Context Aware NMT (Target side context, Monolingual repair and Decoding)

Ife Adebara

Outline

- Introduction
- Context aware NMT
- Example

Context Aware NMT

- Document-level NMT
  - Learn contextual information from surrounding sentences
    - Information for translation may not be within the current sentence
    - Sentence-level NMT encodes individual sentences

Why we need context aware NMT

- Mo ra **ìwé** fún Kólá. (Indefinite non-specific)
  1SG buy book for Kólá.
  ‘I bought **a book** for Kólá.’
- L’ójó kejì, Kólá ti so **ìwé** nù (Definite)
  In day second Kólá PERF throw book away
  ‘By the second day, Kólá had lost **the book**.’

Improving Context-aware Neural Machine Translation with Target-side Context

- Most context-aware systems use source-side contexts
- Claims:
  - The target-side context is as important as the source-side context.
  - The effectiveness of source-side context depends on language pairs.
  - Weight sharing between current and context states is effective for context-aware NMT.

Basics

$$p(Y^i|X^i,Z^{i-1}) = \prod_{n=1}^{N^i} p(y_n^i|y_{<n}^i,X^i,Z^{i-1}) \quad (1)$$

$$p(y_n^i|y_{<n}^i,X^i,Z^{i-1}) = \text{softmax}(W_o\tilde{h}_n^i) \quad (2)$$

$$\tilde{h}_n^i = W_h[h_n^i;c_n^i;c_n^{i-1}] \quad (3)$$

$$c_n^i = \sum_{m=1}^{M^i} \alpha_{n,m}^i s_m^i \quad (4)$$

$$\alpha_{n,m}^i = \text{softmax}(s_m^i \cdot h_n^i) \quad (5)$$

$$s_m^i = \text{LSTM}_{enc}(W_x x_m^i, s_{m-1}^i) \quad (6)$$

$$h_n^i = \text{LSTM}_{dec}(W_y y_n^i, h_{n-1}^i) \quad (7)$$

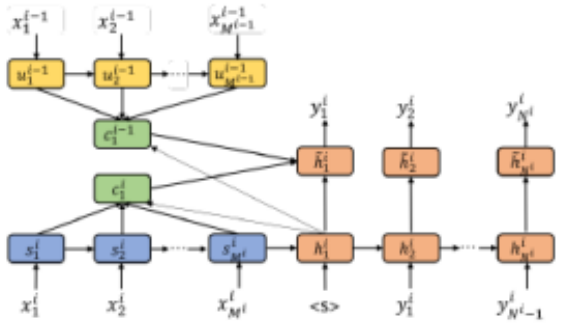
$$c_n^{i-1} = \sum_{t=1}^{|Z^{i-1}|} \beta_{n,t}^{i-1} z_t^{i-1} \quad (8)$$

$$\beta_{n,t}^{i-1} = \text{softmax}(z_t^{i-1} \cdot h_n^i) \quad (9)$$

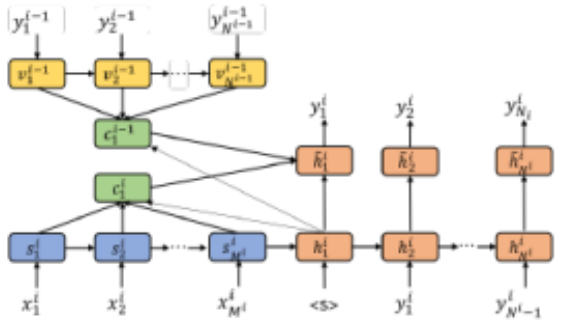
- Y^i = Source sentence
- y^i = Source token
- X^i = Target Sentence
- x^i = Target token
- Z^{i-1} = Previous sentence (of source or target)
- s_m^i = encoder states
- h_n^i = decoder states
- c_n^i = attention
- c_n^{i-1} = attention using previous sentence
- $W_o \in \mathbb{R}^{V \times H}$ $W_o \in \mathbb{R}^{H \times 3H}$ = weights
- $W_x \in \mathbb{R}^{E \times V}$ $W_y \in \mathbb{R}^{E \times V}$ = embeddings

Models - Separated Model

- It has an additional encoder, referred to as a context encoder.
- Each context encoder u_t^{i-1}or v_t^{i-1} reads a previous source-side or target-side sentence as context, respectively
- The weights of a context encoder are different from those of a current encoder which encodes a current source sentence.



