Larger-Context Neural Machine Translation

by

Sébastien Jean

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

New York University

May, 2021

Professor Kyunghyun Cho

Acknowledgements

I first want to thank my advisor Kyunghyun Cho, with whom I have been collaborating even before joining NYU. Early on, he encouraged me in my studies and convinced me to start my PhD journey. Ever since, he has challenged me and pushed me to challenge myself, all the while being incredibly supportive.

I am especially grateful to Orhan Firat, whom I have also worked with across many years. He is both creative and methodical. I still remain impressed by his experimental spreadsheets! I learnt a lot from him and he has shaped me into a better researcher.

I am also very thankful to Stanislas Lauly, who shared an office with me for almost two years and brightened my time at NYU. I truly appreciate our many conversations. I admire his ability to maintain an upbeat attitude, even when research wasn't going as planned.

I also want to thank He He and Nizar Habash for serving on my defense committee. My experience at NYU would not have been the same without all the students, visitors and professors of the ML^2 group.

I appreciate the guidance of Marc'Aurelio Ranzato, Sumit Chopra, Melvin Johnson, Ankur Bapna and Orhan Firat during my internships at Facebook at Google. I very much appreciate the employees and fellow interns I had to opportunity to know then.

None of this would have been possible without all my teachers throughout the years. Jean-Christophe Nave introduced me to research while I was an undergraduate student. Roland Memisevic further helped me develop as a researcher and strengthened my knowledge of machine

learning during my master's degree.

I appreciate the financial assistance from NSERC, Adeptmind, Samsung and NYU GSAS throughout the years that lead to this dissertation.

Finally, I want to thank my parents Pierre and Michelle, as well as my sister Marie-Pier, for their support. Merci!

Abstract

Translation helps connect people by bridging language barriers. It can make travel more enjoyable, allow our minds to explore imaginary worlds, let us talk to others, and so on. Given the need for translation, but the limited availability of human translators, machine translation has flourished. Most systems translate sentences one by one, ignoring its context, which isn't always sufficient as the missing information can lead to incorrect or inconsistent translations. We believe that neural machine translation (NMT) is particularly well-suited to incorporate the surrounding context. Indeed, NMT systems can attend to arbitrarily distant words, while the use of continuous representations improves generalization on unseen examples.

As such, in this thesis, we extend neural machine translation to leverage information from the surrounding context. To do so, we first highlight the potential of the then-nascent NMT paradigm. We subsequently introduce architectural changes to integrate information from the surrounding document, initially starting from the preceding sentence. We further encourage models to use context from either a learning or data augmentation perspective. We also consider the efficient use of document-level neural language models for this task. While some challenges still remain, our work has helped establish larger-context translation on a solid footing, and we are optimistic about future progress.

Contents

List of Figures

List of Tables

1 Introduction

Communication, in all of its forms, is essential in many facets of life. While people speaking different languages may establish basic communication with signs, drawings and possibly other methods, the most effective approach is almost assuredly translation.

In the late 40's, Warren Weaver's memorandum sew the seeds of computer-generated translation. From then on, multiple approaches have been proposed to enable faster and more accurate translations. In particular, large-scale parallel and monolingual data can be leveraged for statistical machine translation [Brown et al. 1990]. Neural machine translation (NMT) [Cho et al. 2014; Sutskever et al. 2014; Bahdanau et al. 2015], where words are embedded in a continous vector space and fed into a neural network, allows for greater generalization and has become the favored approach today.

While most NMT models translate sentences in isolation, some crucial information may be missing, leading to impaired communication. In this thesis, we consider the problem of larger-context neural machine translation, where the models take into account neighboring sentences within a document [Jean et al. 2017; Tiedemann and Scherrer 2017; Wang et al. 2017] to produce more accurate and coherent outputs.

We first present our submission to the WMT'15 news translation task, where we demonstrate the potential of neural machine translation by applying it to multiple language pairs [Jean et al. 2015b]. We integrate advancements on large vocabularies and language model integration, obtain human judgements and establish state-of-the-art results on English to German translation.

Motivated by these promising results, but aware of the limitations of sentence-to-sentence models, we then propose network architecture changes to integrate nearby document information [Jean et al. 2017]. In addition to general translation, we also consider a pronoun translation task that specifically targets anaphora resolution [Jean et al. 2016].

We then explore different techniques to make greater use of context. We introduce a learning algorithm, based on a multilevel pairwise ranking loss, that explicitly encourages the translation model to take additional context into account [Jean and Cho 2019]. The resulting model does indeed rely more on extra-sentential information, but can become less robust. The model is sometimes unable to produce a reasonable translation, especially when given random or unrelated contexts. Alternatively, we can further encourage the use of context with data augmentation [Jean et al. 2019]. We generate artificial context for some training examples, and also validate the effectiveness of back-translating monolingual data.

In the next chapter, we consider an approach to integrate language models into document-level neural machine translation [Jean and Cho 2020]. In particular, we reformulate a noisy channel framework [Yu et al. 2020] so that the predictions now depend on a sentence-level translation system, as well as competing sentence and document-level language models. We train a merging module to dynamically control the contribution of the language models.

1.1 List of contributions

- **Sébastien Jean***, Orhan Firat*, Kyunghyun Cho, Roland Memisevic and Yoshua Bengio. Montreal Neural Machine Translation Systems for WMT'15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation, 2015*

  We collectively decided to participate in the WMT news translation shared task. Orhan Firat and I ran most of the experiments, under the close supervision and advice of Kyunghyun Cho, and participated in the human evaluation campaign. The three of us co-wrote the pa-

per. Roland Memisevic and Yoshua Bengio advised us throughout the project.

- **Sébastien Jean***, Stanislas Lauly*, Orhan Firat and Kyunghyun Cho. Does Neural Machine Translation Benefit from Larger Context? *arXiv preprint, 2017*

  Kyunghyun Cho proposed the general idea of larger-context neural machine translation. Stanislas Lauly and I refined the idea jointly and co-wrote the paper, with valuable help from Orhan Firat and Kyunghyun Cho. Orhat Firat also ran some preliminary experiments.

- **Sébastien Jean***, Stanislas Lauly*, Orhan Firat and Kyunghyun Cho. Neural Machine Translation for Cross-Lingual Pronoun Prediction. In *Proceedings of the Third Workshop on Discourse in Machine Translation, 2017*

  (Follow-up paper to "Does Neural Machine Translation Benefit from Larger Context?") Stanislas Lauly, Orhan Firat, Kyunghyun Cho and I jointly started the project. Stanislas Lauly and I ran most of the experiments, had numerous discussions and wrote most of the paper. Orhan Firat and Kyunghyun Cho helped to refine the ideas and revised the paper.

- **Sébastien Jean** and Kyunghyun Cho. Context-Aware Learning for Neural Machine Translation. *arXiv preprint, presented at the 3rd Workshop on Neural Generation and Translation, 2019*

  I conceived the initial idea and ran the experiments. Kyunghyun Cho had numerous discussions with me and helped refining the idea. We co-wrote the paper.

- **Sébastien Jean**, Ankur Bapna and Orhan Firat. Fill in the Blanks: Imputing Missing Sentences for Larger-Context Neural Machine Translation. *arXiv preprint, 2019*

  I started the project and proposed most of the ideas. Ankur Bapna and Orhan Firat advised me, refined some of the ideas and helped with debugging. We all co-wrote the paper.

- **Sébastien Jean** and Kyunghyun Cho. Log-Linear Reformulation of the Noisy Channel

Model for Document-Level Neural Machine Translation. In *Proceedings of the Fourth Workshop on Structured Prediction for NLP, 2020*

I initiated the project, ran the experiments and wrote most of the paper. Kyunghung Cho suggested the use of language models for document-level NMT years earlier, advised me throughout the project, proposed some analysis techniques and revised the paper.

2 Background

In this chapter, we introduce different approaches for sentence-level machine translation, starting from rule-based translation and culminating with neural machine translation. We also motivate the use of context beyond the sentence, describe previous efforts to do so, and justify why neural machine translation is suitable for this challenge.

2.1 Sentence-level translation

Many machine translation systems translate sentences in isolation. We briefly describe rule-based and statistical machine translation. We then present neural machine translation in more detail as this approach is used throughout this dissertation.

2.1.1 Rule-based translation

Rule-based machine translation (RBMT) systems such as SYSTRAN [Toma 1977] helped establish the viability of computer-generated translations. These systems are composed of two main components, lexicons and grammar rules [Nirenburg 1989; Lagarda et al. 2009; Torregrosa et al. 2019]. Lexicons contain morphological, syntactic, and semantic information, while the grammar rules formalize the structure of the languages. Building these resources rely on expert human knowledge.

Most rule-based translation systems follow a pipeline approach. For example, the Lucy LT

system [Alonso and Thurmair 2003] generates translations in three major steps: analysis, transfer and generation. In the analysis phase, the source sentence is deconstructed and represented as an annotated tree. That tree is then further annotated and transformed according to the target language, after which the final translation can be produced.

2.1.2 Statistical lexical translation

Instead of relying on hand-crafted rules, statistical machine translation (SMT) systems learn possible translations from large amounts of parallel data [Koehn 2009]. Some of the simplest models rely on word-by-word translations, whose probabilities are stored in a translation table t. These translation models also rely on a alignment function a that maps the target word at position i to source position j.

In IBM model 2 [Brown et al. 1990, 1993], given a source sentence $X = (x_1, \ldots, x_{l_X})$ and its target translation $Y = (y_1, \ldots, y_{l_Y})$, the probability of a target sentence given an alignment $a = (a(1), ..., a(l_Y))$ follows

$$p(Y, a|X) = \epsilon \prod_{j=1}^{l_Y} t(y_j|x_{a(j)}) p(a(j)|j, l_Y, l_X)$$

The parameters of the translation table and the alignment probabilities are iteratively fitted with the Expectation-Maximization (EM) algorithm [Moon 1996]. In the expectation step, the probabilities of the unobserved alignments are computed. In the maximization step, the model parameters are updated according to count statistics over the training data.

More complex lexical translation models (IBM 3-5, HMM) take into account perplexity (the number of target words generated from each source word), reorder generated words or propose more sophisticated alignment models.

These models can further leverage monolingual data by integrating language models through

the noisy channel model. Applying Bayes' rule, the probability of a sentence is expressed as

$$p(Y|X) \propto p(X|Y)p(Y), \tag{2.1}$$

which requires using a reverse-direction translation model in conjunction with the language model.

N-gram language models are built from large monolingual corpora. For a given span n, the conditional empirical probability of a token x_i is given by

$$p(x_i|x_{i-n+1}, \ldots, x_{i-1}) = \frac{C(x_{i-n+1}, \ldots, x_i)}{C(x_{i-n+1}, \ldots, x_{i-1})}, \tag{2.2}$$

where C denotes the count of the word subsequence. Lower-order n-grams provide more robust statistics, but only consider a very limited context. Conversely, higher order n-grams models look further into the past, but can suffer from sparsity issues, especially for rarer tokens. To take advantage of these different benefits and account for unseen n-grams, the probabilities of different order n-gram models may be smoothed [Chen and Goodman 1999]. In particular, Kneser-Ney smoothing is commonly employed within the language models used by SMT systems [Kneser and Ney 1995].

2.1.3 Phrase-based translation

The correct translation of a source word often depends on its surroundings. As such, phrase-based translation extends lexical translation by instead considering phrases, or sequences of consecutive words, as its atomic unit [Koehn et al. 2003].

One main component of such systems is a phrase translation table, which assigns probabilities between phrases of the two languages. To generate that table, word alignments are first obtained through a lexical translation model. To obtain more robust alignments, the word-based models may be applied in both translation directions, and then merged through some heuristics.

Given the alignment between two sentences, consistent phrase pairs are extracted. A phrase pair is consistent if none if its words are aligned to another word out of it. By collecting phrase pair counts over a large corpora, the conditional probabilities of phrase pairs may be estimated by their relative frequency and stored into the phrase table.

Other important features of phrase-based statistical translation systems include a reordering and language model. Extending the noisy channel approach, the full system consists of a log-linear combination of all considered features

$$p(Y|X) = \exp \sum_{i=1}^{n} \lambda_i h_i(X, Y). \tag{2.3}$$

To improve the modelling abilities of these systems, additional features h_i include alignment log-probabilities, lexical log-probabilities, word count (to control length), lexicon-based features and many others [Och et al. 2004]. The coefficients of all these features may be efficiently adjusted with minimum error rate training (MERT) [Och 2003].

Other approaches for machine translation include example-based [Somers 1999], hierarchical phrase-based [Chiang 2005] and syntax-based translation [Yamada and Knight 2001].

2.1.4 Neural machine translation

Neural machine translation (NMT) systems are also built from large amounts of parallel data, but instead of relying on a phrase table and other features, they are parameterized as a sequence-to-sequence neural networks [Kalchbrenner and Blunsom 2013; Cho et al. 2014; Sutskever et al. 2014].

2.1.4.1 Modelling

A neural machine translation system is a parameterized function f_θ that computes the conditional probability $p(Y|X)$ of a target sentence Y given a source X, where θ are the model's parameters.

NMT systems often follow the encoder-decoder framework. The source words or tokens are first embedded into a continuous space. These vectors are fed through an encoder network that produces final source representations. In the auto-regressive formulation, a decoder then predicts each target word according to the preceding ones, as well as the source representations.

$$p(Y|X) = \prod_{i=1}^{l_Y} p(y_i|X, y_{<i}) \tag{2.4}$$

The encoder and decoder networks may be parameterized in multiple ways, among others as recurrent neural network (RNN) or transformers [Elman 1990; Vaswani et al. 2017]. Given a sequence of embeddings $(\mathbf{x}_1, \ldots, \mathbf{x}_{l_X})$, a recurrent neural network sequentially applies a transformation $h_i = f(\mathbf{x}_i, h_{i-1})$. Specialized recurrent functions, such as the long short-term memory (LSTM) and the gated recurrent unit (GRU), may be used to facilitate the flow of information across long sentences and prevent undesirable vanishing or exploding gradients [Hochreiter and Schmidhuber 1997; Cho et al. 2014].

These functions rely on multiple gates of the form $g = \sigma(W_*\mathbf{x}_i + U_*h_{i-1} + b)$, where W_* and U_* are weight matrices and b a bias vector. σ is the logistic sigmoid function, applied element-wise, resulting in values between 0 and 1. In particular, the GRU has a reset gate r and an update gate z. The reset gate is used to compute an intermediate state $\widetilde{h_i} = tanh(W\mathbf{x}_i + U(r \odot h_{i-1}))$, controlling how much the previous hidden state is considered. The update gate is used to obtain the new hidden state $h_i = (z \odot h_{i-1} + (\mathbf{1} - z) \odot \widetilde{h_i})$.

To allow the decoder to adaptively integrate information from all source representations, it can be augmented with an attention mechanism [Bahdanau et al. 2015]. An attention mechanisms computes affinity scores between a query vector q and a set of keys $(k_1, ..., k_K)$. These scores $e_1, \ldots, e_K$ are converted to probabilities $\lambda_1, \ldots, \lambda_K$ with a softmax function $\lambda_i = \frac{\exp(e_i)}{\sum_{j=1}^{K} \exp(e_j)}$. Values $(v_1, ..., v_K)$ are then combined as a weighted average $\sum_{i=1}^{K} \lambda_i v_i$. Within an RNN decoder, the decoder hidden states act as queries, whereas the keys and values are the source representa-

tions (or transformations thereof). The weighted average of these values can then be used as an additional input to the recurrent unit.

For additional expressiveness, multi-head attention computes multiple parallel attentions after independently transforming the queries, keys and values [Vaswani et al. 2017]. The resulting vectors may then be combined.

Attention mechanisms may also form the backbone of a translation system, as in the case of transformers [Vaswani et al. 2017]. To account for word order, positional embeddings are first added to the word embeddings. The encoder consists of a stack of layers, each of which alternates between self-attention, where the queries, keys and values are shared (up to feed-forward transformations) and feed-forward projections. The transformer decoder layers have a similar structure, with additional cross-attention between their hidden states and the final source representations. To prevent the model from incorporating future information, self-attention within the decoder is restricted to the current and past tokens. Compared to RNN systems, transformers simultaneously integrate information from all tokens, regardless of their distance within the sequence, and exhibit superior parallelization.

To obtain token-level probabilities $p(y_i|X, y_{<i})$, the hidden state of the decoder is projected into a vector that assigns a score to every target vocabulary token. Similarly to attention mechanisms, these scores are normalized with a softmax function.

2.1.4.2 Learning

The goal of neural machine translation is to produce appropriate translations for possibly unseen source sentences. We can do so by learning the parameters θ of the model. We define learning as a combination of the objective function used to train the model, the initialization of the parameters, as well as the optimization procedure to refine the parameter values.

Given a parallel corpora $(\mathcal{X} * \mathcal{Y}) = ((X^{(1)}, Y^{(1)}), \ldots, (X^{(N)}, Y^{(N)}))$, autoregressive neural

machine translation systems are generally trained to minimize the negative log-likelihood (NLL)

$$\mathcal{L}_\theta(\mathcal{X} * \mathcal{Y}) = -\sum_{j=1}^{N} \log p(Y^{(j)}|X^{(j)}) = -\sum_{j=1}^{N}\sum_{i=1}^{l_{Y^{(j)}}} \log p(y_i^{(j)}|X^{(j)}, y_{<i}^{(j)}). \tag{2.5}$$

As we are mostly concerned about the generalization ability to unseen sentences, the NLL objective function may be supplemented by different regularization terms. For example, L2 regularization constrains the magnitude of the model weights, preventing overfitting to the training data, where the model would memorize, but not generalize well. These regularization terms may also encourage particular behaviors. For example, we could add one that would favor short or long translations.

The parameters of the model are first initialized to values that will make further optimization easier [Sutskever et al. 2013]. Given the intractability of the exact global minimum and the large size of parallel corpora, parameters are generally updated with stochastic gradient descent (SGD), or variants thereof [Kiefer and Wolfowitz 1952]. Given a batch of parallel sentences $(\hat{\mathcal{X}} * \hat{\mathcal{Y}}) \subset (\mathcal{X} * \mathcal{Y})$, the model parameters θ are updated as

$$\theta \leftarrow \theta - \alpha \nabla \mathcal{L}_\theta(\hat{\mathcal{X}} * \hat{\mathcal{Y}}), \tag{2.6}$$

so that the loss decreases on the batch.

It is important to choose an appropriate learning rate α. If it is too low, the model will converge slowly. However, if it is too high, given that the gradient only measures changes in an infinitesimally small neighborhood around the current parameters, the loss function on the batch may increase, leading to unstable training.

For faster convergence, SGD may be supplemented with momentum [Sutskever et al. 2013]. The learning rate α may also be controlled adaptively with Adadelta, Adam or similar adaptive optimizers [Zeiler 2012; Kingma and Ba 2014]. For example, Adam maintains bias-corrected estimates of the first and second moments of the gradients (mean and uncentered variance) to better

adjust the learning rate. Pre-defined schedules, which often decrease the learning rate in the later stages of training, may also be employed [Vaswani et al. 2017].

2.1.4.3 Inference and evaluation

Inference is the application of the model to possibly unseen examples, and it can take many forms. First, we can compute the conditional probability $P(Y|X)$ of a sentence pair.

If the model only scores reference translations on a test set $(\mathcal{X}' * \mathcal{Y}')$, it can be intrinsically evaluated by its perplexity

$$\exp\left(-\frac{\sum_{j=1}^{N'}\sum_{i=1}^{l_{Y'(j)}}\log p\left(y_i'^{(j)}|X'^{(j)}, y_{<i}'^{(j)}\right)}{\sum_{j=1}^{N'} l_{Y'(j)}}\right).$$

A lower perplexity indicates that the translation model assigns, on average, high probability to the reference translations.

More frequently, given the source X, the model will be tasked to generate a likely translation Y. As it is computationally expensive to search for the most probable translation, approximate approaches are generally used to translate sentences. In particular, beam search generates outputs sequentially, pruning the search space at each time step to conserve only the most promising partial hypotheses [Graves 2012].

The translation of a sentence may be evaluated by humans, which can for example assign it fluency and adequacy scores, or rank it compared to other translations of the same sentence [Snover et al. 2009; Bojar et al. 2016]. Scores (or ranks) can be aggregated over the test set to obtain an estimate of the model's performance.

Given the high cost and slow speed of human evaluation, automated metrics, designed to correlate well with human evaluation, can also be used. The most common is BLEU [Papineni et al. 2002], which computes the geometric average of modified n-gram precisions ($n = 1$ to 4), with an additional brevity penalty that penalizes outputs that are shorter than the reference.

