translates the gender of the interlocutor: both in the top example (myślał vs myślała) and the bottom one (aš vs eš, even though a synonymous expression is used in translation, agreement remains correct). Finally, the IWSLT22 example shows how MTC\(^{UE}\) produces correct possessive adjectives for each formality.

EAMT22

<table>
<thead>
<tr>
<th>Source</th>
<th>I just didn’t want you to think you had to marry me.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context</td>
<td>I am a woman. I am talking to a man</td>
</tr>
<tr>
<td>Reference</td>
<td>Bo nie chciałam, żebyś myślał, że cię zmuszam do ślubu.</td>
</tr>
<tr>
<td></td>
<td>“Because I didn’t want you to think I am forcing you into a wedding.”</td>
</tr>
<tr>
<td>Baseline</td>
<td>Po prostu nie chciałem, żebyś myślała, że musisz za mnie wyjść.</td>
</tr>
<tr>
<td></td>
<td>“I just didn’t want you to think you had to marry me.”</td>
</tr>
<tr>
<td>MTCUE</td>
<td>Nie chciałam, żebyś myślał, że musisz się ze mną ożenić.</td>
</tr>
<tr>
<td></td>
<td>“I didn’t want you to think you had to marry me.”</td>
</tr>
</tbody>
</table>

Source | So then you confronted Derek. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Context</td>
<td>I am talking to a woman</td>
</tr>
<tr>
<td>Reference</td>
<td>A więc doprowadziłaś do konfrontacji z Derekiem.</td>
</tr>
<tr>
<td></td>
<td>“So then you led to a confrontation with Derek.”</td>
</tr>
<tr>
<td>Baseline</td>
<td>Więc wtedy skonfrontowałaś się z Derekiem.</td>
</tr>
<tr>
<td></td>
<td>“So then you confronted Derek.”</td>
</tr>
<tr>
<td>MTCUE</td>
<td>Więc skonfrontowała się z Derekiem.</td>
</tr>
<tr>
<td></td>
<td>“So then you confronted Derek.”</td>
</tr>
</tbody>
</table>

IWSLT22

<table>
<thead>
<tr>
<th>Source</th>
<th>I got a hundred colours in your city.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTCUE (formal)</td>
<td>Ich habe 100 Farben in Ihrer Stadt.</td>
</tr>
<tr>
<td>MTCUE (informal)</td>
<td>Ich hab 100 Farben in deiner Stadt.</td>
</tr>
</tbody>
</table>

Table 7: Examples of MTCUE’s outputs (zero-shot) versus a non-contextual Transformer baseline.
ACL 2023 Responsible NLP Checklist

A  For every submission:

☑️ A1. Did you describe the limitations of your work?
   Unnumbered section after section 7 (Conclusions)

☒ A2. Did you discuss any potential risks of your work?
   There are no relevant risks associated with our work

☑️ A3. Do the abstract and introduction summarize the paper’s main claims?
   Abstract + section 1 (Introduction)

☒ A4. Have you used AI writing assistants when working on this paper?
   Left blank.

B  ✔ Did you use or create scientific artifacts?

   Section 2 (proposed architecture); section 3.1 (used the OpenSubtitles corpus); section 3.2 (used two evaluation suites); section 2.2 (used sentence embedding models); section 3.4 (used software for implementation)

☑️ B1. Did you cite the creators of artifacts you used?
   Yes (sections 3.1, 3.2, 2.2, 3.4)

☑️ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
   For created artifacts: section 1 For used artifacts: section 3.1

☑️ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
   section 3.1. For sentence-transformers we did not explicitly discuss this but made all the necessary steps requested by the authors, such as citing the library and relevant papers.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
   Not applicable. Left blank.

☑️ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
   Section 1

☑️ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
   Table 1, Table 2, sections 3.1, 3.2

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.
C  ✔ Did you run computational experiments?

*Section 3 (experimental setup), section 4 (results), section 5 (ablation study)*

✔ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?

*Section 3.4, Table 4*

✔ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?

*Section 3, section 3.4*

✔ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?

*Section 4*

✔ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?

*Sections 2.2, 3.1, 3.2*

D  ✗ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?

*No response.*

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants’ demographic (e.g., country of residence)?

*No response.*

☐ D3. Did you discuss whether and how consent was obtained from people whose data you’re using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?

*No response.*

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?

*No response.*

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?

*No response.*