(a) Separated source model.



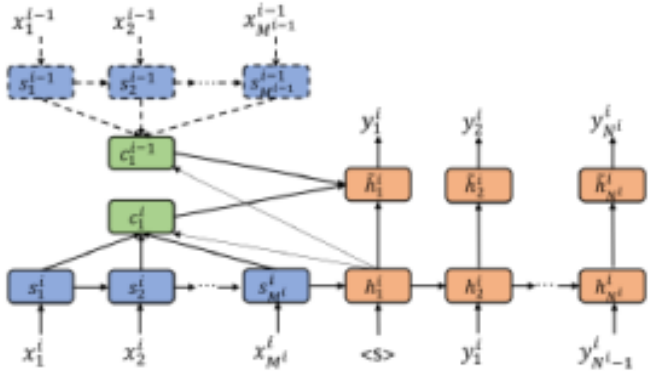
(b) Separated target model.

$$\boldsymbol{u}_t^{i-1} = \mathrm{LSTM}_{\mathrm{src\_enc}}(W_{\mathrm{x}}\boldsymbol{x}_t^{i-1}, \boldsymbol{u}_{t-1}^{i-1}) \tag{10}$$

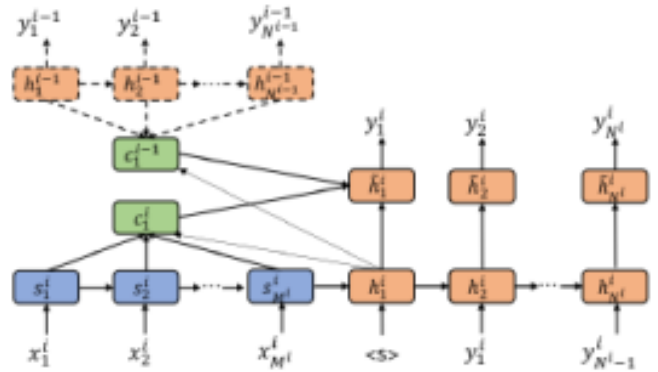
$$\boldsymbol{v}_t^{i-1} = \mathrm{LSTM}_{\mathrm{trg\_enc}}(W_{\mathrm{y}}\boldsymbol{y}_t^{i-1}, \boldsymbol{v}_{t-1}^{i-1}) \tag{11}$$

Models - Shared Models

- It uses the hidden states of an encoder or decoder to calculate c_n^{i-1} when translating a current sentence.
- The target-side context can be incorporated into a decoder instead of an encoder.
- The shared model does not require much additional parameters and extra computational times because this model simply loads the saved hidden states.
- These models are examples of weight sharing between a current encoder or decoder and a context encoder.
- The shared source model uses s_t^{i-1} as z_t^{i-1} and the shared target model uses h_t^{i-1} as z_t^{i-1}



(c) Shared source model.



(d) Shared target model.

Models – Shared Mixed Model

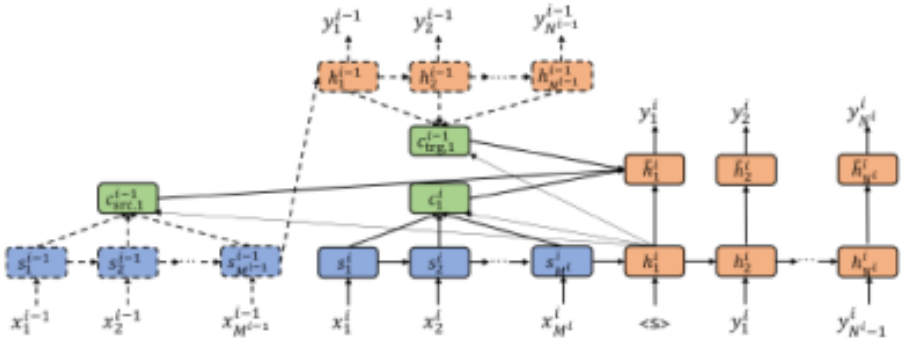


Fig. 2: Shared mix model.