Phenomenon	Source	Translation(s)
Anaphora resolution	I bought flowers. They are pretty.	J'ai acheté des fleurs. Elles sont belles.
Lexical cohesion	Do you some soup? Some soup?	Tu veux de la soupe? De la soupe? Tu veux du potage? Du potage?
Verb phrase ellipsis	Did you win? Yes, I did.	As-tu gagné? Oui, j'ai gagné.
Social deixis	I wrote to you. Then, I talked to you.	Je t'ai écrit. Puis, je t'ai parlé. Je vous ai écrit. Puis, je vous ai parlé.

Table 2.1: Examples of context-dependent translations. Lexical cohesion example from [Bawden et al. 2018].

2.2 Document-level translation

Sentence-level translation, even if performed by human experts, is inherently limited. Some information needed to confidently translate a source sentence might be missing, although it may be present within the neighboring sentences. Without that information, a sentence-level system may still be correct at times, but it will inevitably make mistakes. Even if multiple sentence-level translations are valid, they must still remain coherent across the generated target document.

Multiple linguistic phenomena sometimes require extra-sentential knowledge. We describe some, although the list is certainly not exhaustive. Examples are presented in Table 2.1.

Anaphora resolution is the ability to identify what a pronoun or noun phrase refers to [Mitkov 2002]. For example, given the source segment "*I bought flowers. They are pretty.*", the correct French translation of *They* as *Elles* relies on the fact that *They* refers to *flowers*. As *flowers* will be translated into the feminine word *fleurs*, the corresponding pronoun should agree in gender. The translation of the referent may also affect grammatical agreement within the rest of the output[Matthews 1991]. Here, *pretty* will be translated as the feminine adjective *belles*, instead of the masculine *beaux*.

Source-side context will often be sufficient, but if the antecedent has multiple valid translations with different grammatical properties, target-side context provides the necessary information to ensure a reliable translation.

Semantically related words, either through reiteration (same entity) or collocation (related entities), foster lexical cohesion, allowing a document to "stick together" [Halliday et al. 1976; Morris and Hirst 1991]. For instance, a word such as *soup* could be translated as either *soupe* or *potage*, which have similar, but not exactly identical meanings [Bawden et al. 2018]. Additionally, especially for languages with different alphabets, there can be multiple appropriate translations (or sometimes transliterations) of a proper noun. If the translation occurs multiple times within a document, consistent choices should be taken.

Omitted words in one language, such as the verb after *do* in English, must sometimes be explicitly translated in others such as French and Russian [Huang 2000; Voita et al. 2019b]. For example, given "*Did you win? Yes, I did.*", the second sentence would become "*Oui, j'ai gagné.*", referring to the missing *win*.

Deixis [Weissenborn and Klein 1982] is the grounding of utterances in specific contexts, especially in time, place or person. Social deixis includes the use of different formality levels [Levinson 1983], such as the T-V distinction when translating *you*. Namely, the segment "*I wrote to you. Then, I talked to you.*" could be translated as the singular informal "*Je t'ai écrit. Puis, je t'ai parlé.*" or the formal (or plural) "*Je vous ai écrit. Puis, je vous ai parlé.*".

More generally, it should be preferable to ensure stylistic consistency. Especially for a longer document, it should appear that only a single person (or system) translated it. For instance, a popular book such as "*Le petit prince*" has been translated into English multiple times, with different translators using various degrees of lyricism, archaic or more modern vocabularies, and so on [Hsieh 2017].

2.2.1 Extensions to statistical machine translation

Most rule-based and statistical translation systems only translate sentences in isolation. Even within sentences, the effective context often only spans a few tokens as most long phrases are not sufficiently common to establish reliable statistics. Likewise, n-gram language models generally

only directly account for a few tokens in the past, even when the probabilities are smoothed.

Nevertheless, document-level information has been integrated in these systems. In particular, anaphora resolution rules (pronoun agreement with its antecedent) may be added to RBMT models [Mitkov 1999]. For English-to-French translation, Le Nagard and Koehn [2010] identify the antecedent of "it" and "they" with a coreference resolution system and align it to the corresponding French word. They identify the gender of that word (masculine, feminine or neutral) in order to annotate the data with that information, for example replacing *it* by *it-feminine*, after which a standard sentence-level SMT system may be built. Alternatively, after identifying coreference links, a separate word-dependency model may be integrated as a separate feature within the SMT log-linear model [Hardmeier and Federico 2010].

Hardmeier et al. [2012] propose an iterative decoding algorithm which maintains a representation of the translation of an entire document. This allows for the effective integration of arbitrary cross-sentence features, such as a semantic document language model, which improve lexical cohesion. In particular, for content words, the semantic document language model is conditioned on a latent semantic analysis model (LSA) that considers the previous 30 context words within the document [Landauer et al. 1998]. Additional features that target lexical cohesion or other phenomena such as verb tense agreement and discourse connectives may also be integrated into SMT systems [Xiong et al. 2013; Meyer 2015].

2.2.2 Potential advantages of neural machine translation

While neural machine translation was initially introduced for sentence-level translation, it exhibits multiple characteristics that make it particularly suitable for document-level translation.

In particular, information between distant tokens may be selectively propagated in a single step through attention mechanisms. As neural networks can learn complex non-linear functions [Cybenko 1989; Barron 1994; Montúfar et al. 2014], they can potentially use relevant contextual information appropriately, while still being able to ignore it if or when it is not useful.

While the computational complexity of standard attention mechanisms grows proportionally to the length of the queries and of the keys, this potential issue can be circumvented through hierarchical modeling, sparse attention masks or other techniques [Tay et al. 2020].

Moreover, the continuous nature of neural representations is also suitable for document-level neural machine translation. Long phrases spanning multiple sentences observed during training are unlikely to appear again in unseen examples. Continuous representations address this data sparsity [Bengio et al. 2003]. Given sufficient data, neural machine translation systems can generalize to unseen examples [Li et al. 2019]. The final token representations will generally differ from those obtained during the training phase. Yet, they can be sufficiently similar, yielding accurate predictions.

3 Neural machine translation at WMT'15

Neural machine translation (NMT) systems have recently achieved results comparable to the state of the art on a few translation tasks, including English→French and English→German. The main purpose of our submission to WMT'15 is to evaluate this new approach on a greater variety of language pairs. Furthermore, the human evaluation campaign may help us and the research community to better understand the behaviour of our systems. We use the RNNsearch architecture, which adds an attention mechanism to the encoder-decoder. We also leverage some of the recent developments in NMT, including the use of large vocabularies, unknown word replacement and, to a limited degree, the inclusion of monolingual language models.

3.1 Introduction

Neural machine translation (NMT) is a recently proposed approach for machine translation that relies only on neural networks. The NMT system is trained end-to-end to maximize the conditional probability of a correct translation given a source sentence [Kalchbrenner and Blunsom 2013; Cho et al. 2014; Sutskever et al. 2014; Bahdanau et al. 2015]. Although NMT has only recently been introduced, its performance has been found to be comparable to the state-of-the-art statistical machine translation (SMT) systems on a number of translation tasks

[Luong et al. 2015; Jean et al. 2015a]. The main purpose of our submission to WMT'15 is to test the NMT system on a greater variety of language pairs. As such, we trained systems on Czech↔English, German↔English and Finnish→English. Furthermore, the human evaluation campaign of WMT'15 will help us better understand the quality of NMT systems which have mainly been evaluated using the automatic evaluation metric such as BLEU [Papineni et al. 2002].

Most NMT systems are based on the *encoder-decoder* architecture [Cho et al. 2014; Sutskever et al. 2014; Kalchbrenner and Blunsom 2013]. The source sentence is first read by the encoder, which compresses it into a real-valued vector. From this vector representation the decoder may then generate a translation word-by-word. One limitation of this approach is that a source sentence of any length must be encoded into a fixed-length vector. To address this issue, our systems for WMT'15 use the *RNNsearch* architecture from [Bahdanau et al. 2015]. In this case, the encoder assigns a context-dependent vector, or annotation, to every source word. The decoder then selectively combines the most relevant annotations to generate each target word.

NMT systems often use a limited vocabulary of approximately 30, 000 to 80, 000 target words, which leads them to generate many out-of-vocabulary tokens (⟨UNK⟩). This may easily lead to the degraded quality of the translations. To sidestep this problem, we employ a variant of importance sampling to help increase the target vocabulary size [Jean et al. 2015a]. Even with a larger vocabulary, there will almost assuredly be words in the test set that were unseen during training. As such, we replace generated out-of-vocabulary tokens with the corresponding source words with a technique similar to those proposed by [Luong et al. 2015].

Most NMT systems rely only on parallel data, ignoring the wealth of information found in large monolingual corpora. On Finnish→English, we combine our systems with a recurrent neural network (RNN) language model by the recently proposed *deep fusion* [Gulcehre et al. 2015]. For the other language pairs, we tried reranking n-best lists with 5-gram language models [Chen and Goodman 1999].

3.2 System Description

In this section, we describe the RNNsearch architecture as well as the additional techniques we used.

Mathematical Notations Capital letters are used for matrices, and lower-case letters for vectors and scalars. x and y are used for a word in source and target sentences, respectively. We boldface them into $\mathbf{x}$, $\mathbf{y}$ and $\hat{\mathbf{y}}$ to denote their continuous-space representation (word embeddings).

3.2.1 Bidirectional Encoder

To encode a source sentence $(x_1, \ldots, x_{T_x})$ of length T_x into a sequence of annotations, we use a bidirectional recurrent neural network [Schuster and Paliwal 1997]. The bidirectional recurrent neural network (BiRNN) consists of two recurrent neural networks (RNN) that read the sentence either forward (from left to right) or backward. These RNNs respectively compute the sequences of hidden states $(\overrightarrow{h}_1, \ldots, \overrightarrow{h}_{T_x})$ and $(\overleftarrow{h}_1, \ldots, \overleftarrow{h}_{T_x})$. These two sequences are concatenated at each time step to form the annotations $\left(h_1, \ldots, h_{T_x}\right)$. Each annotation h_i summarizes the entire sentence, albeit with more emphasis on word x_i and the neighbouring words.

We built the BiRNN with gated recurrent units (GRU, [Cho et al. 2014]), although long short-term memory (LSTM) units could also be used [Hochreiter and Schmidhuber 1997], as in [Sutskever et al. 2014]. More precisely, for the forward RNN, the hidden state at the i-th word is computed as

$$\overrightarrow{h}_i = \begin{cases} (1 - \overrightarrow{z}_i) \odot \overrightarrow{h}_{i-1} + \overrightarrow{z}_i \odot \overrightarrow{\underline{h}}_i & , \text{if } i > 0 \\ 0 & , \text{if } i = 0 \end{cases}$$

where

$$\begin{aligned}
\overrightarrow{\underline{h}}_i &= \tanh\left(\overrightarrow{W}\mathbf{x}_i + \overrightarrow{U}\left[\overrightarrow{r}_i \odot \overrightarrow{h}_{i-1}\right] + \overrightarrow{b}\right) \\
\overrightarrow{z}_i &= \sigma\left(\overrightarrow{W}_z\mathbf{x}_i + \overrightarrow{U}_z\overrightarrow{h}_{i-1}\right) \\
\overrightarrow{r}_i &= \sigma\left(\overrightarrow{W}_r\mathbf{x}_i + \overrightarrow{U}_r\overrightarrow{h}_{i-1}\right).
\end{aligned}$$

To form the new hidden state, the network first computes a proposal $\overrightarrow{\underline{h}}_i$. This is then additively combined with the previous hidden state $\overrightarrow{h}_{i-1}$, and this combination is controlled by the update gate $\overrightarrow{z}_i$. Such gated units facilitate capturing long-term dependencies.

3.2.2 Attentive Decoder

After computing the initial hidden state $s_0 = \tanh\left(W_s\overleftarrow{h}_1\right) + b_s$, the *RNNsearch* decoder alternates between three steps: *Look*, *Generate* and *Update*.

During the *Look* phase, the network determines which parts of the source sentence are most relevant. Given the previous hidden state s_{i-1} of the decoder recurrent neural network (RNN), each annotation h_j is assigned a score e_{ij}:

$$e_{ij} = v_a^\top \tanh\left(W_a s_{i-1} + U_a h_j\right).$$

Although a more complex scoring function can potentially learn more non-trivial alignments, we observed that this single-hidden-layer function is enough for most of the language pairs we considered.

These scores e_{ij} are then normalized to sum to 1:

$$\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{k=1}^{T_x}\exp\left(e_{ik}\right)}, \tag{3.1}$$

which we call alignment weights.

The context vector c_i is computed as a weighted sum of the annotations $(h_1, ..., h_{T_x})$ according to the alignment weights:

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j.$$

This formulation allows the annotations with higher alignment weights to be more represented in the context vector c_i.

In the *Generate* phase, the decoder predicts the next target word. We first combine the previous hidden state s_{i-1}, the previous word y_{i-1} and the current context vector c_i into a vector $\tilde{t}_i$:

$$\tilde{t}_i = U_o s_{i-1} + V_o \mathbf{y}_{i-1} + C_o c_i + b_o.$$

We then transform $\tilde{t}_i$ into a hidden state m_i with an arbitrary feedforward network. In our submission, we apply the maxout non-linearity [Goodfellow et al. 2013] to $\tilde{t}_i$, followed by an affine transformation.

For a target vocabulary V, the probability of word y_i is then

$$p(y_i | s_{i-1}, y_{i-1}, c_i) = \frac{\exp\left(\hat{\mathbf{y}}_i^\top m_i + b_{y_i}\right)}{\sum_{y \in V} \exp\left(\hat{\mathbf{y}}^\top m_i + b_y\right)}. \tag{3.2}$$

Finally, in the *Update* phase, the decoder computes the next recurrent hidden state s_i from the context c_i, the generated word y_i and the previous hidden state s_{i-1}. As with the encoder we use gated recurrent units (GRU).

Table 3.1 summarizes this three-step procedure. We observed that it is important to have *Update* to follow *Generate*. Otherwise, the next step's *Look* would not be able to resolve the uncertainty embedded in the previous hidden state about the previously generated word.

Phase	Output ← Input
Look	$c_i \leftarrow s_{i-1}, (h_1, ..., h_{T_x})$
Generate	$y_i \leftarrow s_{i-1}, y_{i-1}, c_i$
Update	$s_i \leftarrow s_{i-1}, y_i, c_i$

Table 3.1: Summary of *RNNsearch* decoder phases

3.2.3 Very Large Target Vocabulary Extension

Training an RNNsearch model with hundreds of thousands of target words easily becomes prohibitively time-consuming due to the normalization constant in the softmax output (see Eq. (3.2).) To address this problem, we use the approach presented in [Jean et al. 2015a], which is based on importance sampling [Bengio and Sénécal 2008]. During training, we choose a smaller vocabulary size τ and divide the training set into partitions, each of which contains approximately τ unique target words. For each partition, we train the model as if only the unique words within it existed, leaving the embeddings of all the other words fixed.

At test time, the corresponding subset of target words for each source sentence is not known in advance, yet we still want to keep computational complexity manageable. To overcome this, we run an existing word alignment tool on the training corpus in advance to obtain word-based conditional probabilities [Brown et al. 1993]. During decoding, we start with an initial target vocabulary containing the K most frequent words. Then, reading a few sentences at once, we arbitrarily replace some of these initial words by the K' most likely ones for each source word.[1]

No matter how large the target vocabulary is, there will almost always be those words, such as proper names or numbers, that will appear only in the development or test set, but not during training. To handle this difficulty, we replace unknown words in a manner similar to [Luong et al. 2015]. More precisely, for every predicted out-of-vocabulary token (⟨UNK⟩), we determine its most likely origin by choosing the source word with the largest alignment weight α_{ij} (see Eq. (3.1).) We may then replace ⟨UNK⟩ by either the most likely word according to a dictionary,

[1] This step differs very slightly from [Jean et al. 2015a], where the sentence-specific words were added on top of the K common ones instead of replacing them.

or simply by the source word itself. Depending on the language pairs, we used different heuristics according to performance on the development set.

3.2.4 Integrating Language Models

Unlike some data-rich language pairs, most of the translation tasks do not have enough parallel text to train end-to-end machine translation systems. To overcome with this issue of low-resource language pairs, external monolingual corpora is exploited by using the method of *deep fusion* [Gulcehre et al. 2015].

In addition to the RNNsearch model, we train a separate language model (LM) with a large monolingual corpus. Then, the trained LM is plugged into the decoder of the trained RNNsearch with an additional controller network which modulates the contributions from the RNNsearch and LM. The controller network takes as input the hidden state of the LM, and optionally RNNsearch's hidden state, and outputs a scalar value in the range $[0, 1]$. This value is multiplied to the LM's hidden state, controlling the amount of information coming from the LM. The combined model, the RNNsearch, the LM and the controller network, is jointly tuned as the final translation model for a low-resource pair.

In our submission, we used recurrent neural network language model (RNNLM). More specifically, let s_i^{LM} be the hidden state of a pre-trained RNNLM and s_i^{TM} be that of a pre-trained RNNsearch at time i. The controller network is defined as

$$g_t = \sigma\left(V_g^\top s_t^{\text{LM}} + W_g^\top s_t^{\text{TM}} + b_g\right),$$

where σ is a logistic sigmoid function, v_g, w_g and b_g are model parameters. The output of the controller network is multiplied to the LM's hidden state s_i^{LM}:

$$p_t^{\text{LM}} = s_t^{\text{LM}} \odot g_t.$$

The *Generate* phase in Sec. 3.2.2 is updated as,

$$\tilde{t}_i = U_o^{\text{TM}} s_{i-1}^{\text{TM}} + U_o^{\text{LM}} p_{t-1}^{\text{LM}} + V_o \mathbf{y}_{i-1} + C_o c_i + b_o.$$

This lets the decoder fully use the signal from the translation model, while the the signal from the LM is modulated by the controller output.

Among all the pairs of languages in WMT'15, Finnish↔English translation has the least amount of parallel text, having approximately $2M$ aligned sentences only. Thus, we use the *deep fusion* for the Fi-En in the official submission. However, we further experimented German→English, having the second least parallel text, and Czech→English, which has comparably larger data. We include the results from these two language pairs here for completeness.

3.3 Experimental Details

We now describe the settings of our experiments. Except for minor differences, all the settings were similar across all the considered language pairs.

3.3.1 Data

All the systems, except for the English→German (En→De) system, were built using all the data made available for WMT'15. The En→De system, which was showcased in [Jean et al. 2015a], was built earlier than the others, using only the data from the last year's workshop (WMT'14.)

Each corpus was tokenized, but neither lowercased nor truecased. We avoided badly aligned sentence pairs by removing any source-target sentence pair with a large mismatch between their lengths. Furthermore, we removed sentences that were likely written in an incorrect language, either with a simple heuristic for En→De, or with a publicly available toolkit for the other language pairs [Shuyo 2010]. In order to limit the memory use during training, we only trained the

Language pair	BLEU-c		BLEU-c ranking		Human ranking
	single	ensemble	constrained	unconstrained	
En→Cs	15.7	18.3	1/6	2/7	4/8
En→De	22.4	24.8	1/11	1/13	1-2/16
Cs→En	20.2	23.3	3/6	3/6	3-4/7
De→En	25.6	27.6	6/9	6/10	6-7/13
Fi→En	10.1	13.6	7/9	9/12	10/14

Table 3.2: Results on the official WMT'15 test sets for single models and primary ensemble submissions. All our own systems are constrained. When ranking by BLEU, we only count one system from each submitter. Human rankings include all primary and online systems, but exclude those used in the Cs↔En tuning task.

systems with sentences of length up to 50 words only. Finally, for some but not all models, we reshuffled the data a few times and concatenated the different segments before training.

In the case of German (De) source, we performed compound splitting [Koehn and Knight 2003], as implemented in the Moses toolkit [Koehn et al. 2007]. For Finnish (Fi), we used Morfessor 2.0 for morpheme segmentation [Virpioja et al. 2013] by using the default parameters.

An Issue with Apostrophes In the training data, apostrophes appear in many forms, such as a straight vertical line (U+0027) or as a right single quotation mark (U+0019). The use of, for instance, the normalize-punctuation script[2] could have helped, but we did not use it in our experiments. Consequently, we encountered an issue of the tokenizer from the Moses toolkit not applying the same rule for both kinds of apostrophes. We fixed this issue in time for Czech→English (Cs→En), but all the other systems were affected to some degree, in particular, the system for De→En.

3.3.2 Settings

We used the RNNsearch models of size identical to those presented in [Bahdanau et al. 2015; Jean et al. 2015a]. More specifically, all the words in both target and source vocabularies were projected into a 620-dimensional vector space. Each recurrent neural network (RNN) had a 1000-

[2] http://www.statmt.org/wmt11/normalize-punctuation.perl

dimensional hidden state. The models were trained with Adadelta [Zeiler 2012], and the norm of the gradient at each update was rescaled [Pascanu et al. 2013]. For the language pairs other than Cs→En and Fi→En, we held the word embeddings fixed near the end of training, as described in [Jean et al. 2015a].