- Incorporates the source and target side contexts
- Attention vector $C^{i-1} = c_{source}^{i-1} + c_{target}^{i-1}$

Experiments

- Data - IWSLT2017 German–English, Chinese–English, and Japanese–English datasets from TED and Recipe Corpus.
  - Each talk of TED is considered as a document.
  - The documents that include sentences of more than 100 words are eliminated from the training corpus.
  - Evaluate methods on the 2014 test set
- Settings – 2-layer bi-LSTM encoder, 2-layer uni-LSTM
  - Dimensions of hidden states and embeddings = 512
  - Dropout = 0.2
  - AdaGrad optimization = 0.01
  - Batch = 128 documents
  - Dot global attention
  - $C^0 = 0$ (because no previous contexts)
  - 30 epochs

Corpus	Train	Dev	Test
TED De–En	203,998	888	1,305
TED Zh–En	226,196	879	1,297
TED Ja–En	194,170	871	1,285
Recipe Ja–En	108,990	3,303	2,804

Table 1: Number of sentences in each dataset.

Results

- Bootstrap resampling toolkit in Travatar – measure statistical significance between baseline and the methods
- Weight sharing – shared source model improves with fewer parameters when compared with the separated source
- Language dependency -

Experiment	Baseline	Separated		Shared		
		Source	Target	Source	Target	Mix
TED De–En	26.55	26.29 ± .37	26.52 ± .12	*27.20 ± .11	***27.34** ± .11	27.18 ± .21
TED En–De	21.26	21.04 ± .64	20.77 ± .10	21.63 ± .27	**21.83** ± .30	21.50 ± .29
TED Zh–En	12.54	12.52 ± .33	12.63 ± .24	*13.36 ± .41	***13.52** ± .10	*13.23 ± .09
TED En–Zh	8.97	8.94 ± .11	8.71 ± .06	9.45 ± .22	***9.58** ± .13	9.42 ± .19
TED Ja–En	5.84	*6.64 ± .26	*6.37 ± .12	*6.95 ± .07	***6.96** ± .18	*6.81 ± .16
TED En–Ja	8.40	8.58 ± .12	8.26 ± .00	8.51 ± .31	8.59 ± .08	**8.66** ± .14
Recipe Ja–En	25.34	*26.51 ± .09	*26.69 ± .15	*26.90 ± .17	***26.92** ± .10	*26.78 ± .11
Recipe En–Ja	20.81	*21.87 ± .12	*21.45 ± .14	***22.02** ± .20	*21.97 ± .09	*21.81 ± .15

Table 2: BLEU scores of our context-aware NMT in each language pair. Each score is the average of three runs. "$*$" represents the statistically significant results against the baseline at $p < 0.05$ in all the runs.

Context-Aware Neural Machine Translation Decoding

- Fuses the information from a neural translation model and the context semantics enclosed in a Semantic Space Language Model (SSLM) based on word embeddings.
- The method extends the beam search decoding process.