With the very large target vocabulary technique in Sec. 3.2.3, we used 500K source and target words for the En→De system, while 200K source and target words were used for the De→En and Cs↔En systems.[3] During training we set τ between 15K and 50K, depending on the hardware availability. As for decoding, we mostly used $K = 30,000$ and $K' = 10$.

Given the small sizes of the Fi→En corpora, we simply used a fixed vocabulary size of 40K tokens to avoid any adverse effect of including every unique target word in the vocabulary. The inclusion of every unique word would prevent the network from decoding out ⟨UNK⟩ at all, even if out-of-vocabulary words will assuredly appear in the test set.

For each language pair, we trained a total of four independent models that differed in parameter initialization and data shuffling, monitoring the training progress on either *newstest2012+2013*, *newstest2013* or *newsdevs2015*.[4] Translations were generated by beam search, with a beam width of 20, trying to find the sentence with the highest log-probability (single model), or highest average log-probability over all models (ensemble), divided by the sentence length [Boulanger-Lewandowski et al. 2013]. This length normalization addresses the tendency of the recurrent neural network to output shorter sentences.

For Fi→En, we augmented models by *deep fusion* with an RNN-LM. The RNN-LM, which was built using the LSTM units, was trained on the English Gigaword corpus using the vocabulary comprising of the 42K most frequent words in the English side of the intersection of the parallel corpora of Fi→En, De→En and Cs→En. Importantly, we use the same RNN-LM for both Fi→En, Cs→En and De→En. In the experiments with deep fusion, we used the randomly selected 2/3

[3] This choice was made mainly to cope with the limited storage availability.
[4] For En→De, we created eight semi-independent models. See [Jean et al. 2015a] for more details.

of *newsdev2015* as a validation set and the rest as a held-out set. In the case of De→En, we used *newstest*2013 for validation and *newstest*2014 for test.

For all language pairs except Fi→En, we also simply built 5-gram language models, this time on all appropriate provided data, with the exception of the English Gigaword [Heafield 2011]. In our contrastive submissions only, we re-ranked our 20-best lists with the LM log-probabilities, once again divided by sentence length. The relative weight of the language model was manually chosen to maximize BLEU on the development set.

3.4 Results

Results for single systems and primary ensemble submissions are presented in Table 3.2.[5] When translating from English to another language, neural machine translation works particularly well, achieving the best BLEU-c scores among all the constrained systems. On the other hand, NMT is generally competitive even in the case of translating to English, but it not yet as good as well as the best SMT systems according to BLEU. If we rather rely on human judgement instead of automated metrics, the NMT systems still perform quite well over many language pairs, although they are in some instances surpassed by other statistical systems that have slightly lower BLEU scores.

In our contrastive submissions for Cs↔En and De↔En where we re-ranked 20-best lists with a 5-gram language model, BLEU scores went up modestly by 0.1 to 0.5 BLEU, but interestingly translation error rate (TER) always worsened. One possible drawback about the manner we integrated language models here is the lack of translation models in the reverse direction, meaning we do not implicitely leverage the Bayes' rule as most other translation systems do.

In our further experiments, which are not part of our WMT'15 submission, for single models we observed the improvements of approximately 1.0/0.5 BLEU points for *dev/test* in {Cs,De}→En

[5] Also available at http://matrix.statmt.org/matrix/

tasks, when we employ *deep fusion* for incorporating language models.[6]

3.5 Conclusion

We presented our neural machine translation (NMT) systems for WMT'15, using the encoder–decoder model with the attention mechanism [Bahdanau et al. 2015] and the recent developments in NMT [Jean et al. 2015a; Gulcehre et al. 2015]. We observed that the NMT systems are now competitive against the conventional SMT systems, ranking first by BLEU among the constrained submission on both the En→Cs and En→De tasks. In the future, more analysis is needed on the influence of the source and target languages for neural machine translation. For instance, it would be interesting to better understand why performance relative to other approaches was somewhat weaker when translating into English, or how the amount of reordering influences the translation quality of neural MT systems.

3.6 Since the release of this chapter

Neural machine translation has become the *de facto* standard approach, encompassing most of the top-performing systems at WMT competitions in the following years. A significant research effort has been undertaken, and is still taking place, to further advance neural machine translation. The most significant advance is likely the introduction of the Transformer model [Vaswani et al. 2017], which strongly relies on self-attention, cross-attention and feed-forward transformations to produce high-quality translations. It has been argued that neural machine translation reached human parity under some circumstances [Hassan et al. 2018], although a closer inspection revealed weaknesses of these NMT systems when evaluated at the document-level [Läubli et al. 2018].

[6] Improvements are for single models only. See [Gulcehre et al. 2015] for more details.

4 First steps towards translation in context

In this chapter, we propose neural machine translation architectures that model the surrounding text in addition to the source sentence. These models lead to better performance, both in terms of general translation quality and pronoun prediction, under some data conditions.

In particular, we present our systems for the DiscoMT 2017 cross-lingual pronoun prediction shared task. Attention-based neural machine translation is well suited for pronoun prediction and compares favorably with other approaches that were specifically designed for this task. We also analyze the behaviour of our systems, as well as other contemporaneous larger-context translation models.

4.1 Introduction

A major strength of neural machine translation, which has recently become *de facto* standard in machine translation research, is the capability of seamlessly integrating information from multiple sources. Due to the nature of continuous representation used within a neural machine translation system, any information, in addition to tokens from source and target sentences, can be integrated as long as such information can be projected into a vector space. This has allowed researchers to build a non-standard translation system, such as multilingual neural translation

systems [see, e.g., Firat et al. 2016; Zoph and Knight 2016], multimodal translation systems [see, e.g., Caglayan et al. 2016; Specia et al. 2016] and syntax-aware neural translation systems [see, e.g., Nadejde et al. 2017; Eriguchi et al. 2016, 2017]. At the core of all these recent extensions is the idea of using context larger than a current source sentence to facilitate the process of translation.

In this chapter, we investigate the potential for implicitly incorporating discourse-level structure into neural machine translation. As an initial attempt, we focus on incorporating a small number of preceding and/or following source sentences into the attention-based neural machine translation model [Bahdanau et al. 2015]. More specifically, instead of modelling the conditional distribution $p(Y|X)$ over translations given a source sentence, we build a network that models the conditional distribution $p(Y|X, X_{-n}, \ldots, X_{-1}, X_1, \ldots, X_n)$, where X_{-i} is the i-th preceding source sentence, and X_i the i-th following source sentence. We propose a novel larger-context neural machine translation model based on the recent works on larger-context language modelling [Wang and Cho 2016] and multi-way, multilingual neural machine translation [Firat et al. 2016].

We first evaluate the proposed model against the baseline model without any context other than a source sentence using BLEU and RIBES [Isozaki et al. 2010], both of which measure translation quality *averaged* over all the sentences in a corpus. This evaluation strategy reveals that the benefit of larger context is not always apparent when the evaluation metric is average translation quality, confirming the earlier observation, for instance, by Hardmeier et al. [2015]. Then, we turn to a more focused evaluation based on pronoun prediction [Guillou et al. 2016] which was a shared task at WMT'16. On this cross-lingual pronoun prediction task, we notice benefits from incorporating larger context when training models on small corpora, but not on larger ones. Interestingly, we also observe that neural machine translation can predict pronouns as well as the top ranking approaches from the shared task at WMT'16.

We then look at additional network architecture variants and evaluate these models on the DiscoMT 2017 cross-lingual pronoun prediction shared task [Loáiciga et al. 2017]. We consider four language pairs: En-Fr, En-De, De-En and Es-En. We also examine the attention patterns of

some of these models, as well as other approaches proposed at that time.

4.2 Larger-Context Neural Machine Translation

4.2.1 Attention-based Neural Machine Translation

Attention-based neural machine translation, proposed by Bahdanau et al. [2015], has become *de facto* standard in recent years, both in academia [Bojar et al. 2016] and industry [Wu et al. 2016; Crego et al. 2016]. An attention-based translation system consists of three components; (1) encoder, (2) decoder and (3) attention model. The encoder is often a bidirectional recurrent network with a gated recurrent unit [GRU, Cho et al. 2014; Hochreiter and Schmidhuber 1997], which encodes a source sentence $X = (x_1, x_2, \ldots, x_{T_x})$ into a set of annotation vectors $\{h_1, h_2, \ldots, h_{T_x}\}$, where $h_t = \left[\overrightarrow{h}_t; \overleftarrow{h}_t\right]$. $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ are the t-th hidden states from the forward and reverse recurrent networks respectively.

The decoder is a recurrent language model [Mikolov et al. 2010; Graves 2013] which generates one target symbol $y_{t'}$ at a time by first computing the attention scores $\{\alpha_{t,t'}\}_{t=1}^{T_x}$ over the annotation vectors. Each attention score is computed by

$$\alpha_{t,t'} \propto \exp(f_{\text{att}}(\hat{y}_{t'-1}, z_{t'-1}, h_t)),$$

where f_{att} is the attention model implemented as a feedforward network taking as input the previous target symbol $\hat{y}_{t'-1}$, the previous decoder hidden state $z_{t'-1}$ and one of the annotation vector h_t. These attention scores are used to compute the time-dependent source vector $s_{t'} = \sum_{t=1}^{T_x} \alpha_{t,t'} h_t$, based on which the decoder's hidden state and the output distribution over all possible target symbols are computed:

$$p(y_{t'}|y_{<t'}, X) \propto \exp(g_{\text{out}}^{y_{t'}}(z_{t'})),$$

where

$$z_{t'} = \phi(\hat{y}_{t'-1}, z_{t'-1}, s_{t'}). \tag{4.1}$$

ϕ is a recurrent activation function such as a GRU or long short-term memory (LSTM) unit.

The whole model, consisting of the encoder, decoder and attention model, is fully differentiable, and can be jointly trained by maximizing the log-likelihood given a training corpus using stochastic gradient descent with backpropagation-through-time [Werbos 1990].

4.2.2 Larger-Context Neural Machine Translation

We extend the attention-based neural machine translation described above by including an additional set of an encoder and attention model. This additional encoder is similarly a bidirectional recurrent network, and it encodes a context sentence, in our case a source sentence immediately before the current source sentence,[1] into a set of *context annotation vectors* $\left\{h^c_1, \ldots, h^c_{T_c}\right\}$, where $h^c_t = \left[\overrightarrow{h}^c_t; \overleftarrow{h}^c_t\right]$. Similarly to the original source encoder, these two vectors are from the forward and reverse recurrent networks.

On the other hand, the additional attention model is different from the original one. The goal of incorporating larger context into translation is to provide additional discourse-level information necessary for translating a given source token, or a phrase. This implies that the attention over, or selection of, tokens from larger context be done with respect to which source token, or phrase, is being considered. We thus propose to make this attention model take as input the previous target symbol, the previous decoder hidden state, a context annotation vector as well as the

[1] Although we use a single preceding sentence in this paper, the proposed method can easily handle multiple preceding and/or following sentences either by having multiple sets of encoder and attention mechanism or by concatenating all the context sentences into a long single sequence.

source vector from the main attention model. That is,

$$\alpha^c_{t,t'} \propto \exp(f^c_{\text{att}}(\hat{y}_{t'-1}, z_{t'-1}, h^c_t, s_{t'})).$$

Similarly to the source vector, we compute the time-dependent *context vector* as the weight sum of the context annotation vectors: $c_{t'} = \sum_{t=1}^{T_c} \alpha^c_{t,t'} h^c_t$.

Now that there are two vectors from both the current source sentence and the context sentence, the decoder transition in Eq. (4.1) changes accordingly:

$$z_{t'} = \phi(\hat{y}_{t'-1}, z_{t'-1}, s_{t'}, c_{t'}). \tag{4.2}$$

We later refer to this approach as larger-context neural machine translation (LC-NMT) or as the *simple context model (SCM)*.

4.2.3 Variants for DiscoMT'17

For our submission to the DiscoMT'17 shared task, we additionally consider two architectural variants.

4.2.3.1 Double-Gated Context Model (DGCM)

Our second approach is very similar to the first with the exception that, for both functions f and g, distinct gates (g_1 and g_2) are applied to the context representation c^c_i. Similar context-modulating gates were previously used by [Wang et al. 2017].

$$s_i = f(s_{i-1}, y_{i-1}, c_i, g_1 \odot c^c_i) \tag{4.3}$$

$$p(y_i|y_1, ..., y_{i-1}, \mathbf{x}, \mathbf{x}^c) = g(y_{i-1}, s_i, c_i, g_2 \odot c_i^c) \tag{4.4}$$

Each gate has its own set of parameters and depends on the previous target symbol, the current source representation and the decoder hidden state, at time $i-1$ for g_1 and i for g_2.

4.2.3.2 Combined Context Model (CCM)

The last method first combines the source and context representations into a vector d_i through a multi-layer perceptron. As in the second approach, the context is also gated.

$$d_i = \mathbf{W}_3\Big(\tanh(\mathbf{W}_1 c_i + \mathbf{W}_2(g_1 \odot c_i^c))\Big) \tag{4.5}$$

$$s_i = f(s_{i-1}, y_{i-1}, d_i) \tag{4.6}$$

$$p(y_i|y_1, ..., y_{i-1}, \mathbf{x}, \mathbf{x}^c) = g(y_{i-1}, s_i, d_i) \tag{4.7}$$

4.2.4 Other contemporaneous approaches

As we introduced larger-context neural machine translation, other approaches were also developed simultaneously or soon after.

4.2.4.1 2+1

The 2+1 (break) model uses exactly the same architecture as the baseline, but modifies the training data [Tiedemann and Scherrer 2017]. It concatenates the previous sentence to the source sentence to be translated, with a special token between the two segments. This demarcation lets the model know what part of the input must be translated, while still allowing it to attend to some context words if useful. Note that the single attention system is shared between the context and source.

4.2.4.2 DCU LC-NMT

The DCU LC-NMT model uses a global context representation instead of attending specific words, which lets it incorporate a longer context [Wang et al. 2017]. This representation is obtained by means of a hierarchical RNN, which first encodes the K preceding sentences (here 3), after which it summarizes them into one document representation D.

This representation D is used in multiple ways. First, it helps initializing the encoder and decoder hidden states. Furthermore, at every timestep t', a gated context representation $g_{t'} \odot D$ is employed as an additional input to the decoder RNN.

4.3 Evaluating Larger-Context Neural Machine Translation

A standard metric for automatically evaluating the translation quality of a machine translation system is BLEU [Papineni et al. 2002]. BLEU is computed on a validation or test corpus by inspecting the overlap of n-grams (often up to 4-grams) between the reference and generated corpora. BLEU has become *de facto* standard after it has been found to correlate well with human judgement for phrase-based and neural machine translation systems. Other metrics, such as METEOR [Denkowski and Lavie 2014], TER [Snover et al. 2006] and RIBES [Isozaki et al. 2010], are

often used together with BLEU, and they also measure the *average translation quality* of a machine translation system over an entire validation or test corpus.

It is not well-known how much positive or negative effect larger context has on machine translation. It is understood that larger context allows a machine translation system to capture properties not apparent from a single source sentence, such as style, genre, topical patterns, discourse coherence and anaphora [see, e.g., the preface of Webber et al. 2015], but the degree of its impact on the average translation quality is unknown.

It is rather agreed that the impact should be measured by a metric specifically designed to evaluate a specific effect of larger context. For instance, discourse coherence has been used as one of such metrics in analyzing larger-context language modelling in recent years [Ji et al. 2015, 2016]. In the context of machine translation, cross-lingual pronoun prediction [Hardmeier et al. 2015; Guillou et al. 2016] has been one of the few established tasks by which the effect of larger-context modelling, or the ability of a machine translation system for incorporating larger-context information, is evaluated.

We therefore compare the baseline sentence-level neural machine translation model against the proposed larger-context model based on both the average translation quality, measured by BLEU, and the pronoun prediction accuracy, measured in macro-averaged recall.

Unlike the existing approaches to cross-lingual pronoun prediction, we do not train any of the models specifically for the pronoun prediction task, but train them to maximize the average translation quality. Once the model is trained, we conduct pronoun prediction by

$$\hat{y} = \arg\max_{y \in P} \log p(y^*_{<n}, y, y^*_{>n}|X), \tag{4.8}$$

where P is the set of all possible pronouns,[2] and the goal is to predict the pronoun in the n-th position in the target sentence.

[2]In addition all possible pronouns, there is a class designated for any non-pronoun token.

4.4 Initial experiments

4.4.1 Data and Tasks

We use En-Fr and En-De data for our initial experiments, as provided by the WMT'16 cross-lingual pronoun prediction shared task organizers.[3] The target side of the parallel corpus for each language pair has been heavily preprocessed, including tokenization and lemmatization. Although both of the corpora come with part-of-speech (POS) tags, we do not use them. In the case of En-Fr, the set P of all pronouns includes "ce", "elle", "elles", "il", "ils", "cela", "on" and OTHER. The pronoun set consists of "er", "sie", "es", "man" and OTHER in the case of En-De. Macro-average recall is used as a main evaluation metric. There are 2,441,410 and 2,356,313 sentence pairs in the En-Fr and En-De training corpora, respectively.

For pronoun prediction, the input to the model is a source sentence and the corresponding target sentence of which some pronouns are replaced with a special token REPLACE. The goal is then to figure out which pronoun should replaced the REPLACE token, and this is done by finding a combination that maximizes the log-probability, as in Eq. (4.8). When there are multiple REPLACE tokens in a single example, we exhaustively try all possible combinations, which is feasible as the size of the pronoun set P is small.

For translation, the input to the model is a source sentence alone, and the model is expected to generate a translation. We use beam search to approximately find the maximum-a-posterior translation, i.e, $\arg\max_Y \log p(Y|X)$.

In addition to the data/tasks from the cross-lingual pronoun prediction shared task, we also check the average translation quality using IWSLT'15 En-De as the training set. We use the IWSLT'12 and IWSLT'14 test set for development and test respectively. This is to ensure that our observation from the earlier lemmatized corpora transfers to non-lemmatized ones. This corpus

[3] http://data.statmt.org/wmt16/pronoun-task/

	5%	10%	20%	40%	100%
			En-Fr		
NMT	27.6	32.7	35.7	38.2	39.9
LC-NMT	28.8	33.9	36.7	38.6	39.0
			En-De		
NMT	16.3	19.8	22.1	24.3	25.6
LC-NMT	17.4	20.9	22.7	23.9	25.1

(a) BLEU

	5%	10%	20%	40%	100%
			En-Fr		
NMT	82.0	84.0	85.0	85.9	86.9
LC-NMT	82.4	84.8	85.6	86.0	86.4
			En-De		
NMT	76.6	78.9	80.4	81.4	81.7
LC-NMT	77.3	79.5	80.6	81.5	81.7

(b) RIBES

Table 4.1: Translation quality in (a) BLEU and (b) RIBES on the cross-lingual pronoun prediction corpora

5%	10%	20%	40%	100%	100%
		En-Fr			
49.7	54.1	57.6	64.2	67.6	65.7*
50.7	54.0	60.4	64.2	59.2	65.35°
		En-De			
44.6	44.1	44.9	50.2	56.4	64.6*
54.2	46.3	44.8	52.3	51.1	52.5$^\bullet$

Table 4.2: Macro-average recall for cross-lingual pronoun prediction. We display two top rankers from the shared task in the last column. ($\star$) [Luotolahti et al. 2016] ($\circ$) [Stymne 2016] ($\bullet$) [Dabre et al. 2016]

has 194,371 sentence pairs for training, and 1700 and 1305 for development and test.

4.4.2 Models and Learning

Baseline Model (NMT) We train a baseline attention-based neural machine translation system based on the code publicly available online.[4] The dimensionalities of word vectors, encoder recurrent network and decoder recurrent network are 620, 1000 and 1000, respectively. We use a one-layer feedforward network with one tanh hidden units as an attention model. We regularize the models with Dropout[Pham et al. 2014].

[4] https://github.com/nyu-dl/dl4mt-tutorial/

	BLEU	RIBES
NMT	19.7	77.8
LC-NMT	20.7	79.0

Table 4.3: Translation quality on IWSLT (En-De).

Larger-Context Model (LC-NMT) A larger-context model closely follows the configuration of the baseline model. The additional encoder has two GRU's, and thus outputs a 2000-dimensional time-dependent context vector each time.

Learning We train both types of models to maximize the log-likelihood given a training corpus using Adadelta [Zeiler 2012]. We early-stop with BLEU on a validation set.[5] We do not do anything particular for the cross-lingual pronoun prediction task.

Varying training corpus sizes We experiment by varying the size of the training corpus to see if there is any meaningful difference in performance between the vanilla and larger-context models w.r.t. the size of training set. We do it for the corpora from the pronoun prediction task, using 5%, 10%, 20%, 40% and 100% of the original training set.