Semantic Space Language Model (SSLM)

- A SSLM intuitively mimics a traditional n-gram LM but it is computed over semantic information to promote translation choices that are semantically similar to the target context.
- A score is computed based on cosine similarity between the vector representations of *w* and the sum of the vector representations of the *n* target words that precede *w* in the document translation.
- w_k = current word
- $y_k - 1$ = $w_1 w_2 \ldots w_{k-1}$
- p_{uni} = maps each stop word to its relative frequency
- α = proportion of content words in training corpus
- $\vec{c}_{yk\_1}$ = sum of the vector representation of the last n non-stop words of $y_k - 1$
- sim = cosine similarity scaled to range [0,1]
- μ = word vector model maps words to vector representation
- dom = domain
- $\in$ = a small probability
- n = 30 (large n makes context cross sentence boundaries)

$$sim(\vec{a}, \vec{b}) = \frac{1}{2} \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|} + \frac{1}{2},$$

$$PSSLM(wk|yk-1) =$$

$$\begin{cases} p_{uni}(w_k) & \text{if } w_k \text{ is a SW} \\ \alpha \, sim(\vec{c}_{y_{k-1}}, \mu(w_k)) & \text{if } w_k \in \text{dom}(\mu) \text{ is not SW} \\ \varepsilon & \text{otherwise} \end{cases}$$

Shallow Fusion between NMT and SSLM

- Shallow Fusion combines the probabilities of a translation model and a language model at inference time by introducing a gating mechanism that learns to balance the weight of the additional language model.
- λ = weight
  - Weight can be adjusted using a grid search on the dev. data
- PSSLM = language model trained on target data
- Cache mechanism keeps track of context information from previously generated words extending beyond sentence boundaries
  - Adds word embeddings from previously generated words
- Computing $PSSLM(wk|yk-1)$ for each word is computationally expensive, thus computes only the N target words with the highest probabilities from the NMT

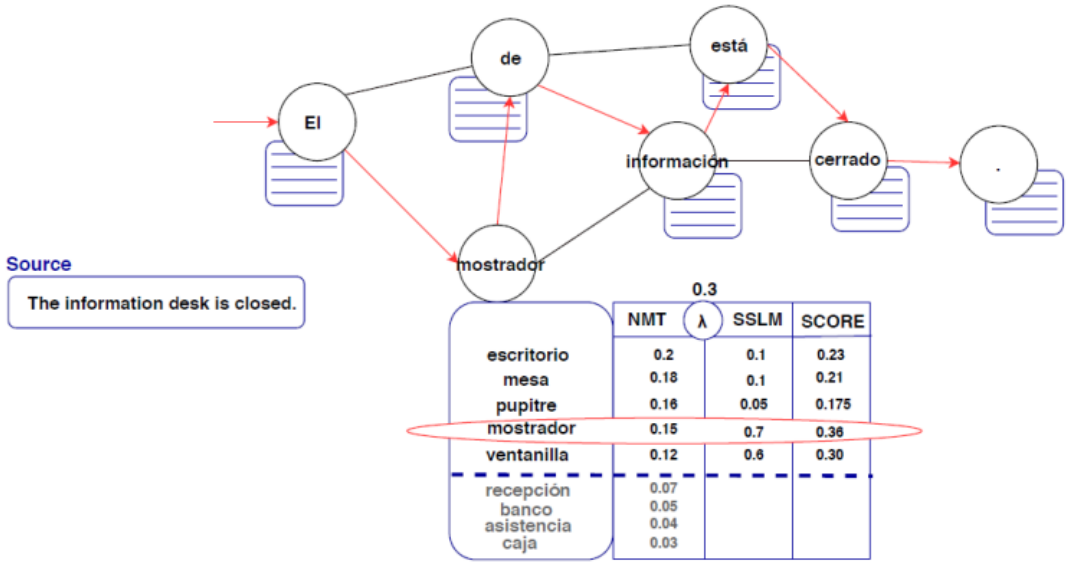


Figure 1: Sketch of the shallow fusion of an SSLM and an NMT inside the beam search algorithm. In this example, the process re-scores the $N = 5$ best candidates from the NMT model using the scores from the SSLM. Directed edges in the graph mark the path found by the beam search that maximizes the translation probability, whereas undirected edges mark possible steps considered by the beam search algorithm.