4.4.3 Results

From the results presented in Table 4.1 and 4.2, we observe that the larger-context models generally outperform the sentence-level ones in terms of BLEU, RIBES and macro-average recall. However, this improvement vanishes as the size of training set grows. We confirm that this is not due to the lemmatization of the target side of the pronoun task corpora by observing that the proposed larger-context model also outperforms the baseline on IWSLT En-De, of which the training corpus size is approximately 10% of the full pronoun task corpus, as shown in Table 4.3.

4.5 DiscoMT'17

The DiscoMT 2017 pronoun prediction task serves as a platform to improve pronoun prediction and shares many similarities with the previous WMT'16 shared task. We are provided source documents and their lemmatized translations for four language pairs: En-Fr, En-De, De-En and

[5] We use greedy decoding for early-stopping.

Es-En. In each translation, some sentences have one or more pronouns substituted by the placeholder "REPLACE". For each of these tokens, we must select the correct pronoun among a small set of candidates.

There are respectively 8, 5, 9 and 7 target classes for En-Fr, En-De, De-En and Es-En. For example, in the case of En-Fr, the task is concentrated on the translation of "it" and "they". The possible target classes are:

- **ce, elle, elles, il, ils, cela, on, OTHER.**

Although only a subset of the data has context dependencies, it is not difficult to find such instances. The following set of sentences taken from the En-Fr development data is a good example:

- **Context:** *So the idea is that accurate perceptions are fitter perceptions .*
- **Source:** *They give you a survival advantage .*

And here are the source sentence translation with the missing token and the corresponding target:

- **Translation:** *REPLACE vous donner un avantage en terme de survie .*
- **Target:** elles

In this example, "REPLACE" should be the translation of the word "They", which refers to "perceptions" in the previous sentence. This is important because in French, "perceptions" is feminine. Correctly choosing a good pronoun here can only be done confidently with contextual information.

We use similar experimental settings as for the initial experiments. To account for sentences with multiples pronoun to predict, we use a modified beam search where the beam is expanded

	Baseline	SCM	DGCM	CCM
En-Fr	67.9	66.2	**68.9**	64.5
En-De	58.2	57.1	**59.0**	57.6
De-En	70.9	70.3	72.4	**72.8**
Es-En	69.9	**77.1**	70.8	72.3

Table 4.4: Validation macro-average recall (in %) for cross-lingual pronoun prediction.

	Baseline	SCM	DGCM	CCM	Best
En-Fr	58.1	52.2	62.3	52.1	66.9
En-De	60.9	63.2	61.3	59.5	78.4
De-En	63.3	63.8	64.8	65.5	69.2
Es-En	58.9	56.1	58.7	56.4	58.9

Table 4.5: Test macro-average recall (in %) for cross-lingual pronoun prediction. The "Best" column displays the highest score across all primary and contrastive submissions to the DiscoMT 2017 shared task [Loáiciga et al. 2017].

only at the "REPLACE" placeholders, and is otherwise constrained to the reference. The beam size is set to the number of pronoun classes, so that our approach is equivalent to exhaustive search for sentences with a single placeholder. Models for which beam search lead to the highest validation macro-average recall were selected and submitted for the shared task. The baselines were also sent as contrastive submissions.

4.5.1 Results

Table 4.4 and 4.5 respectively present validation and test results across all language pairs for a few architectural variants. Among the four models we evaluated on the test sets, a different one performs best for each language pair. Nevertheless, the DGCM model is the most consistent, always ranking second or first among our systems. Moreover, it beats the baseline on all tasks except Es-En, which it trails by a marginal 0.2%.

Our models, which don't leverage the given part-of-speech tags and external alignments, are generally competitive with the best submissions [Loáiciga et al. 2017]. For Es-En, our contrastive submission achieves the best performance. As for En-Fr and De-En, our systems obtain a macro-

	valid	test
Baseline	67.9	58.1
DCU LC-NMT	69.4	53.7
2+1	71.6	59.1
DGCM	68.9	62.3

Table 4.6: Macro-average recall (in %) for cross-lingual pronoun prediction.

average recall within 5% of the winning systems. Finally, the relatively poor performance of our models for En-De is due to their incapacity at correctly predicting the rare pronoun 'er'. Indeed, the recall of 0/8 for that class greatly affects the results.

4.6 Comparison to contemporaneous approaches

We now compare the double-gated context model (DGCM) [Jean et al. 2016], the 2+1 (break) approach [Tiedemann and Scherrer 2017] and the DCU LC-NMT system [Wang et al. 2017]. For these experiments, we use the En-Fr dataset from the DiscoMT 2017 cross-lingual pronoun prediction task. We are provided source documents and their translations, for a total of 2,441,410 sentence pairs. The target side of the parallel corpus has been heavily preprocessed, including tokenization and lemmatization. This prevents models to infer the correct pronoun by relying on number, gender or person information within the target sentence, making the task more challenging.

4.6.1 Results

Systems were trained until convergence using a validation set with macro-average recall for model selection and avoiding overfitting. While we trained the 2+1 and DCU models for more than a week, some marginal improvements may still be obtainable. Nevertheless, we certainly don't expect the qualitative behaviour of these models to vary in any significant manner.

The pronoun prediction task is evaluated with macro-average recall (in %). Table 4.6 presents

	valid	test
Baseline	40.3	35.5
DCU LC-NMT	40.3	35.9
2+1	40.8	36.1
DGCM	40.2	35.6

Table 4.7: BLEU score for pronoun prediction dataset.

Pronouns	ce	elle	elles	il	ils	cela	on	OTHER
Total	0	6	8	5	23	1	0	0
Baseline	-	4	0	1	22	1	-	-
DGCM	-	2	3	4	22	1	-	-
2+1	-	1	2	3	22	1	-	-
DCU LC-NMT	-	3	2	4	21	1	-	-

Table 4.8: Pronoun prediction performance on context dependent examples.

results across four models. We observe that there is no major difference between all four.

As all four models are also complete translation systems, rather than being only able to predict pronouns in isolation, we may also evaluate their general quality with BLEU (Table 4.7). Again, we can observe that there is no major difference between all four systems.

As we would have expected a larger improvement by incorporating context into neural translation systems, we need to analyze how models behave when the previous sentences are necessary for a confident translation. By traversing the validation set manually, we were only able to find 43 pronouns whose antecedent appeared exclusively in the previous sentence, spread over 36 sentences. The results for those 43 specific examples are presented in Table 4.8. The high recall of the target "ils" can be explained by the fact that it appears frequently in the training data, driving the models to assign it a high probability.

4.6.2 Analysis

We will now present some informative examples in more detail, starting the analysis with the one originally introduced in subsection 4.5. In particular, we are interested in the attention patterns, although we note that attention is not necessarily explanation [Jain and Wallace 2019].

In particular, the representation at the same position as a given word still integrates information from the entire sequence.

Example 1:

- **Context:** *So the idea is that accurate perceptions are fitter perceptions .*
- **Source:** *They give you a survival advantage .*
- **Translation:** *REPLACE vous donner un avantage en terme de survie .*
- **Target:** elles

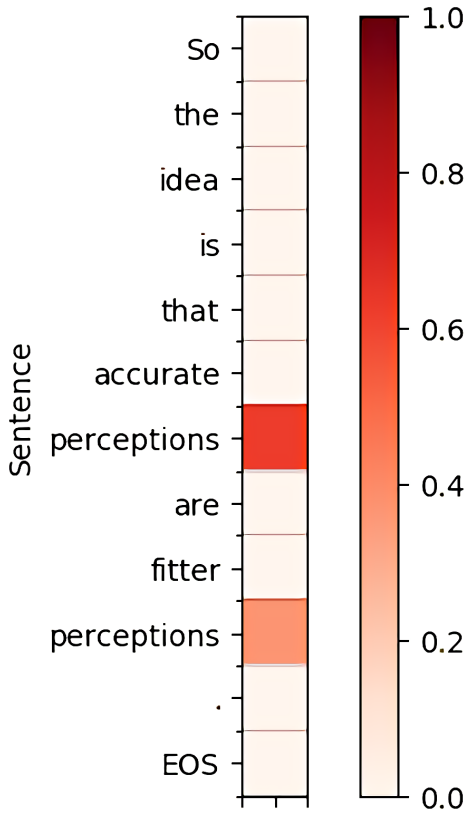

Figure 4.1: DGCM attention over context for example 1. The probability distribution of the attention mechanism is represented by color intensity.

As previously mentioned in section 4.5, "REPLACE" refers to the contextual words "perceptions", whose translation is feminine. It is promising to see that the DGCM indeed concentrates on these words, as depicted in figure 4.1.

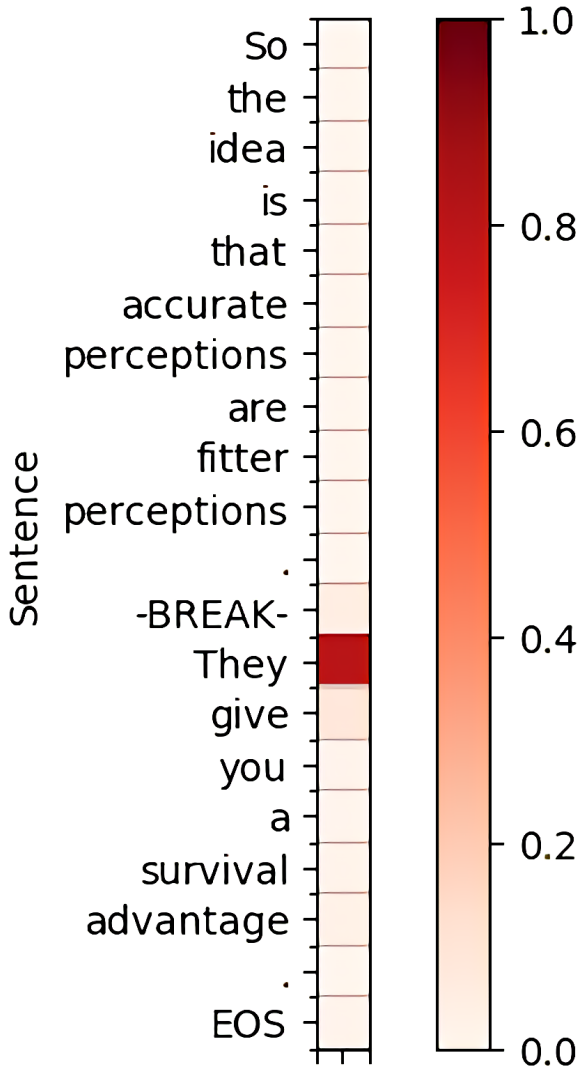

Figure 4.2: 2+1 attention over context and the source for example 1. The probability distribution of the attention mechanism is represented by color intensity.

The 2+1 model also has a attention mechanism over the context sentence, but it is shared with the source. As shown in Figure 4.2, the attention concentrates on the pronoun "They", ignoring

the context sentence almost entirely.

A similar behavior was also observed for all subsequent examples. To avoid being redundant, we will not display the attention probabilities again for the 2+1 model. In general, the context appears to be attended much less than reported by [Tiedemann and Scherrer 2017] on movie subtitles, which is likely due to the greater complexity of our training data, which includes longer sentences.

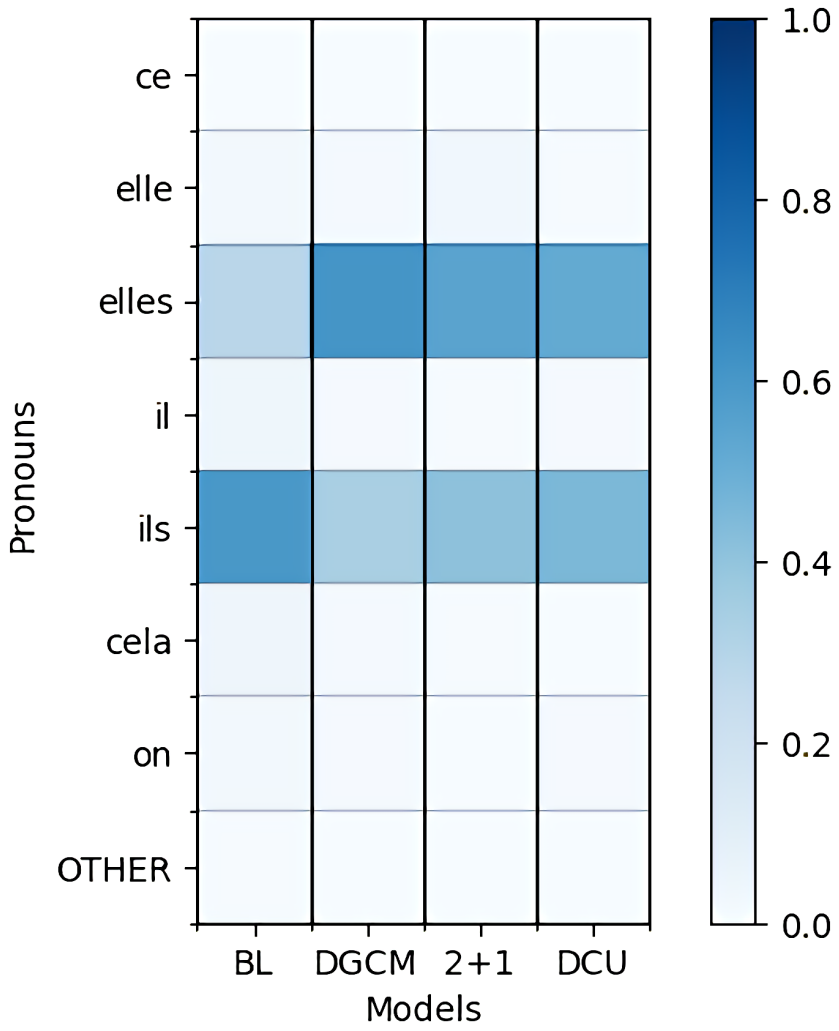

Figure 4.3: Pronoun probabilities for example 1. The probability distribution is represented by color intensity.

In this example, the baseline does not know what "They" refers to, so it predicts "ils", which has the highest context-agnostic probability. However, context clearly affects the prediction of the DGCM, which now correctly outputs "elles". The DCU and 2+1 models aren't as confident in the correct prediction, but still prefer "elles".

We move on to another example [6] where context is clearly useful.

Example 2:

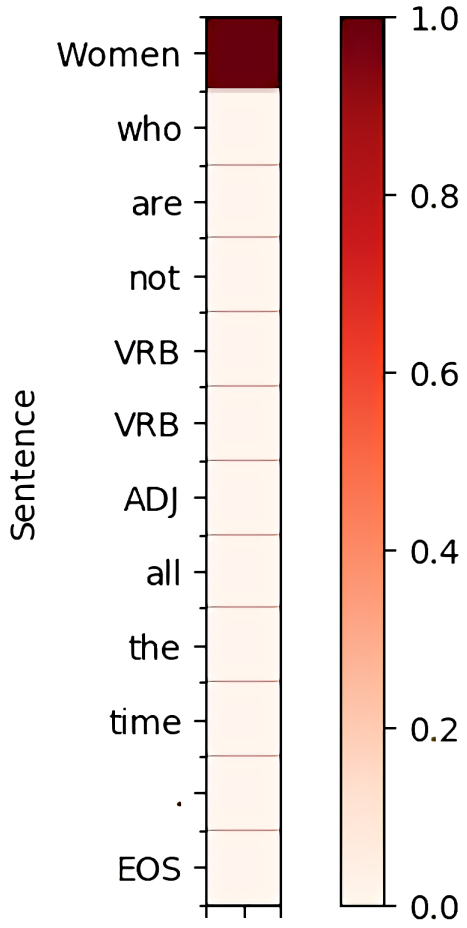

Figure 4.4: DGCM attention over context for example 2.

[6] For this example, we replaced some words that could be considered inappropriate by they part of speech tags.

- **Context:** *Women who are not VRB VRB ADJ all the time .*
- **Source:** *Then they VER ADV PRP everybody .*
- **Translation:** *et alors , REPLACE VER avec ADV VER qui .*
- **Target:** elles

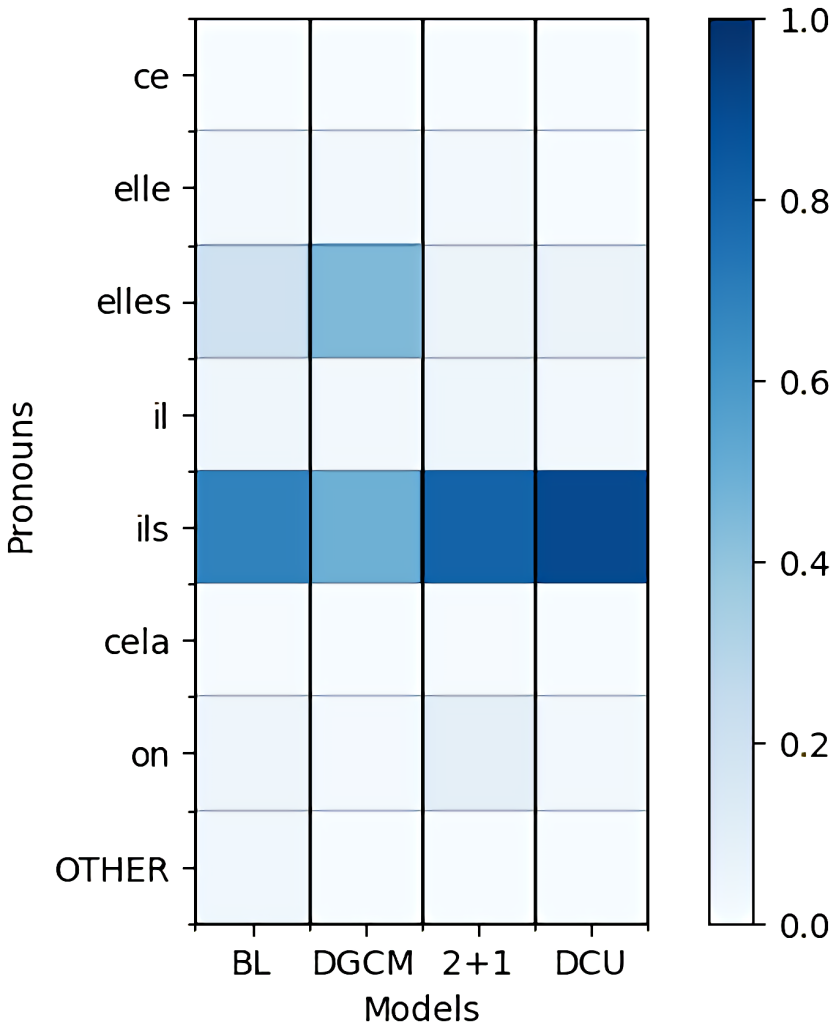

Figure 4.5: Pronoun probabilities for example 2.

This time, the pronoun refers to the word "Women" in the previous sentence. Figure 4.4 shows

that the attention mechanism of the DGCM focuses on the correct noun.

The baseline predicts the pronoun "ils" with a high probability because it cannot know that "they" refers to "Women". The context helps the DGCM to increase the probability of the correct pronoun "elles", but not enough to change its prediction. As for the 2+1 and DCU models, they still predict the wrongly gendered pronoun "ils" with high confidence.

We now consider a challenging example with multiple nouns in the preceding sentence.

Example 3:

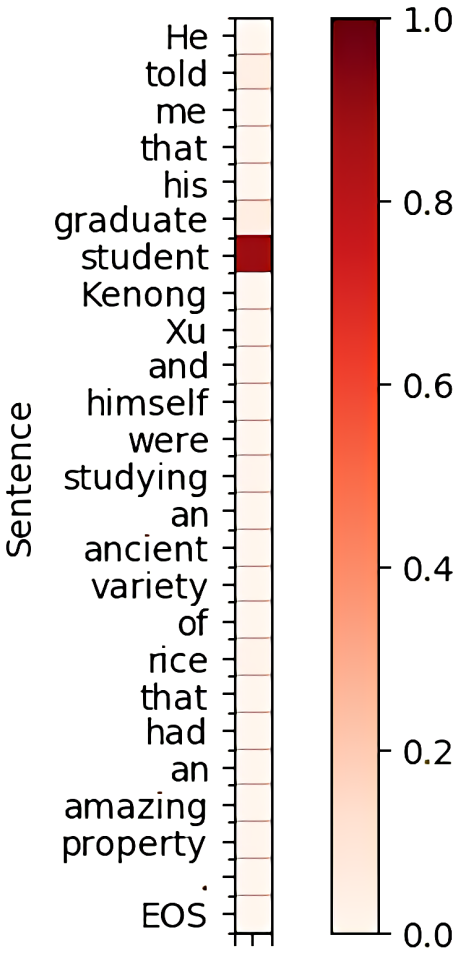

Figure 4.6: DGCM attention over context for example 3.

- **Context:** *He told me that his graduate student Kenong Xu and himself were studying an ancient variety of rice that had an amazing property .*
- **Source:** *It could withstand two weeks of complete submergence .*
- **Translation:** *REPLACE pouvoir supporter deux semaine en immersion complet .*
- **Target:** elle

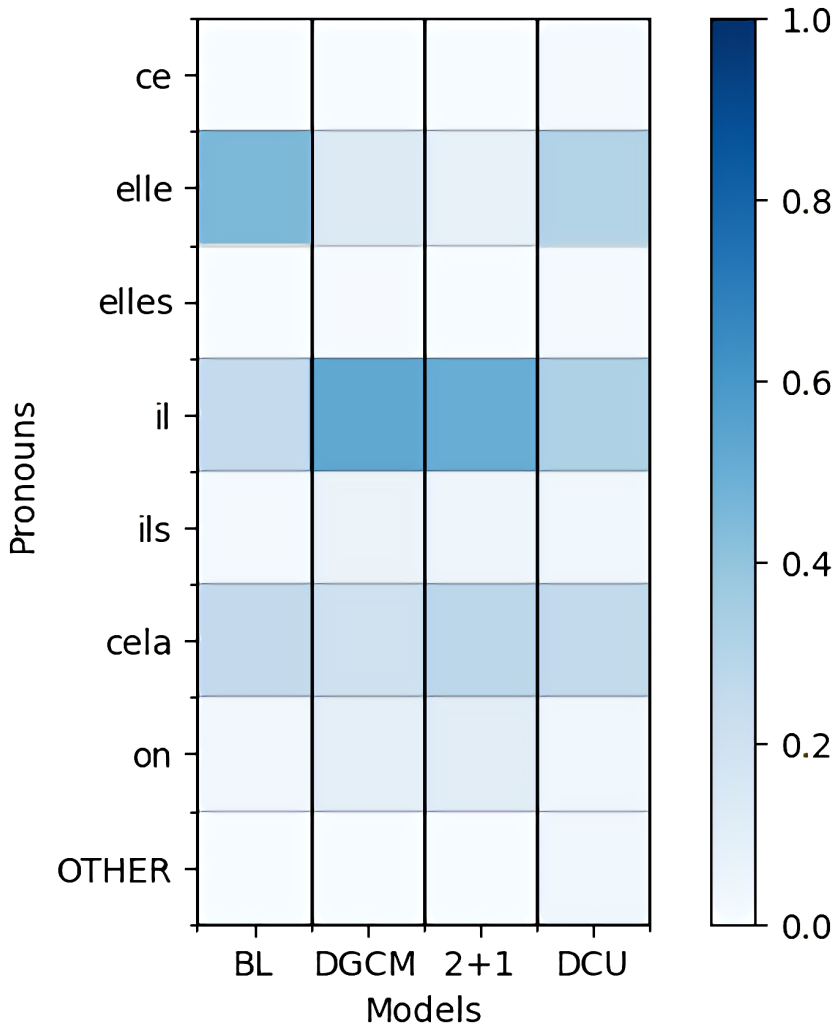

Figure 4.7: Pronoun probabilities for example 3.

In this case, the pronoun refers to the word "variety" in the previous sentence, which is feminine in French. Figure 4.6 shows that the DGCM strongly attends to the wrong noun "student" instead.

For this sentence, the baseline and the DCU model are fairly uncertain and mostly share their probabilities between three words, but manage to choose the correct one (Figure 4.7). The 2+1 model also spreads most of its probability mass over the two or three most likely pronouns, but ultimately makes a mistake. As for the DGCM, it predicts the wrong word, potentially due to its faulty attention.

We finally move on to an example in which context is informative, although the baseline already predicted the correct pronoun.

Example 4:

- **Context:** *You can see the Sub1 variety does great .*
- **Source:** *In fact , it produces three and a half times more grain than the conventional variety .*
- **Translation:** *en faire , REPLACE produire trois foi/fois et demi plus de grain que le variete conventionnel .*
- **Target:** elle

To confidently predict "elle", it is helpful for the model to know that the pronoun to be replaced once again refers to the word "variety" in the previous sentence. Figure 4.8 shows that the DGCM correctly attends to this word.

For this example, all models except 2+1 correctly choose "elle", although the DGCM is slightly more assured in its decision (Figure 4.9). As for the baseline, it could predict "elle" because "it" is compared to the "conventional variety". As such, without context, it is possible (but uncertain) that "it" refers to a "non-conventional variety".

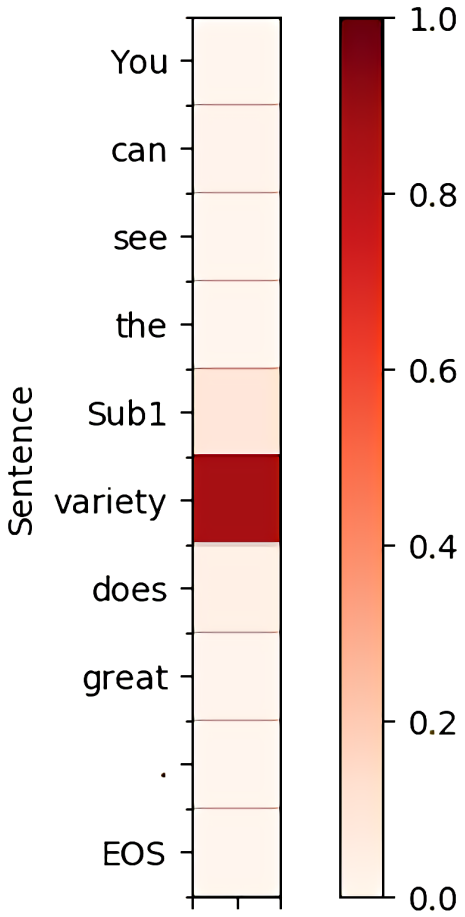

Figure 4.8: DGCM attention over context for example 4.

Full sentence attention Until now, we have only considered pronoun prediction, but it would also be interesting to know how the models behave when translating other words. We first observe how the attention mechanism of DGCM acts. Figure 4.10 represents the focus on the context sentence for example 4 at each decoding time step. Those probabilities are generated using force decoding, meaning that the true target tokens are always fed to the model.

While the attention on "variety" is appropriately at its strongest when translating a pronoun, it remains high for many other target words. Moreover, a few different source tokens, such as "Sub1" and "EOS", are also significantly attended, although this may very well not be useful.

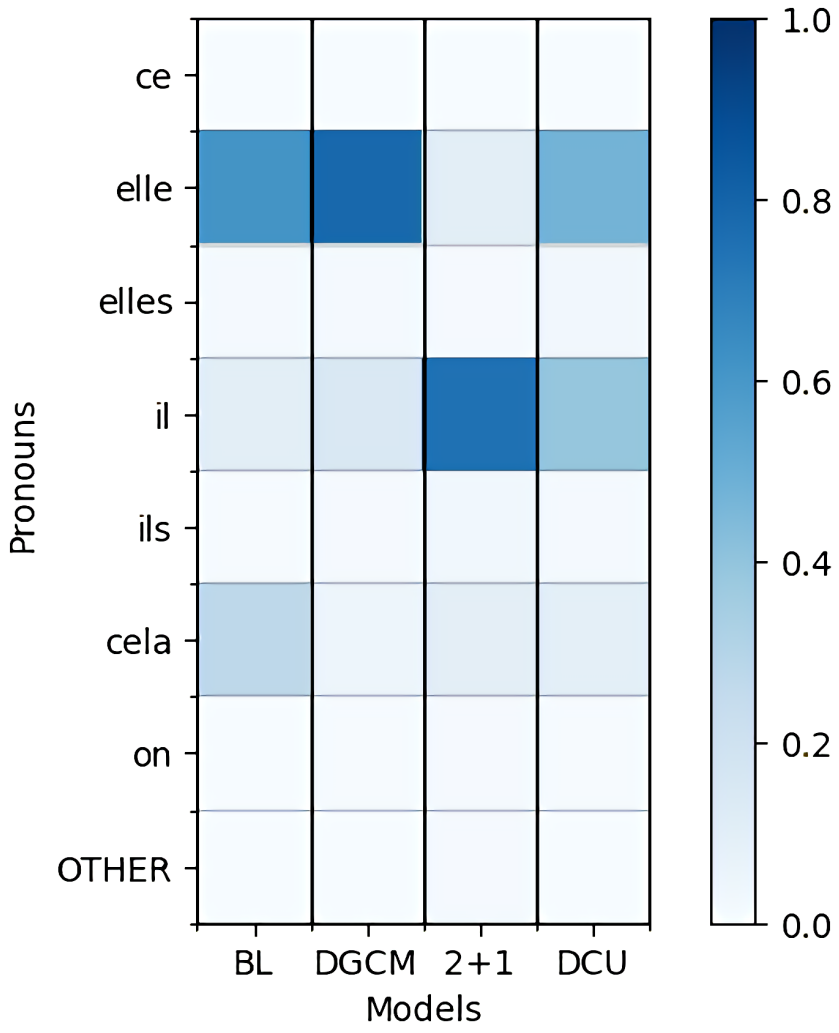

Figure 4.9: Pronoun probabilities for example 4.

GATES As larger context is not always necessary to produce an appropriate translation, both the DGCM and DCU models use gates to modulate this information. For DGCM, Figure 4.11 presents the mean activation of the gates, as well as their standard deviation, over the 36 chosen sentences. Note that early on, the average is over all examples, while the later time steps only take into consideration long enough sentences.

Within the GRU (Figure 4.11, in orange), the average activation is higher at the beginning and stabilizes fairly quickly. As such, it appears that the model learns that context may often be

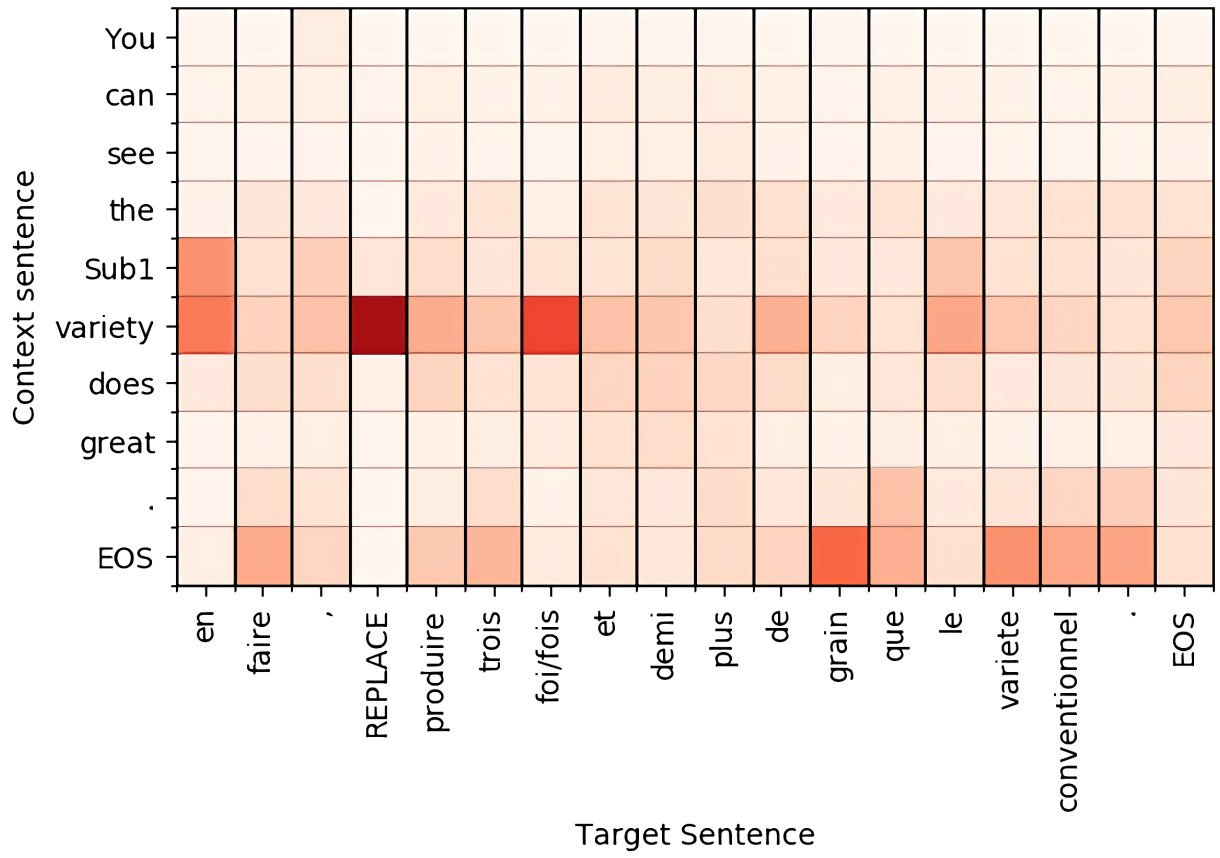

Figure 4.10: DGCM attention over the context for all the words in the target sentence

more relevant early on, or that it relies more on it when there is limited information within the current sentence. However, we did not observe particularly high gates values for pronouns or other words whose translation obviously depends on the context. The trend differs at the output level, where the gates behave similarly at every time step (Figure 4.11, in blue).

Similarly, the DCU LC-NMT model also uses gates to filter the larger-context information. In the paper [Wang et al. 2017], the gates are presented as one vector applied to the larger-context representation. Looking at the code, we realized that separate gates are applied in three distinct parts of the GRU. More specifically, the context is gated when used as input for the reset gates, the update gates and the hidden state proposal (Figure 4.12, in blue, orange and green respectively). Moreover, the gates are inversely tied with the hidden state and source vector representation, so

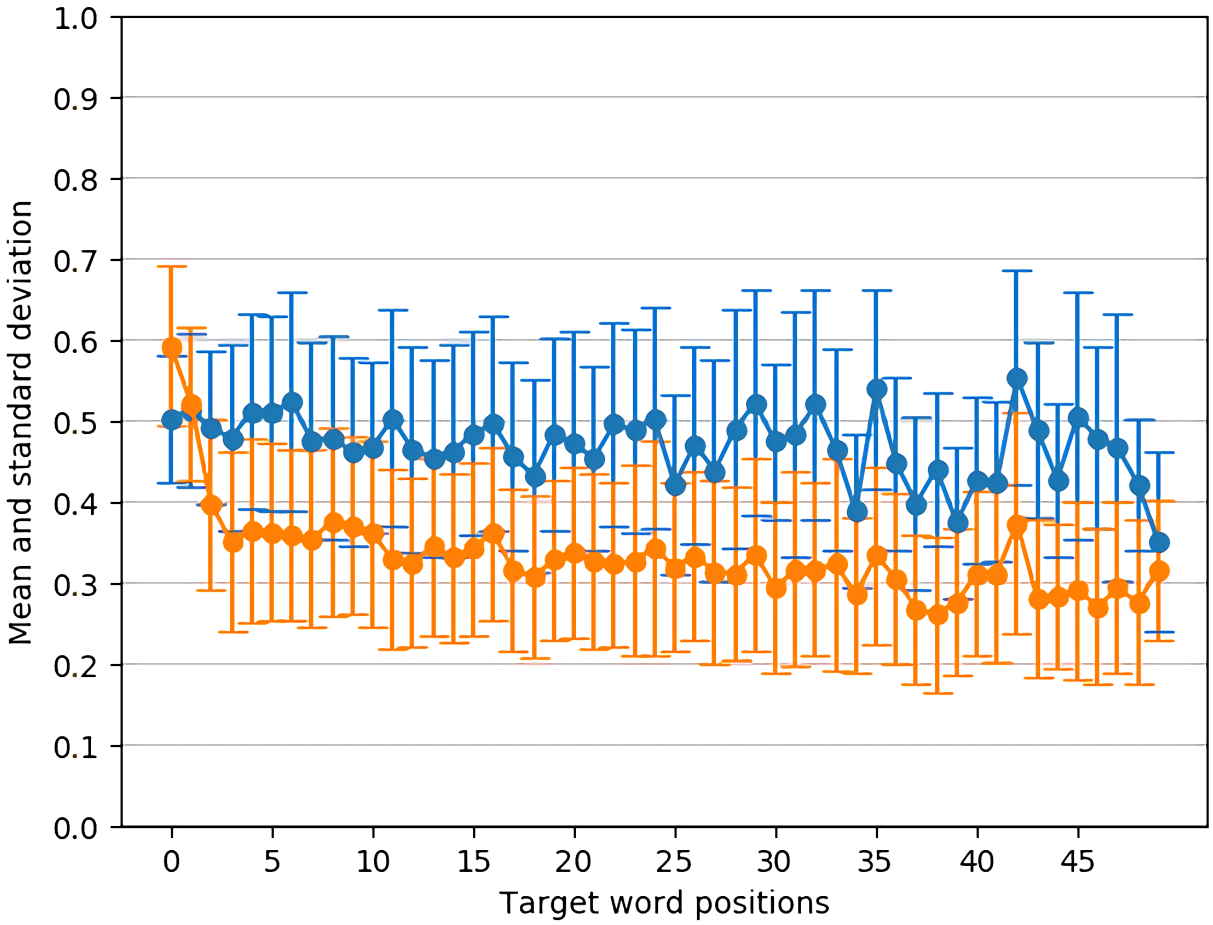

Figure 4.11: DGCM gates. Mean larger-context gate values (and standard deviation) at the decoder RNN level (orange) and the output level (blue), for each target word position.

that a high context gate activation decreases their importance and vice-versa.

The three set of gates behave quite differently. While there are important fluctuations for specific gates, those at the reset level often take smaller values, in contrast to the gates applied for the hidden state proposal. Moreover, the gates seem to behave distinctly for the first few words, similarly to the DGCM.

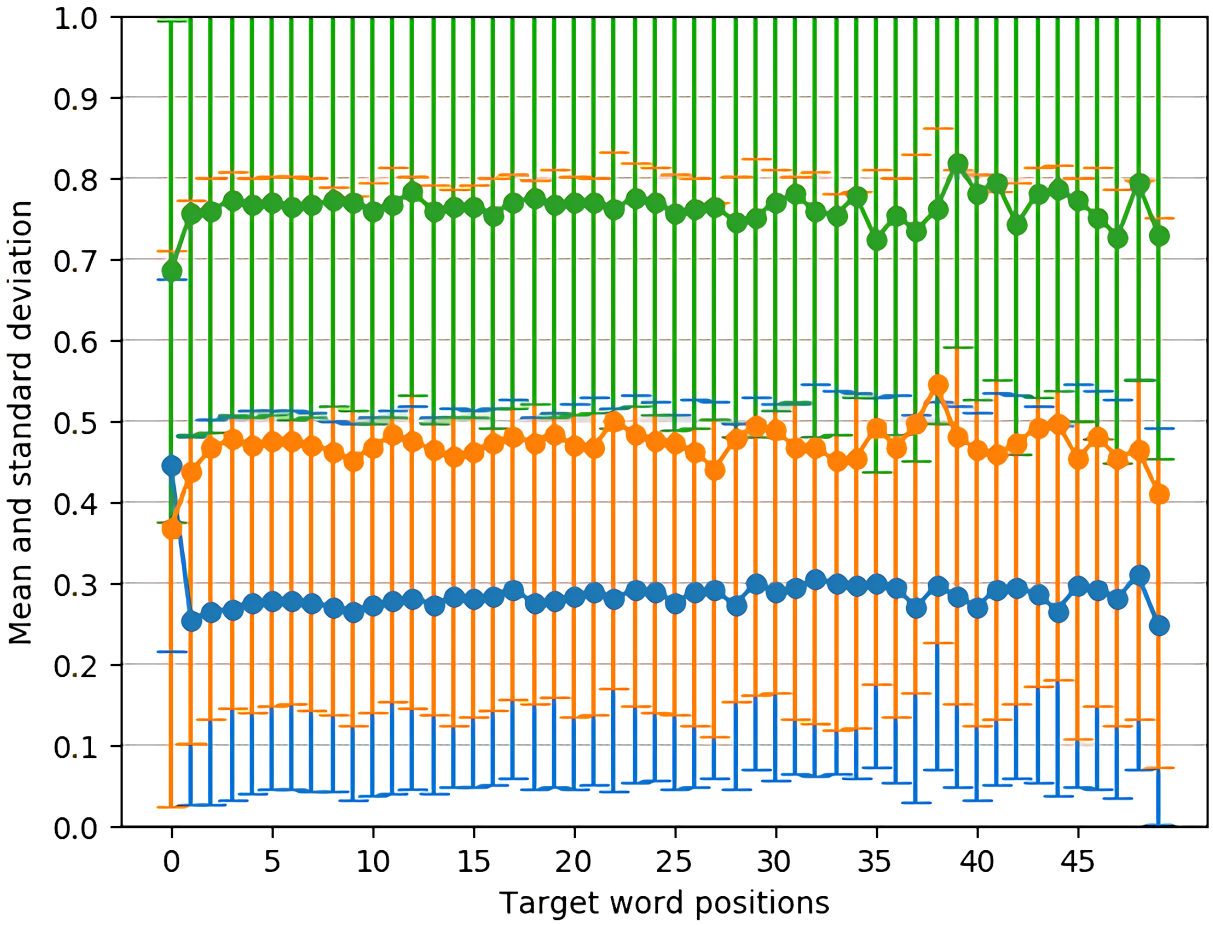

Figure 4.12: DCU gates. Mean larger-context gate values (and standard deviation) at the reset gates (blue), the update gates (orange) and the hidden state proposal (green).