$$\hat{y} = \arg\max_{y} \left(\log p(y|x) + \lambda \log p_{SSLM}(y) \right)$$

Experiments

- Settings in OpenNMT-LUA:
  - 4 layer bidirectional RNN encoder and decoder; 800 dimension hidden layers; 500 dimension word embedding
  - Optimizer algorithm - Stochastic gradient descent ; Initial learning rate = 1; Learning decay = 0.7 after 10 epochs
  - Batch size = 64 sentences; Dropout = 0.3; Max sentence length = 50 tokens; Vocab size = 50000 for both source and target
- Data: Europarl-V7 parallel corpus, NEWS-COMMENTARY2009 corpus as validation
- NMT model shallow fused with SSLM at Epoch 20
- SSLM : WORD2VEC with CBOW with context window size = 5 and 600 dimensions.
  - Trained on Spanish side of Europarl-V7, United Nations, Multilingual United Nations and Subtitles-2012 corpora = 759 million words

Oracle Analysis

- **Oracle 1** - for each sentence translated, for each target word t:
  - Uses the attention to map ***t*** to source ***s*** and ***s*** to target word ***r*** in reference.
  - It uses maximal attention
  - Replace ***t*** with ***r*** if $t \neq r$ and ***r*** is among the ***M*** words closest to ***t*** (w.r.t. cosine Similarity)
- Accuracy increases with the number ***M*** of considered closest words.
  - Accuracy increases by 8.02 in BLEU when M encompasses target vocabulary.
- It tests whether the WVM of the SSLM properly clusters semantically-valid candidates close together

Oracle Analysis

- **Oracle 2** – similar to Oracle 1 but proceeds online with beam search
- Does not have full attention information
  - Uses a minimal threshold and refines criterion for mapping ***t*** to ***s***
    - $\mathbf{t} \xrightarrow{\mathbf{1},a} s$ ***iff*** $\mathbf{t} \xrightarrow{\mathbf{1}} s \wedge att(t,s) \geq a$
    - $\mathbf{t} \xrightarrow{\mathbf{1},a} s\ \mathbf{s} \xrightarrow{\mathbf{1}} r$
- It tests whether incomplete attention information does not hinder the oracle's ability to approximate the alignments
- **Oracle 3 –** proceeds online with beam search
  - Replacement is when ***r*** appears among the ***N*** best candidates proposed by the NMT model
- It checks if there is a wide enough margin for improvement when fusing the systems.

System results and analysis

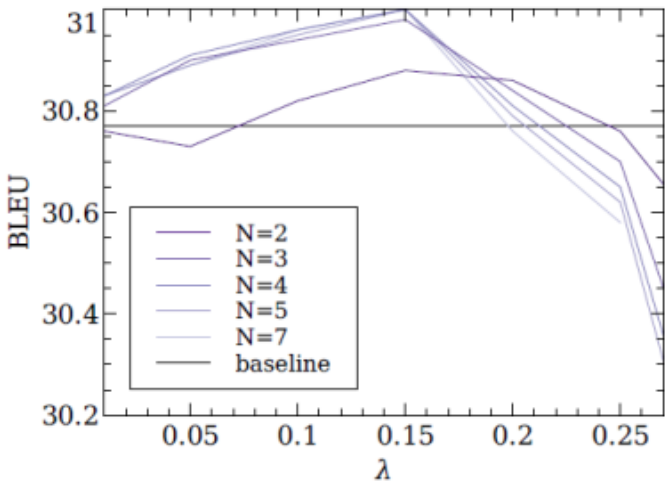


Figure 3: BLEU score of the fused system as a function of the weight λ, for several values of the parameter N.

N	BLEU↑	METEOR↑	#unknown
-	30.77	49.86	5901
2	30.88	50.17	4632
3	† 30.98	50.14	4501
4	† 31.00	50.15	4475
5	† 31.00	50.14	4459
7	† 31.00	50.14	4463
10	† 31.00	50.14	4463

Table 2: BLEU and METEOR scores obtained with the fused systems with $\lambda = 0.15$, together with the amount of unknown words in their output, where the first row corresponds to the baseline. † marks systems that are significantly different to the baseline with a p-value of 0.05, according to bootstrap resampling (Koehn, 2004).