4.7 Conclusion

In this chapter, we proposed some novel extensions of attention-based neural machine translation that seamlessly incorporate the context from surrounding sentences. Our extensive evaluation, measured both in terms of average translation quality and cross-lingual pronoun prediction, has revealed that the larger context can potentially be beneficial, at least under some data conditions. To validate our findings, we participated in the DiscoMT 2017 cross-lingual pronoun prediction shared task. The DGCM model often achieves better performance than the baseline by taking in

account the previous sentence, although we believe there is still important progress to be made. Finally, we analyzed the behavior of neural MT models in scenarios where a larger context should be helpful.

4.8 Since the release of this chapter

A few other RNN-based model variants were subsequently evaluated [Bawden et al. 2018]. The Transformer architecture was also introduced shortly after the release of this chapter [Vaswani et al. 2017]. Due to the better results it achieved, as well as its improved training efficiency, many Transformer-based larger-context NMT architectures have been proposed [Zhang et al. 2018; Voita et al. 2018; Miculicich et al. 2018]. These models differ in terms of the considered context (source/target) and attention patterns. A large-scale Transformer model, using concatenated data, was submitted to the WMT'19 shared task [Junczys-Dowmunt 2019].

The evaluation of larger-context NMT models has also progressed. Bawden et al. [2018] introduced challenge sets targeting coreferences and cohesion/coherence for English-to-French translation. For a given context, the likelihood of a correct (or semi-correct) and incorrect completion are compared. Accuracy is computed over all these contrastive pairs. Other linguistic phenomena, such as deixis, lexical cohesion and ellipsis were covered in Russian-to-English challenge sets [Voita et al. 2019b]. Test sets that directly evaluate the model outputs, instead of scoring right and wrong translations, were recently introduced [Jwalapuram et al. 2020].

5 | Emphasizing context

Interest in larger-context neural machine translation, including document-level and multimodal translation, has been growing. In this chapter, we explore different techniques to encourage models to emphasize context when needed. In particular, we propose a novel learning algorithm that explicitly encourages a neural translation model to take into account additional context using a multilevel pair-wise ranking loss. We evaluate the proposed learning algorithm with a transformer-based larger-context translation system on document-level translation. By comparing performance using actual and random contexts, we show that a model trained with the proposed algorithm is more sensitive to the additional context. The increased sensitivity can however potentially be harmful, with the model sometimes being unable to output a reasonable translation if it recognizes the context as fake.

Given this lack of robustness, we also consider data augmentation as an alternative approach to encourage NMT models to take context into account. In particular, given that most early statistical translation systems were constructed at the sentence level, many parallel corpora did not retain document-level information or metadata, leaving only aligned sentences. As context is often missing from many training parallel examples, it hinders the training of larger-context (i.e. document-level) machine translation systems. We consider the viability of filling in the missing contexts within the training data, and assume that source-side context will be available at test time. In particular, we consider three distinct approaches to generate the missing context: using randomly selected sentences, applying a copy heuristic, or generating the missing context with a

language model. We find that the copy heuristic significantly improves lexical coherence across generated sentences, as compared against other heuristics or two-stage refinement approaches previously proposed in the literature. We also validate the finding that using back-translation to augment our data with additional contextual data helps larger-context machine translation models better capture long-range phenomena, while also improving overall quality as measured by BLEU.

5.1 Recap: Larger-Context Neural Machine Translation

A larger-context neural machine translation system extends upon the conventional neural machine translation system by incorporating the context C, beyond a source sentence X, when translating into a sentence Y in the target language. This additional context could be an image described by X (multimodal machine translation), other source sentences in the same document (document-level MT), and possibly their translations as well.

In particular, in this chapter, we start by still considering the preceding source sentence as context, which acts as an additional input. We later use the previous translations, which can help maintain a more coherent output. In this scenario, at least for inference, the target-side context is also predicted.

A larger-context neural machine translation system may, for example, consist of an encoder f^C that encodes the additional context C into a set of vector representations that are combined with those extracted from the source sentence X by the original encoder f^X. Alternatively, by concatenating the source context and current sentence, a single encoder may jointly produce representations for both C and X. In the autoregressive paradigm, the conditional distribution over a target sequence Y is computed as $p_\theta(y_t|y_{<t}, X, C)$, where θ is a collection of all the parameters in the neural translation model. Before generating the target Y, some models also predict previous target-side context. The modules forming the larger-context machine translation system are of-

ten implemented as neural networks, such as recurrent networks with attention [Bahdanau et al. 2015], convolutional networks [Gehring et al. 2017] and self-attention [Vaswani et al. 2017].

Training is often done by minimizing the negative log-likelihood over the set of training examples (2.5).

5.1.1 Existing approaches to larger-context neural translation

Existing approaches to larger-context neural machine translation have mostly focused on either modifying the input or the network architecture. Simply concatenating the context to the input or target allows the re-use of existing architectures [Tiedemann and Scherrer 2017; Bawden et al. 2018; Grönroos et al. 2018; Junczys-Dowmunt 2019]. Other groups have proposed various modifications to neural translation systems [Jean et al. 2017; Wang et al. 2017; Voita et al. 2018; Zhang et al. 2018; Miculicich et al. 2018; Maruf and Haffari 2018; Tu et al. 2018] in the case of document-level translation, while using usual maximum likelihood learning. Alternatively, a multiple pass approach may be employed, for example by translation sentences in isolation and then post-editing the output in context [Voita et al. 2019b].

In parallel, there have been many proposals on novel network architectures for multimodal translation [Calixto et al. 2017; Caglayan et al. 2017; Ma et al. 2017; Libovický and Helcl 2017].

In personalized translation, Michel and Neubig [2018] bias the output distribution according to the context. Zheng et al. [2018] introduce a discriminator that forces the network to improve signal-to-noise ratio in the additional context.

Another complementary line of research explores evaluation metrics for large-context NMT, as the aggregate BLEU score only provides a generic measure of translation quality, hiding more subtle differences between systems. Moreover, as BLEU matches n-grams (n up to 4) between the output and the reference, it may fail to capture phenomena than span multiple sentences. Bawden et al. [2018] evaluate coreference resolution and coherence, while Müller et al. [2018] build a large test set to evaluate pronoun translation. Voita et al. [2019b] evaluate other linguistic

phenomena such as deixis and ellipsis.

5.2 Context-aware learning

Neural machine translation [Sutskever et al. 2014; Bahdanau et al. 2015] has already achieved considerable success (see e.g. Hassan et al. [2018]), although further progress may likely be reached by translating sentences in context [Läubli et al. 2018]. Despite efforts towards building models that can exploit additional context better, they sometimes appear to mostly ignore it, at least in terms of general automated metrics [Elliott 2018; Grönroos et al. 2018]. Nevertheless, there are encouraging signs showing that such models still learn about specific long-distance translation phenomena [Voita et al. 2019b], as well as promising human evaluation results [Junczys-Dowmunt 2019].

In the first part of this chapter, we approach the problem of larger-context neural machine translation from the perspective of "learning" instead of modelling. We propose to explicitly encourage the model to exploit additional context by assigning a higher log-probability to a translation paired with a correct context than with an incorrect one. We design this regularization term to be potentially applied at the token, sentence and batch levels to cope with the fact that the benefit from additional context may differ from one level to another. For example, we may expect context to be useful to predict some but not all tokens, while being generally helpful in aggregate. The proposed criterion is agnostic to the underlying model architecture, so that it can easily be combined with any future modelling improvements.

From a causal perspective, both the source sentence and the extra-sentential context may affect the translation output. As the second causal link is potentially weaker, we aim to amplify it by contrasting the usage of real and artificial contexts.

In our experiments on document-level translation using transformer networks [Vaswani et al. 2017; Voita et al. 2018], we see some improvement in terms of overall quality (measured in BLEU).

We also reveal that models trained using the proposed learning algorithm are indeed more sensitive to the context comparatively to some previous works [Elliott 2018]. From a robustness standpoint, many of the changes are detrimental. If the model recognizes the context as foreign, it can fail to produce a reasonable translation. Generally, a model trained with the proposed criterion is able to discriminate between real and fake contexts, acting as an implicit classifier.

5.2.1 Learning to use the context

In this first part, we focus on "learning" rather than on a network architecture. Our goal is to come up with a learning algorithm that can complement any underlying larger-context neural machine translation system.

5.2.1.1 Causality perspective

Let us introduce a hidden variable Z, which we assume to control the generation process for the underlying documents to be translated. As such, Z creates, or causes, the sources sentences X and their contexts C. In some cases, Y can be explained completely from X. However, as the source and target languages differ, the additional context C may provide information necessary to correctly explain Y. The idealized translation generation process is illustrated in Figure 5.1.[1]

As we expect the training signal from C to be weaker than the signal from the source X, we want the learning process to maximally exploit the effect of C on Y. If we cut the causal link between Z and C, or in other words generate the context from a different distribution, we expect the translation to sometimes differ. It would arguably be ideal to know how the output $\tilde{Y}$ would vary according to the modified context $\tilde{C}$, so that we could train the model directly on altered examples $(X, \tilde{Y}, \tilde{C})$. However, doing so may be difficult without significant expert knowledge. We instead leverage the assumption that, as long as the context partially causes the output, the probability of the original translations Y should on average decrease if an explaining factor is

[1] If the context C is generated before X, there may arguably be an additional link between C and X.

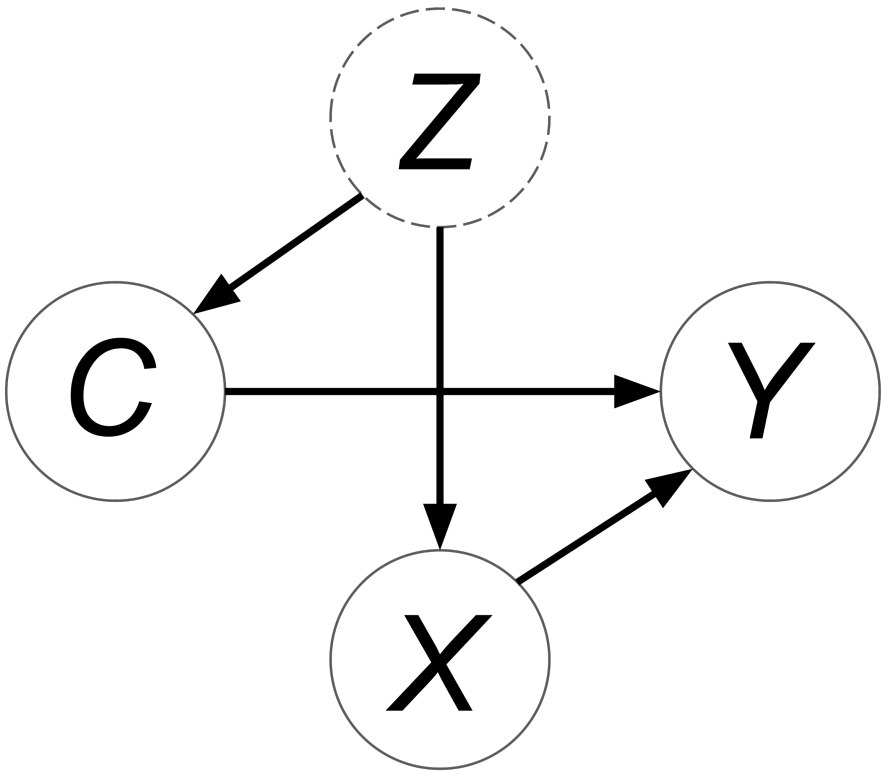

Figure 5.1: Causal diagram of a translation Y given a source-context pair $\{X, C\}$ generated from a hidden variable Z.

corrupted.

5.2.1.2 Neutral, useful and harmful context

We notice that by the law of total probability,

$$p_\theta(y_t|y_{<t}, X) = \sum_C p_\theta(y_t|y_{<t}, X, C)p(C|X) = \mathbb{E}_{C\sim C|X}\left[p_\theta(y_t|y_{<t}, X, C)\right] \tag{5.1}$$

As such, over the entire distribution of contexts given a source X, the additional context is overall "neutral".

The context C is "useful" if the model can assign a better probability to a correct target token y_t^* with it: $p_\theta(y_t^*|y_{<t}, X, C) > p_\theta(y_t^*|y_{<t}, X)$. On the other hand, the additional context can certainly

be used harmfully.

Although these "neutral", "useful" and "harmful" behaviours are defined at every timestep (*token level*), we can easily extend them to various levels by defining the following score functions:

$$\text{(token)} \qquad s^{\text{tok}}(y_t|\cdot) = \log p_\theta(y_t|\cdot),$$

$$\text{(sent.)} \qquad s^{\text{sent}}(Y|\cdot) = \sum_{t=1}^{T} \log p_\theta(y_t|y_{<t}, \cdot),$$

$$\text{(data)} \qquad s^{\text{data}}(\mathcal{Y}|\cdot) = \sum_{Y \in \mathcal{Y}} s^{\text{sent}}(Y|\cdot).$$

5.2.1.3 Context regularization

With these scores defined at three different levels, we propose to regularize learning to encourage a neural translation system to prefer using the context in a useful way. Our regularization term works at all three levels–tokens, sentences and the entire data– and is based on a margin ranking loss [Collobert et al. 2011]:

$$\begin{aligned}\mathcal{R}(\theta; \mathcal{D}) = \alpha_d & \left[\left(\sum_{n=1}^{N} T_n \right) \delta_d - s^{\text{data}}(\mathcal{Y}|\mathcal{X}, C) + s^{\text{data}}(\mathcal{Y}|\mathcal{X}) \right]_+ \\ & + \alpha_s \sum_{n=1}^{N} \left[T_n \delta_s - s^{\text{sent}}(Y_n|X_n, C_n) + s^{\text{sent}}(Y_n|X_n) \right]_+ \\ & + \alpha_\tau \sum_{n=1}^{N} \sum_{t=1}^{T_n} \left[\delta_\tau - s^{\text{tok}}(y_t^n|y_{<t}^n, X_n, C_n) + s^{\text{tok}}(y_t^n|y_{<t}^n, X_n) \right]_+ ,\end{aligned} \tag{5.2}$$

where α_d, α_s and α_τ are the regularization strengths at the data-, sentence- and token-level. δ_d, δ_s and δ_τ are corresponding margin values.

All three terms are similar, although they act at different granularities. The token-level term will be activated for every word or subword whose log-probability, after discounting the margin, is lower with context. Conversely, the data regularization term, generally applied for every batch during training, will only be activated if the aggregated log-probabilities are not high enough.

5.2.1.4 Estimating context-less scores

It is not trivial to compute the score when the context was missing based on Eq. (5.1), as it requires (1) the access to $p(C|X)$ and (2) the intractable marginalization over all possible C. In this paper, we explore the most practical strategy of approximating $p(C|X)$ with the data distribution of sentences $p_{\text{data}}(C)$.

We assume that the context C is independently distributed from the source X, i.e., $p(C|X) = p(C)$ and that the context C follows the data distribution. This allows us to approximate the expectation by uniformly selecting M training contexts at random:

$$s(\cdot|\cdot) = \log p(\cdot|\cdot) \approx \log \frac{1}{M} \sum_{m=1}^{M} p(\cdot|\cdot, C_m),$$

where C^m is the m-th sample.

A better estimation of $p(C|X)$ is certainly possible, such as with a larger-context recurrent language model [Wang and Cho 2016] or an off-the-shelf retrieval engine to build a non-parametric sampler.

5.2.1.5 An intrinsic evaluation metric

The conditions for "neutral", "useful" and "harmful" context also serve as bases on which we can build an intrinsic evaluation metric of a larger-context neural machine translation system. We propose this metric by observing that, for a well-trained larger-context translation system,

$$\Delta^{\mathcal{D}}(\theta) = s^{\text{data}}(\mathcal{Y}|\mathcal{X}, C; \theta) - s^{\text{data}}(\mathcal{Y}|\mathcal{X}; \theta) > 0.$$

That is, in aggregate, using additional contextual information should help the model predict the correct output with increased confidence. We compute this metric using the sample-based approximation scheme from above. Alternatively, we may compute the difference in BLEU

	Context	Context-Aware Reg.	BLEU		
			Normal	Context-Marginalized	$\Delta_{BLEU}^{\mathcal{D}^{test}}(\theta)$
(a)	○	○	29.16 (29.62)	-	-
(b)	○†	○	29.23 (29.65)	29.23 (29.65)	0
(c)	●	○	29.34 (29.63)	28.94 (29.23)	0.40
(d)	●	●	29.91 (30.13)	26.17 (25.82)	3.74

Table 5.1: We report the BLEU scores with the correctly paired context as well as with the incorrectly paired context (context-marginalized). Context-marginalized BLEU scores are averaged over three randomly selected contexts. BLEU scores on the validation set are presented within parentheses. † Instead of omitting the context, we give a random context to make the number of parameters match with the larger-context model.

($\Delta_{BLEU}^{\mathcal{D}}(\theta)$) over the validation or test data.

While this metric can provide important information about the sensitivity of a larger-context machine translation system, it doesn't distinguish improvements with appropriate contexts from deterioration with mismatched ones. As such, a high value is not always indicative of model quality.

5.2.2 Experimental Settings

Data We use En→Ru parallel data from OpenSubtitles2018 [Lison et al. 2018] and choose the same training data subset of 2M examples as [Voita et al. 2018] did. The dataset contains aligned movie and TV show subtitles, although there is no direct speaker annotation. As context, we use one preceding source sentence. We build a joint vocabulary of BPE subword tokens between the source and target languages using 32k merge operations [Sennrich et al. 2015].

Context-less score estimation We simply shuffle the context in each minibatch to create $M = 1$ random context per example. We could use multiple samples instead, but the estimator of $p(C|X)$ would be more costly to obtain and still remain biased.

Models We build a larger-context variant of the base transformer [Vaswani et al. 2017] that takes as input both the current and previous sentence, similarly to that by Voita et al. [2018]. The context C and source X are independently encoded by a common 6-layer transformer encoder. Another layer produces the final source representations by merging the context and source encodings (c and x respectively), as depicted in Figure 5.2. Using x as queries (q), a multi-head attention mechanism attends to c as key-values (k, v). The input and output of that attention layer are merged through a gate.[2] The final source representation is obtained through a feed-forward module (FF) used in typical transformer layers. We use a standard transformer decoder, which attends to the merged context-source representations, and share all the word embedding matrices.

We use Adam with an initial step size of 10^{-4}. We evaluate models every half epoch using greedy decoding and halve the learning rate when BLEU does not improve on the development set for five consecutive evaluations, following [Denkowski and Neubig 2017]. Models are evaluated with a beam size of 5, where sentence-level log-probabities are adjusted according to length [Wu et al. 2016] in order not to prefer too short outputs.

Based on the BLEU score on the validation set during the preliminary experiments, we set the coefficients and margins of the proposed regularization term (5.2) to $\alpha_\tau = \alpha_d = 1$, $\alpha_s = 0$, $\delta_\tau = \delta_s = 0$ and $\delta_d = \log(1.1)$. Early experiments didn't show clear benefits with $\alpha_s > 0$.

5.2.3 Results and Analysis

In Table 5.1, we present the translation quality (in BLEU) of the four variants. We make a number of observations. First, the use of previous sentence (c) does not improve over the baselines (a–b) when the larger-context model was trained only to maximize the log-likelihood (2.5). We furthermore see that the translation quality of the larger-context model only marginally degrades even

[2] Current gate values are unbounded. We did not observe clear improvements by applying a sigmoid function to restrict the range between 0 and 1.