- *Best result is achieved at* λ 0.15 *independent of* N
- N ≥ 4 converge to the same output
- Human annotators found the Fused model better than the baseline 49% of the time.

System	BLEU↑	MTR↑	N	M	a
baseline	30.77	49.86	-	-	-
ORACLE1	38.79	57.85	-	1,000	-
ORACLE2	37.32	54.35	-	1,000	0.1
ORACLE3	33.25	51.74	3	-	0.2

Table 1: BLEU and METEOR (MTR) scores obtained with the oracles defined in Section 4.2.

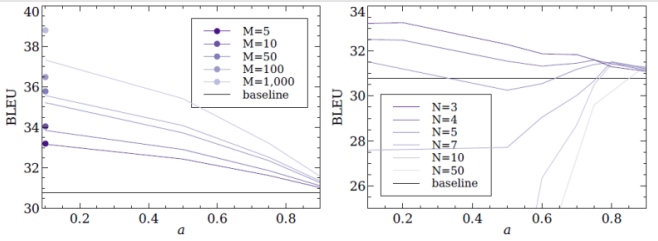


Figure 2: BLEU score of ORACLE1 (left, bullets), ORACLE2 (left, line plots), and ORACLE3 (right, line plots), as a function of the threshold a (ORACLE2 and ORACLE3) and for several values of the parameters M (ORACLE1 and ORACLE2) and N (ORACLE3). For ORACLE1 and ORACLE2, increasing the value of M beyond 1,000 does not affect the obtained scores noticeably.

Context-Aware Monolingual Repair for Neural Machine Translation

- A monolingual DocRepair model to correct inconsistencies between sentence-level translations.
  - DocRepair performs automatic post-editing on a sequence of sentence-level translations, refining translations of sentences in context of each other.
  - For training, the DocRepair model requires only monolingual document-level data in the target language.
  - It is trained as a monolingual sequence-to-sequence model that maps inconsistent groups of sentences into consistent ones. The consistent groups come from the original training data; the inconsistent groups are obtained by sampling roundtrip translations for each isolated sentence

Methodology

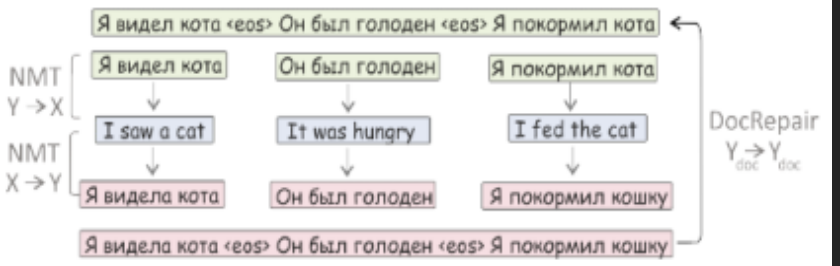


Figure 1: Training procedure of DocRepair. First, round-trip translations of individual sentences are produced to form an inconsistent text fragment (in the example, both genders of the speaker and the cat became inconsistent). Then, a repair model is trained to produce an original text from the inconsistent one.

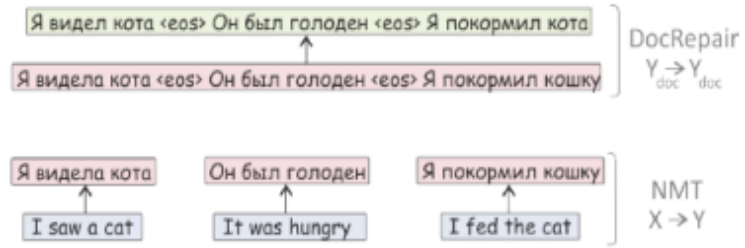


Figure 2: The process of producing document-level translations at test time is two-step: (1) sentences are translated independently using a sentence-level model, (2) DocRepair model corrects translation of the resulting text fragment.

- DocRepair model is the standard sequence-to-sequence Transformer.
  - Sentences in a group are concatenated using a reserved token-separator between sentences.
  - The Transformer is trained to correct long inconsistent pseudo-sentences into consistent ones.
  - The token-separator is removed from corrected translations.

Methodology

- Sample several groups of sentences from the monolingual data;
- for each sentence in a group,
  - translate it using a target-to-source MT model,
  - Sample a translation of this back-translated sentence in the source language using a source-to-target MT model;
- using these round-trip translations of isolated sentences, form an inconsistent version of the initial groups;
- use inconsistent groups as input for the DocRepair model, consistent ones as output.
- At test time,
  - produce translations of isolated sentences using a context-agnostic MT model;
  - apply the DocRepair model to a sequence of context-agnostic translations to correct inconsistencies between translations.

Round-Trip Translations

- The Russian monolingual data is first translated into English, using the Russian-English model and beam search = 4
- Use the English-Russian model to sample translations with temperature of 0.5.
- For each sentence, precompute 20 sampled translations and randomly choose one of them when forming a training minibatch for DocRepair.
- In training, replace each token in the input with a random one with the probability of 10%.

Test sets

- Contrastive test sets that test the ability of the system to adapt to contextual information and handle the phenomenon under consideration.
  - Contrastive translation differs from the true one only in one specific aspect
  - All contrastive translations are correct and plausible at the sentence level
- Tests the following linguistic phenomena:
  - **Deixis** – “I”, “you”, “here”, “there”, “that”, “these” etc
  - **Ellipsis** – Where are you going to? ~~I am going~~ to dance.
  - **Lexical cohesion** – which dress are you going to wear? I will wear a green frock

		distance		
	total	1	2	3
deixis	3000	1000	1000	1000
lex. cohesion	2000	855	630	515
ellipsis (infl.)	500			
ellipsis (VP)	500			

Table 1: Size of test sets: total number of test instances and with regard to the distance between sentences requiring consistency (in the number of sentences). For ellipsis, the two sets correspond to whether a model has to predict correct NP inflection, or correct verb sense (VP ellipsis).

Experimental Setup

- Data – OpenSubtitles2018 corpus English and Russian
- Training on 6m sentence pairs with a relative **time overlap (alignment quality)** of subtitle frames between source and target language subtitles of at least **0.9**.
- Monolingual data - 30m groups of 4 consecutive sentences
  - We used only documents not containing groups of sentences from general development and test sets as well as from contrastive test sets.
- BPE tokens 32000, batch size = 15000 tokens
  - Convergence in bleu score on development sets and consistency development sets
  - Return average of the last five checkpoints after training.

$$l_{rate} = scale \cdot \min(step\_num^{-0.5}, step\_num \cdot warmup\_steps^{-1.5}),$$

where $warmup\_steps = 16000$ and $scale = 4$.

Models

- Baseline, Docrepair
  - Transformer – 6 layers, 8 heads, 512 input and output dimension,

    2048 inner layer feed-forward dimensions
- Baseline 2 - two-pass CADec model (Voita et al., 2019).
  - The first pass produces sentence-level translations.
  - The second pass takes both the first-pass translation and representations of the context sentences as input and returns contextualized translations.
- CADec requires document-level parallel training data, while DocRepair only needs monolingual training data.
- Optimizer – Adam optimizer $\beta_1 = 0.9, \beta_2 = 0.98\ \varepsilon = 10 - 9$
- Varied learning rate

model	BLEU
baseline	33.91
CADec	33.86
sentence-level repair	34.12
DocRepair	**34.60**

Table 2: BLEU scores. For CADec, the original implementation was used.

model	deixis	lex. c	ell. infl.	ell. VP
baseline	50.0	45.9	53.0	28.4
CADec	81.6	58.1	72.2	**80.0**
DocRepair	**91.8**	**80.6**	**86.4**	75.2
	+10.2	+22.5	+14.4	-4.8

Table 3: Results on contrastive test sets for specific contextual phenomena (deixis, lexical consistency, ellipsis (inflection), and VP ellipsis).

all	equal	better	worse
700	367	242	90
100%	52%	35%	13%

Table 5: Human evaluation results, comparing DocRepair with baseline.