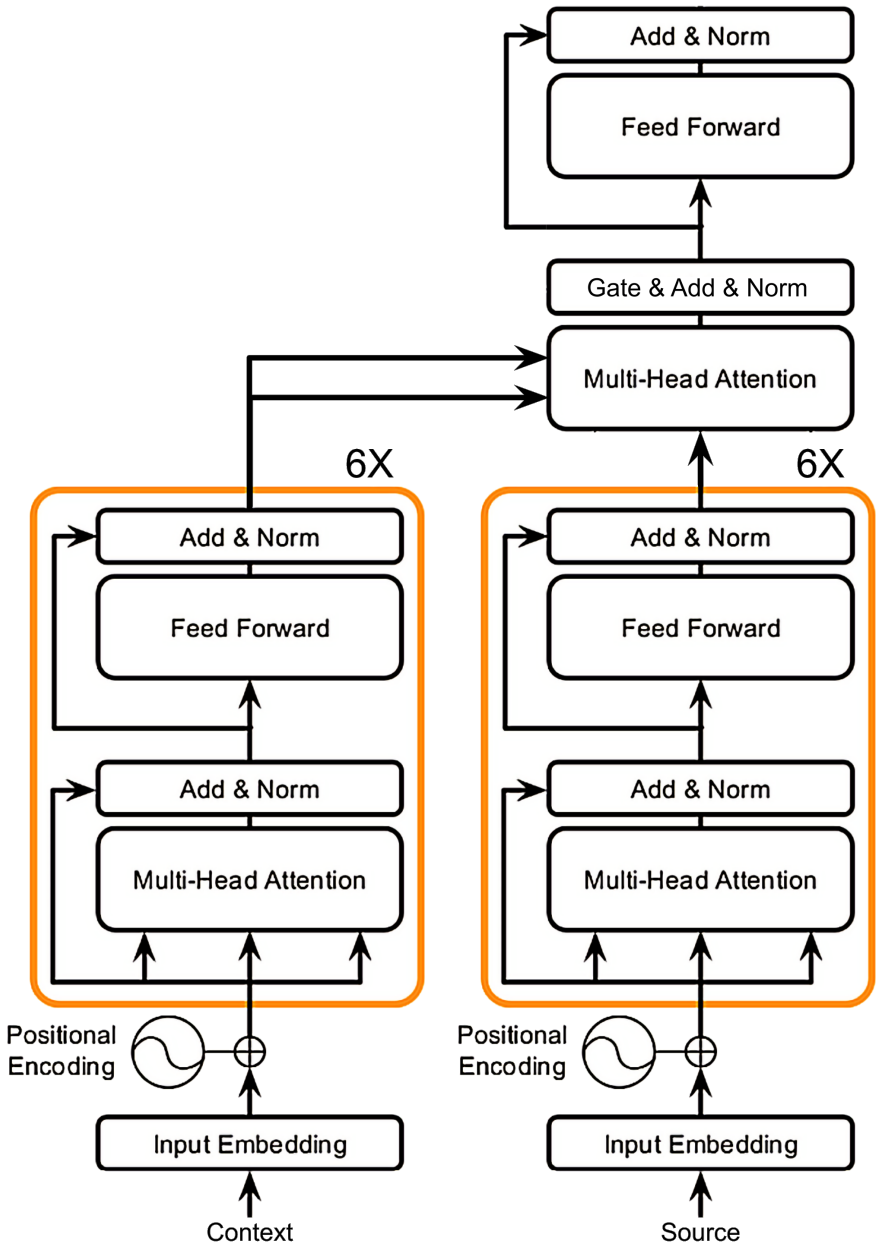

Figure 5.2: Context-aware encoder

when the incorrectly paired previous sentence was given instead ($\Delta^{\mathcal{D}^{test}}_{BLEU}(\theta) = 0.40$), implying that this model potentially ignores the previous sentence.

Second, we observe that the larger-context model, when trained with the proposed regularization term (d), improves upon the baselines, trained either without any additional context (a) or with purely random context (b). The evaluation metric $\Delta^{\mathcal{D}^{test}}_{BLEU}(\theta)$ is also significantly larger than 0, suggesting that the proposed regularization term encourages the model to focus on the additional context.

Cumulative BLEU based on log-likelihood difference In Fig. 5.3, for the regularized model (d), we contrast the translation qualities (measured in BLEU) between having the correctly paired (LC) and incorrectly paired (LC+Rand) previous sentences. The sentences in the validation set were sorted according to the difference $s^{\text{sent}}(Y|X,C) - s^{\text{sent}}(Y|X)$, and we report the cumulative BLEU scores. The gap is large for the sentences that are the most sensitive to additional context. This match between the sentence-level log-likelihood score difference (which uses the reference translation) and the translation quality further highlights the impact of the proposed approach.

5.2.3.1 Implicit context detection

While the model trained with the proposed regularization criterion is more sensitive to context, its ability to implicitly detect fake contexts may lead to a specific failure mode. With some contexts, beam search fails to terminate appropriately, in which case our implementation outputs an empty translation by default.

On the validation data, with the real contexts, this degenerate behaviour still occurs for 188 sentences out of 10,000. However, the impact on BLEU is hardly perceptible because the output length still approximately matches the reference length. As such, the brevity penalty had no impact.

With fake contexts, this phenomena is much more frequent. For one specific random permu-

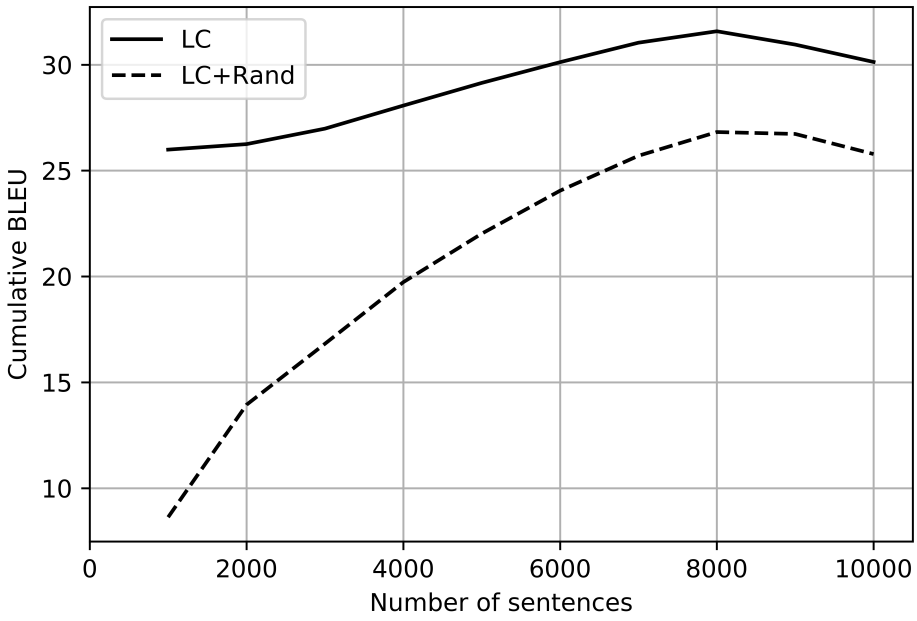

Figure 5.3: Cumulative BLEU scores on the validation set sorted by the sentence-level log-likelihood score difference according to the larger-context model.

tation of the contextual sentences, the beam search doesn't terminate cleanly for 1277 development set examples, resulting in a low BLEU of 25.78. If the poorly formed outputs are replaced by baseline translations, the development set BLEU score reaches 29.54, closing much of the gap to the baseline and larger-context models.

As such, while the model is clearly sensitive to context, it has in part learned to produce inappropriate outputs when it detects fake contexts.

In summary, the model discerns realistic contexts from fake ones, adapts its outputs, but potentially loses robustness. To address this lack of robustness, although perhaps at the cost of expressiveness, it would be possible to prevent back-propagation through the context-less scores, or in other words treating these scores as constants. Alternatively, instead of implicitly classifying contexts, the model could be modified to do so explicitly with a trainable gate, which would shut down if a potentially detrimental context is detected.

5.2.4 Conclusion

We proposed a novel regularization term for encouraging a larger-context machine translation model to focus more on the additional context using a multi-level pair-wise ranking loss. The proposed learning approach is generally applicable to any network architecture. Our empirical evaluation demonstrates that a model trained by the proposed approach becomes more sensitive to the additional context and adapts its output consequently. However, the decreased robustness may lead to degenerate behaviour, mostly with mismatched contexts, but also sometimes with appropriate ones.

5.3 Data augmentation for larger-context NMT

In the second part of this chapter, we encourage the use of extra-sentential context from a data viewpoint instead. We study document-level NMT from the perspective of availability of contextual training data and consider a scenario where contextual information is only available for a subset of the training examples. Previous work [Voita et al. 2019b; Xiong et al. 2019] has tackled this problem with a two-stage approach, initially translating sentences separately and then correcting the output in context using a separate model. Here, we explore the viability of generating the missing context, and then training an end-to-end translation model on the resulting mixture of naturally-occurring and synthetically-augmented document-level data.

We establish the viability of completing missing context in a scenario where only some of the parallel data has document-level context. We present multiple approaches to fill in the missing context and evaluate their impact on multiple linguistic phenomena and translation quality measured in BLEU. We also validate the effectiveness of (tagged) back-translation for document-level machine translation. Through our experiments, we demonstrate that simple context-completion techniques, in combination with back-translation, are competitive with complex two-stage ap-

proaches for larger-context neural machine translation.

In detail, we consider multiple context-augmentation techniques to impute the missing context for a subset of examples in the training set. In all our experiments, we do not modify the examples where context is already provided. As baselines, we leave the context-less examples intact, or simply add random sentence pairs as context to these examples [Junczys-Dowmunt 2019]. We then present a simple partial copy heuristic, where training examples are constructed by combining multiple copies of a source-target pair with random ones. Alternatively, we also evaluate generating the target-side context with a language model and obtaining the corresponding source context through back-translation.

By evaluating these context-aware models on multiple challenge sets [Voita et al. 2019b] targeting specific long-range phenomena, we observe that the choice of context completion technique clearly impacts quality. The partial copy heuristic leads to significantly improved lexical coherence. In contrast, creating examples where all sentences are unrelated is harmful on all challenge sets, even if this negative impact is not necessarily visible from the BLEU scores. We also compare the partial copy and random heuristic with human evaluation, on both generic and targeted test sets.

Given the importance of using appropriate data to capture long-range phenomena, we also provide additional evidence for the effectiveness of (tagged) back-translation [Sennrich et al. 2016; Caswell et al. 2019] for document-level machine translation [Junczys-Dowmunt 2019; Sugiyama and Yoshinaga 2019]. We demonstrate that adding back-translated document-level data is clearly helpful across all specific linguistic phenomena we evaluate, although the initial context completion technique still visibly impacts the end quality. Back-translation also increases the general translation quality as measured by BLEU, inline with the gains for sentence-level models studied in the literature.

5.3.1 Evaluating contextual translation systems

Various challenge sets have been constructed to specifically evaluate specific phenomena, such as coreference resolution, pronoun choice and coherence [Bawden et al. 2018; Müller et al. 2018]. For this paper, we consider four challenge sets targeting lexical cohesion, deixis (politeness marker), verb phrase ellipsis and morphological inflexion for English→Russian (En→Ru) [Voita et al. 2019b].

These four challenge sets consist of multiple-choice questions with identical source sentences, source contexts and target contexts. To answer an example appropriately, the model must assign a higher probability to the correct answer than to the distractors. However, as these are scoring challenges, choosing the right answer does not necessarily entail that the model would generate that exact sentence. Now we detail each challenge set used in our study.

Lexical cohesion This challenge set evaluates whether a model can consistently translate named entities across sentences. For each example, a named entity may be translated in multiple ways, each of which would be acceptable in isolation. However, over multiple sentences, a consensus must be reached. For a model to be successful on this task, use of target-side context is beneficial. It is split into validation and test subsets, of size 500 and 1500 respectively.

Deixis This challenge set specifically targets the T-V distinction (from the latin *tu* and *vos*). The T form is informal, while the V form is more polite, used for example when addressing people with higher social standing. The T-V distinction may involve changes to multiple words within the target sentence in order to maintain grammatical correctness. The T-V distinction is also present in other languages such as French (singular *vous*) or Spanish (*tú* / *usted*). As for lexical cohesion, use of target-side context is again crucial, and there is both validation and test data, of size 500 and 2500.

Sentence	Ru	En
Context-3	Крути педали.	Then pedal faster, Third Wheel.
Context-2	Она постоянно выходила из себя.	She got mad at everything.
Context-1	Что мне кажется удивительным.	Which I find surprising..
Current	Она постоянно выходила из себя.	She got mad at everything.

Table 5.2: Example augmented with the partial copy heuristic.

Verb phrase ellipsis This challenge set mostly considers the translation of the verb *do* (including its different conjugations). In Russian and several other languages, it is necessary to specify what is being done. The necessary information may often be found in neighboring sentences, either on the source or target side. In general, each example within this challenge set contains many more distractors than for lexical cohesion or deixis. Only test data is available.

Ellipsis (inflection) Russian words are richly inflected. There are two numbers (singular/plural), three genders (masculine/feminine/neuter) and six grammatical cases depending of the usage of a word. In some instances, the grammatical case of a translated word can not be inferred with certainty from the source sentence only, but context may help disambiguate the word. Similarly, in this challenge set only the test data is available.

5.3.2 Context completion

Many sources of parallel data include document-level information, such as recent versions of Europarl[3] and News-Commentary[4] [Barrault et al. 2019]. Nevertheless, there are many datasets where only out-of-context sentence pairs are available [Voita et al. 2019b]. In this paper, we consider a scenario where some, but not all, parallel data includes document-level information.

We ask how we may fill in the missing contextual information to improve the model quality. In particular, to match evaluation conditions, we modify the training data so that all examples have three source and target sentence pairs as context. Examples for which there already is context are

[3] statmt.org/europarl/v9
[4] data.statmt.org/news-commentary/v14

left untouched. Validation and test examples also remain unchanged. We explore three different techniques in order to impute missing sentences in context.

Random context A simple technique to fill in missing contextual information is to add random sentence pairs. This approach, although not limited to three contextual sentences exclusively, was previously used to create fake documents [Junczys-Dowmunt 2019] (in combination with other data augmentation techniques). As the artificial context will most likely be unrelated to the sentences to translate, this approach may have the adverse effect of biasing models towards ignoring context.

Partial copy heuristic As an alternative to using random context, we design a heuristic designed to encourage the model to sometimes consider the additional context, as well as to copy words or sub-word tokens.

Given a sentence pair, we build the context by adding a copy of the input, as well as two random sentence pairs. To improve robustness to sentence order, we randomly shuffle the four sentence pairs (2 identical, 2 distinct).

In addition we also tried copying the input multiple times, without any unrelated sentences. However, general performance suffered compared to the single copy with two random case, most likely because the training and evaluation conditions became too different. An augmented example with the proposed partial copy heuristic is included in Table 5.2.

Context generation In addition to two of the above explained heuristic approaches, we also considered a parametric approach to generate the missing context with a conditional neural language model.

Having access to distinct target-side monolingual data, we first extract consecutive segments of four sentences. Given the last sentence, next an encoder-decoder model is trained to predict the previous three sentences, similarly to a skip-thought model [Kiros et al. 2015]. Alternatively, a

standard language model could be also be employed using off-the-shelf language models [Radford et al. 2019], although the generation process would have to be modified slightly.

The trained target-to-context model then is applied to the original target side of the parallel data to generate the missing context sentences. This target-side model should be able to capture many context-dependent phenomena, favoring consistency between the original data and the generated context. In the generation process, we observed very repetitive outputs when decoding with beam search, so we instead use sampling [Cho 2015] to recover more diverse contexts.

One caveat of using auxiliary models to generate the missing context is that it might be distinguishable from real data. Since the translation systems are trained to mimic the output of such models, the quality of the generated samples could potentially affect the final translation quality.

Generation of the target side context is only the half side of the augmented context. In order to augment the source-side context, we back-translate the sampled context with a reverse-direction model. We also include the original target sentence within the input, but remove the last generated sentence from the output and replace it with the original source. For back-translation, we use a reverse-direction context-aware model trained with the partial copy heuristic.

5.3.3 Experiments

5.3.3.1 Parallel data

Both parallel and monolingual data are sourced from OpenSubtitles2018 [Lison et al. 2018], which is a collection of movie and TV show subtitles. We use the same 6 million En→Ru training examples as in Voita et al. [2019b], out of which 1.5 million have context. For these examples, the previous three source and target sentences are provided. Given the fixed-sized context, many of these training instances have overlapping sentences. We use the validation and test sets provided by Voita et al. [2019b] for both general quality and targeted evaluation.

We train a SentencePiece model [Kudo and Richardson 2018] separately on the English and

Russian side of the training data, with vocabularies of size 32,768. Given the different tokenization, BLEU scores are not directly comparable to Voita et al. [2019b,a], so we retrain the necessary baseline models. Note that, scores on the lexical cohesion, deixis, ellipsis (VP) and ellipsis (infl.) challenge sets remain comparable.

5.3.3.2 Monolingual data

We gather document-level Russian monolingual data from OpenSubtitles2018. As the raw data is time-stamped, we concatenate multiple consecutive sentences within the same movie or TV show as long as the time difference between two sentences is at most two seconds.

The retrieved segments, which may contain arbitrarily many sentences, are filtered to avoid overlap with all the validation and test sets. More specifically, the last sentence of any validation or test example is prohibited to be within the monolingual training data, but contextual sentences are allowed.[5] Segments with less than four sentences are removed, while longer segments are split into overlapping four-sentence examples. The final data contains approximately 26 million examples.

This filtering criterion is only applied to the monolingual data we extract, but not to the parallel data [Voita et al. 2019b] in order to be comparable with prior work. In that case, the training and test instances came from different movies, which could still have overlapping sentences.

5.3.3.3 Models

We use Transformer models [Vaswani et al. 2017] for all experiments. In particular, with bilingual data only, we use Transformer base models, with 6 encoder and decoder layers, 8-head attention, model and feed-forward projection dimensions of 512 and 2048 respectively. In order to regularize the model, the dropout rate is set to 20% at all applicable places, i.e. residual connections, feed-forward networks and attention weights.

[5] White-space within sentences is removed for filtering to deal with potential tokenization differences.

With additional target-side monolingual data and back-translated source inputs, we additionally train Transformer big models, with twice as many attention heads and double the dimensions. Even with augmented training data, we observed significant over-fitting, and as such set the dropout hyper-parameter to 40%.

Models are trained on v3 TPUs [Jouppi et al. 2017] in a 4x4 topology (32 cores). Baseline models are trained with a batch size of 8192 target tokens per core. Multiple sentences, of maximum length 98, are packed within each row to improve computational efficiency. To allow for longer inputs for context-aware models, batches are kept at the same size, but this time with maximum length 512 tokens. Sentences within each example are concatenated to each other, with a reserved token separating each sentence. All our experiments were performed using the Tensorflow Lingvo [Shen et al. 2019] framework.

Model outputs are generated with beam search, with a beam width of 8. BLEU scores are evaluated with an in-house re-implementation of mteval-v13a (on the last sentences only, i.e. excluding the context).[6]

Auxiliary systems The reverse-direction system used to back-translate additional data is trained on the parallel corpus only, with the missing context replaced with the partial copy heuristic. All the extracted monolingual four-sentence examples are back-translated, unless their length exceeds 512 subword tokens. Following Caswell et al. [2019], a tag is prepended to the generated source samples. Note that for the baseline system trained with additional monolingual data, we still use the context-aware Ru→En model for back-translation, but only add the last sentence pair (out of four) to the training data.

To train the context generation model, we reformat the monolingual data so that the last sentence within a block of four is used as input to the encoder, while the three previous ones are predicted in a left-to-right manner. To reduce the distance between the input and the first

[6] github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/mteval-v13a.pl

Model	BLEU
Baseline	**32.3** (31.8)
Concat	31.0 (30.8)
Random	31.8 (31.5)
Partial copy	31.6 (31.3)
Context generation*	31.5 (31.4)

Table 5.3: En→Ru BLEU scores, with parallel data only. Validation results in parentheses. *Note that Context generation uses additional monolingual data to train a language model.

Model	Deixis	Lex. cohesion	Ellipsis (infl.)	Ellipsis (VP)
Baseline	50.0 (50.0)	45.9 (46.2)	52.0	25.0
Concat	83.4 (87.2)	48.9 (48.2)	76.0	73.8
Random	69.9 (68.8)	45.9 (46.4)	63.6	62.3
Partial copy	**86.6** (85.1)	**74.9** (75.4)	75.5	77.9
Context generation*	85.6 (86.0)	60.0 (59.6)	74.8	74.2
Concat [Voita et al. 2019b]	83.5	47.5	**76.2**	76.6
CADec [Voita et al. 2019b]	81.6	58.1	72.2	**80.0**

Table 5.4: En→Ru challenge set accuracy, with parallel data only. Validation results in parentheses.

predicted words, we also considered models that generated either sentences or tokens from right-to-left, but settled on the first approach. We use transformer base models to generate the missing context.