Results

- General results:
  - Evaluate on 4-sentence fragments
  - Evaluate sentence-level post editing

	total	distance 1	2	3
deixis				
baseline	50.0	50.0	50.0	50.0
CADec	81.6	84.6	84.4	75.9
DocRepair	**91.8**	**94.8**	**93.1**	**87.7**
	+ 10.2	+10.2	+8.7	+11.8
lexical cohesion				
baseline	45.9	46.1	45.9	45.4
CADec	58.1	63.2	52.0	56.7
DocRepair	**80.6**	**83.0**	**78.5**	**79.4**
	+ 22.5	+20.2	+26.5	+22.3

Table 4: Detailed accuracy on deixis and lexical cohesion test sets.

(a) **EN** No one believed me. But she did.

RU Мне никто не верил. Но она сказала.

(b)**RU** Никто мне не верил. Но она верила.

EN No one believed me. But she believed.

RU Мне никто не верил. Но она поверила.

Figure 3: (a) Example of a discrepancy caused by VP ellipsis: correct meaning is "believe", but MT produces сказала ("say"). (b) Example of producing round-trip translations. From top to bottom: target, first translation, round-trip translation. When translating from Russian, main verbs are unlikely to be translated into auxiliary ones in English, and VP ellipsis is not present.

	BLEU	deixis	lex. c.	ellipsis
2.5m	34.15	89.2	75.5	81.8 / 71.6
5m	34.44	90.3	77.7	83.6 / 74.0
30m	**34.60**	**91.8**	**80.6**	**86.4 / 75.2**

Table 6: Results for DocRepair trained on different amount of data. For ellipsis, we show inflection/VP scores.

data	deixis	lex. c.	ell. infl.	ell. VP
one-way	85.4	63.4	79.8	73.4
round-trip	84.0	61.7	78.4	67.8

Table 7: Consistency scores for the DocRepair model trained on 2.5m instances, among which 1.5m are parallel instances. Compare round-trip and one-way translations of the parallel part.

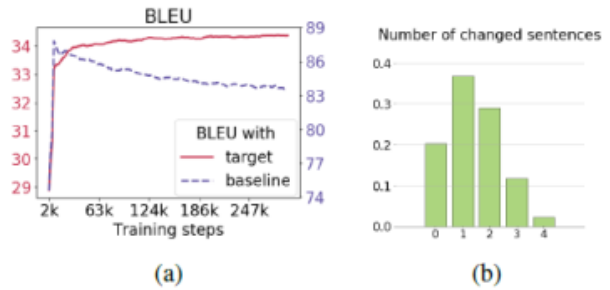


Figure 4: (a) BLEU scores progression in training. BLEU evaluated with the target translations and with the context-agnostic baseline translations (which DocRepair learns to correct). (b) Distribution in the test set of the number of changed sentences in 4-sentence fragments.

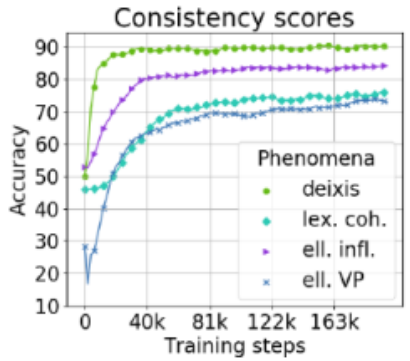


Figure 5: Consistency scores progression in training.

Results

- Varying Training Data
- One way vs round trip
- Learning Dynamics