5.3.4 Results

With parallel data only,[7] En→Ru BLEU scores are shown in Table 5.3. The concat model, where context is available for only a quarter of the examples, and is prepended to the data in such cases, trails the baseline by more than 1 BLEU. This model is also prone to severe over-fitting as the training and test conditions differ significantly.

By completing the missing context in a variety of ways, the BLEU gap is mostly closed, although the baseline still performs better for En→Ru. Using the same hyper-parameters, there is still some over-fitting, but it is less severe than for the concat model. We note that the use of random contexts to fill in data is as effective as other methods, at least in terms of BLEU.

[7]Except for Context generation, which uses additional monolingual data to train the language model.

Model	BLEU	Deixis	Lex. cohesion	Ellipsis (infl.)	Ellipsis (VP)
Random	**31.8**	69.9	45.9	63.6	62.3
Partial copy - 2	31.6	**86.6**	74.9	75.5	77.9
Partial copy - 3	31.5	86.2	76.7	**76.2**	**78.2**
Full copy	30.9	85.4	**81.3**	75.6	76.8

Table 5.5: Progression from random to copied contexts.

In the opposite direction (Ru→En), the concat model still trails the baseline and overfits quickly, even with added regularization. With the partial copy heuristic, overall performance rises (39.4 BLEU), becoming very similar to the baseline model (39.2 BLEU).[8]

On the various challenge sets (Table 5.4), the concat models show gains over the context-agnostic baseline, in agreement with previous results reported by Voita et al. [2018]. However, the gains for lexical cohesion are minimal.

When missing context is replaced with random sentence pairs, performance drops across all tasks compared to the concat model, but remains above the baseline. As such, the model still benefits from data where real context is available (1.5M examples), but its ability to capture cross-sentential phenomena is impeded by the additional examples.

With the partial copy heuristic, the model obtains an accuracy of 74.9% on the lexical cohesion test set, improving by 26% over the concat model with the original data. Compared to CADec [Voita et al. 2019b], a two-stage approach in which a baseline model is trained over the entire parallel corpus, and a post-editing model is built from the subset that has context, there is a gain of 15%. Over the other challenge sets, performance is comparable or slightly higher (-0.5% to 4.1% accuracy difference) than with unmodified training data.

When generating the missing context with separate models, accuracy on the lexical cohesion test set reaches 60.0%, an improvement over the original data, but still less than with the copying heuristic. Results on the other challenge sets are similar to those obtained with unaugmented parallel data.

[8]Given the lack of Ru→En challenge sets, and as we mainly use Ru→En for back-translation, we did not train models with all the context completion techniques.

Model	Run	BLEU	Challenge sets
Random	1	31.9	61.5
	2	32.0	60.2
	3	31.5	60.0
	Avg.	31.8 ± 0.2	60.6 ± 0.7
Partial copy	1	31.7	78.4
	2	31.4	78.8
	3	31.6	79.0
	Avg.	31.6 ± 0.1	78.7 ± 0.2

Table 5.6: Robustness experiments across 3 runs with different data and model random seeds. Each challenge set is weighted equally.

Robustness To characterize the models' robustness to random fluctuations, we trained three distinct copies for both the random and partial copy context completion heuristics. Both the data generation and model parameter seeds were modified for each run.

Results are reported in Table 5.6. For both data completion techniques, BLEU scores differ by 0.5 or less across runs. Averaging results over the four challenge sets, performance variations between runs with similarly generated contexts remain within 2%, while there is almost a 20% difference between the two context imputation approaches.

5.3.4.1 Human evaluation

Given the large improvements on scoring-based challenge sets with the partial copy heuristic over random contexts, but the small BLEU differences, we run human evaluations to assess the general validity of the gains. We use 500 examples each from the main test set, as well as the deixis and lexical cohesion test sets. Raters are shown both translations together, and asked to rate them between 0 and 6.

For general quality, the partial copy output is respectively superior, neutral or inferior in 25%, 52% and 23% of cases, for a win/loss ratio of 1.09. 50% of partial copy translations reach a score superior to 4 ("Most meaning preserved and few grammar mistakes" or better), as opposed to 44% with the random context generation.

Human evaluation results are mostly neutral on the deixis test set. On the lexical cohesion

test set, we observe larger improvements, with a win/loss ratio of 1.25.

Overall, the substantial gains on lexical cohesion from our approach lead to smaller but meaningful improvements on the general test set.

5.3.4.2 From random to fully copied contexts

Using either random or partially copied contexts lead to similar BLEU scores, but very different results on the challenge sets. As such, we are interested in the progression from random contexts to exclusively copying the input when training models. Training examples with existing context (1.5M out of 6M), as well as all test data, remain unchanged.

As the input is copied more frequently, performance on the lexical cohesion challenge set improves, reaching 81.3% accuracy in the most extreme scenario. Nevertheless, as Russian proper nouns are declined, a copying mechanism may not capture all relevant phenomena, especially for words that are not split into multiple tokens.

On each of the other challenge sets, as long as missing context is copied at least once, performance remains within a tight 2% window. However, as previously remarked, using random contexts leads to much worse results on these test sets.

As the input sentence pair is copied more extensively, BLEU starts to decrease, down to 30.9 when missing context is exclusively replaced by copies of the input sentence pair. This behaviour can likely be attributed to the increasing mismatch between training and inference conditions. Moreover, the model may be over-incentivized to copy the context at the expense of translating the source sentence. As such, the copy heuristic may lead to overfitting to the lexical cohesion challenge set, where repetitions are encouraged.

5.3.4.3 With monolingual data

With additional monolingual data, which is back-translated in context, and also using larger models, BLEU scores unsurprisingly improve (Table 5.8), reaching scores between 33.0 and 33.4. The

Model	Deixis	Lex. cohesion	Ellipsis (infl.)	Ellipsis (VP)
Baseline	50.0 (50.0)	45.9 (46.2)	54.0	28.8
Concat	87.8 (88.8)	86.1 (85.6)	**87.6**	**88.8**
Random	85.4 (87.6)	82.5 (80.0)	82.8	85.2
Partial copy	89.6 (89.8)	**90.1** (89.8)	86.6	88.6
Context generation	90.6 (90.8)	85.3 (83.2)	85.6	87.2
DocRepair [Voita et al. 2019a]	**91.8**	80.6	86.4	75.2

Table 5.7: En→Ru challenge set accuracy, with additional back-translated data. Validation results in parentheses.

Model	BLEU
Baseline	33.0 (33.1)
Concat	33.2 (33.1)
Random	**33.4** (33.3)
Partial copy	**33.4** (33.1)
Context generation	**33.4** (33.2)

Table 5.8: En→Ru BLEU scores, with additional back-translated data. Validation results in parentheses.

context-aware models no longer trail behind the baseline, and the concat model now performs comparably to others. Note that, with bilingual and back-translated data mixed in a ratio close to 1, only approximately 3/8 of the data has either missing or automatically generated target context, as opposed to 3/4 with the original bilingual data.

Quality on all challenge sets increase significantly for all models, except for the context-less baseline (Table 5.7). With the original bilingual data and additional back-translated data (concat), accuracy reaches 86.1% or better on all four test sets. If missing contexts are instead replaced by random ones, performance drops on average by 3.65%. A model trained with the partial copy heuristic has the best performance on the lexical cohesion test set, at 90.1%, and otherwise comparable performance to the concat model.

We also present results for the DocRepair model [Voita et al. 2019a], where the outputs of a sentence-to-sentence baseline are refined in context. The post-editing model is trained from round-trip translated monolingual data (without context), so that the output is contextually coherent, but not necessarily the input. Contrarily to our approaches, DocRepair does not use any

document-level parallel data, but could potentially benefit from it.

Note that while we use monolingual data from the same corpus, and a similar amount of examples, our filtering procedures differ, so the training data is not exactly the same.[9]

DocRepair obtains the best performance on the deixis test set, with a margin of 1.2% over our best model on this task. Results on the ellipsis (infl.) challenge sets are similar to those obtained with single-pass context-aware translation systems. However, these single-pass models perform up to 9.5% and 13.6% better on the lexical cohesion and VP ellipsis challenge sets respectively. In particular, while VP ellipsis may be a hard phenomenon to capture with monolingual data only [Voita et al. 2019a], a context-aware translation system trained on sufficiently many examples may perform well on this task.

5.3.5 Conclusion

When document-level context is only partially available, we evaluate the effectiveness of various data completion techniques, using both BLEU, challenge sets targeting specific linguistic phenomena and human evaluation. In particular, a simple copy heuristic helps models achieve much better lexical cohesion, even for a highly inflected language such as Russian. Only adding random context sentence pairs, however, reduces a model's ability to capture cross-sentence interactions, yet these effects are not visible from BLEU scores. Additionally, we confirm the effectiveness of back-translation on overall translation quality, while also demonstrating its usefulness on the four challenge sets.

As even simple context completion techniques have a clear impact on model performance, it may be worthwhile to explore additional approaches. In particular, to obtain more natural contexts, while limiting model generation errors, we can envision embedding sentences such that neighbours in vector space are probable contexts of each other. Moreover, it might be useful

[9]The monolingual data used by Voita et al. [2019a] was not yet publicly available when we conducted our experiments.

to understand how different data augmentation schemes interact with more sophisticated model architectures.

5.4 Since the release of this chapter

Alternative learning algorithms have been proposed for larger-context neural machine translation. Minimum risk training can be applied at the document level [Saunders et al. 2020], or specific discourse rewards can be integrated with reinforcement learning [Unanue et al. 2020]. Ma et al. [2021] further verify the effectiveness of back-translation for larger-context NMT. Context completion still remains largely unexplored. Given its potential impact, but the possible artefacts introduced by copy heuristics, we believe that future approaches should produce natural-looking and diverse contexts. The context generation approach could potentially be combined with techniques such as nucleus sampling and unlikelihood learning to obtain higher-quality contexts [Holtzman et al. 2020; Welleck et al. 2020]. Moreover, improvements to language modelling or sequence-to-sequence approaches could likely be leveraged to obtain better contexts.

6 Log-linear reformulation of the noisy channel model for larger-context NMT

We seek to maximally use various data sources, such as parallel and monolingual data, to build an effective and efficient document-level translation system. In particular, we start by considering a noisy channel approach [Yu et al. 2020] that combines a target-to-source translation model and a language model. By applying Bayes' rule strategically, we reformulate this approach as a log-linear combination of translation, sentence-level and document-level language model probabilities. In addition to using static coefficients for each term, this formulation alternatively allows for the learning of dynamic per-token weights to more finely control the impact of the language models. Using both static or dynamic coefficients leads to improvements over a context-agnostic baseline and a context-aware concatenation model.

6.1 Introduction

Neural machine translation (NMT) [Sutskever et al. 2014; Bahdanau et al. 2015] has been reported to reach near human-level performance on sentence-by-sentence translation [Läubli et al. 2018]. Going beyond sentence-level, document-level NMT aims to translate sentences by taking into

account neighboring source or target sentences in order to produce a more cohesive output [Jean et al. 2017; Wang et al. 2017; Maruf et al. 2021]. These approaches often train new models from scratch using parallel data.

In this chapter, in a similar spirit to Voita et al. [2019a]; Yu et al. [2020], we seek a document-level approach that maximally uses various available corpora, such as parallel and monolingual data, leveraging models trained at the sentence and document levels, while also striving for computational efficiency. We start from the noisy channel model [Yu et al. 2020] which combines a target-to-source translation model and a document-level language model. By applying Bayes' rule, we reformulate this approach into a log-linear model, similarly to phrase-based systems (6.1). It consists of a translation model, as well as sentence and document-level language models. This reformulation admits an auto-regressive expression of token-by-token target document probabilities, facilitating the use of existing inference algorithms such as beam search. In this log-linear model, there are coefficients modulating the impact of the language models. We first consider static coefficients and, for more fine-grained control, we train a *merging module* that dynamically adjusts the LM weights.

With either static or dynamic coefficients, we observe improvements over a context-agnostic baseline, as well as a context-aware concatenation model [Tiedemann and Scherrer 2017]. Similarly to the noisy channel model, our approach reuses off-the-shelf models and benefits from future translation or language modelling improvements.

6.2 Log-linear reformulation of the noisy channel model

Given the availability of various heterogeneous data sources that could be used for document-level translation, we seek a strategy to maximally use them. These sources include parallel data, at either the sentence or document level, as well as more broadly available monolingual data.

As the starting point, we consider the noisy channel approach proposed by Yu et al. [2020].

Given a source document $(X^{(1)}, \ldots, X^{(N)})$ and its translation $(Y^{(1)}, \ldots, Y^{(N)})$, they assume a generation process where target sentences are produced from left to right, and where each source sentence is translated only from the corresponding target sentence. Under these assumptions, the probability of a source-target document pair is given by

$$P(X^{(1)}, \ldots, X^{(N)}, Y^{(1)}, \ldots, Y^{(N)}) = \prod_{n=1}^{N} P(X^{(n)}|Y^{(n)}) P(Y^{(n)}|Y^{(<n)})$$

As such, the conditional probability of the target document given the source is expressed by

$$\begin{aligned} P(Y^{(1)}, \ldots, Y^{(N)}|X^{(1)}, \ldots, X^{(N)}) &\propto \prod_{n=1}^{N} P(X^{(n)}|Y^{(n)}) P(Y^{(n)}|Y^{(<n)}) \\ &= \prod_{n=1}^{N} \underbrace{P(Y^{(n)}|X^{(n)}) \frac{P(Y^{(n)}|Y^{(<n)})}{P(Y^{(n)})}}_{\propto P(Y^{(n)}|X^{(n)}, Y^{(<n)})}. \end{aligned}$$

We therefore generate context-aware translations by combining a translation model (TM) $P(Y^{(n)}|X^{(n)})$ with both sentence-level $P(Y^{(n)})$ and document-level $P(Y^{(n)}|Y^{(<n)})$ language models (LM). To calibrate the generation process, we introduce coefficients $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}$ to control the contribution of each language model, which are tuned on a validation set:

$$\begin{aligned} &\log P(Y^{(n)}|X^{(n)}, Y^{(<n)}) \\ &= \sum_{i=1}^{L_n} \left[\log P(y_i^{(n)}|y_{<i}^{(n)}, X^{(n)}) + \alpha \log P(y_i^{(n)}|y_{<i}^{(n)}, Y^{(<n)}) - \beta \log P(y_i^{(n)}|y_{<i}^{(n)}) + C_i^{(n)} \right], \end{aligned} \tag{6.1}$$

where $C_i^{(n)}$ is a normalization constant and L_n is the target sentence length.

Similarly to the noisy channel approach [Yu et al. 2020], we use off-the-shelf translation and language models. As such, future improvements to either translation or language modelling can

easily be leveraged. Our reformulation however admits a more efficient search procedure, unlike that by Yu et al. [2020].

6.2.1 Model parameterization

The translation model is implemented as any auto-regressive neural translation model. We use the Transformer encoder-decoder architecture [Vaswani et al. 2017]. Given a source sentence $x_1, \ldots, x_L$, each token and its position are projected into a continuous embedding $s_{0,1}, \ldots, s_{0,L}$. These representations are passed through a sequence of M encoder layers that each comprise self-attention and feed-forward modules, resulting in the final representations $s_{M,1}, \ldots, s_{M,L}$. The decoder updates target embeddings through similar layers, which additionally attend to the encoder output, to obtain final hidden states $t_{M,1}, \ldots, t_{M,L}$. Token probabilities may be obtained by projecting these representations and applying softmax normalization.

Language models are implemented as Transformer decoders without cross-attention. We use a single language model trained on sequences of consecutive sentences to obtain both sentence-level and document-level probabilities.

6.3 Dynamic merging

As extra-sentential information is not uniformly useful for translation, we propose dynamic coefficients for the different models by generalizing Eq. 6.1:

$$\mathcal{L} = -\sum_{n=1}^{N}\sum_{i=1}^{L_n}\left[\log P(y_i^{(n)}|y_{<i}^{(n)}, X^{(n)}) + \alpha_i^{(n)}\log P(y_i^{(n)}|y_{<i}^{(n)}, Y^{(<n)}) - \beta_i^{(n)}\log P(y_i^{(n)}|y_{<i}^{(n)}) + C_i^{(n)}\right]. \tag{6.2}$$

With the translation and language models kept fixed, the coefficients $\alpha_i^{(n)}$ and $\beta_i^{(n)}$ are com-

puted by an auxiliary neural network which uses $Y^{(<n)}$, $Y^{(n)}$ and $X^{(n)}$. We call this network a *merging module* and implement it as a feed-forward network on top of the translation and language models.

6.3.1 Dynamic coefficient computation

For every token, the corresponding last hidden states of the translation model, sentence-level LM and document-level LM are concatenated. Each non-final layer ($k = 1, \ldots, K-1$) is a feed-forward block

$$h_k = \text{LN}(h_{k-1} + \text{drop}(W_{k,2}(\text{ReLU}(W_{k,1}h_{k-1}))),$$

where LN and drop respectively denote layer normalization and dropout [Ba et al. 2016; Srivastava et al. 2014]. The final layer is similar, but there is no residual connection (and no dropout) as the final linear transformation projects the result to 2 dimensions, so that $(\alpha, \beta) = W_{K,2}(\text{ReLU}(W_{K,1}h_{K-1}))$.

6.4 Experiments

6.4.1 Settings

Data We run experiments on English-Russian data from OpenSubtitles [Lison et al. 2018], which was used in many recent studies on document-level translation [Voita et al. 2019b,a; Mansimov et al. 2020; Jean et al. 2019]. Language models are trained on approximately 30M sequences of 4 consecutive sentences [Voita et al. 2019a]. The parallel data was originally preprocessed by Voita et al. [2019b], yielding 6M examples. For 1.5M of these data points, the 3 preceding source and target sentences are provided. We use this subset to train the *merging module* that predicts the per-token coefficients for each model. We uniformly set the number of contextual sentences between 1 and 3 to match the test condition.

We apply byte-pair encoding (BPE) [Sennrich et al. 2015], with a total of 32k merge operations, separately on each language pair, as Russian and English use different sets of alphabets.

Models Translation models are standard Transformers in their base configuration [Vaswani et al. 2017]. The language model is implemented as a Transformer decoder of the same size, except for a smaller feed-forward dimension $d_{ff} = 1024$. The *merging module* has 2 layers, with $d_{ff} = 1536$.

Learning The translation and language models, as well as the *merging module*, are trained with label smoothing set to 10%. The TM is trained with 20% dropout, while it is set to 10% for the LMs and *merging module*.

Evaluation Translation quality is evaluated with tokenized BLEU on lowercased data, using beam search with its width set to 5. We average 5 checkpoints for the translation models. Sentences are generated from left to right, and the beam is reset for every sentence.

6.4.2 Results

With our approach, using static coefficients, we reach a BLEU score of 34.31, which is a modest gain of 0.21 BLEU over the baseline and 0.8 over a model trained on concatenated sentences (Table 6.1). By optimizing dynamic coefficients, we reach a similar score of 34.22.

DocRepair [Voita et al. 2019a], a two-pass method that post-edits the output of a baseline system, obtains a slightly higher BLEU score of 34.60. Both approaches could be combined by instead post-editing the output of our models, which we leave for future investigation.

BLEU-NLL correlation We observe limited correlation between BLEU and reference NLL [Och 2003; Lee et al. 2020b]. On the validation set, the per-token baseline loss (with label smoothing)

	BLEU
Baseline	34.10
Concat	33.51
Static coeffs.	34.31
Dynamic coeffs.	34.22
CADec	33.86
DocRepair	**34.60**

Table 6.1: Test set BLEU scores (beam width 5, all 4 sentences concatenated). CADec and DocRepair results from [Voita et al. 2019a].

β \ α	0	0.2	0.4	0.6
0	31.5	31.0	29.3	26.9
0.2	30.7	**31.7**	31.2	29.5
0.4	23.3	30.1	31.6	31.1
0.6	14.3	21.9	26.9	30.8

Table 6.2: Greedy validation BLEU (last sentence only) for different static values of α and β. Both LMs are critical to the approach.

is 13.09. Using static coefficients, it actually increases to 13.23, while it decreases to 12.86 with dynamic coefficients.

Contribution of each language model (static) Table 6.2 presents the BLEU scores on the validation set using greedy validation for different static values of α and β. Only using the document-level LM ($\alpha > 0, \beta = 0$) leads to worse performance than the baseline. It is critical to counter-balance the document-level LM with the sentence-level LM.

Dynamic coefficients The dynamic coefficients α and β predicted by the *merging module* are highly correlated (Figure 6.1 (left)). As a conjecture, this high correlation may be explained by the use of the same language model to obtain both sentence and document-level scores.

Figure 6.1 (right) shows the average value of the dynamic coefficient α for frequent words within the validation reference set. In particular, Ты and Вы, which are translations of *you* that

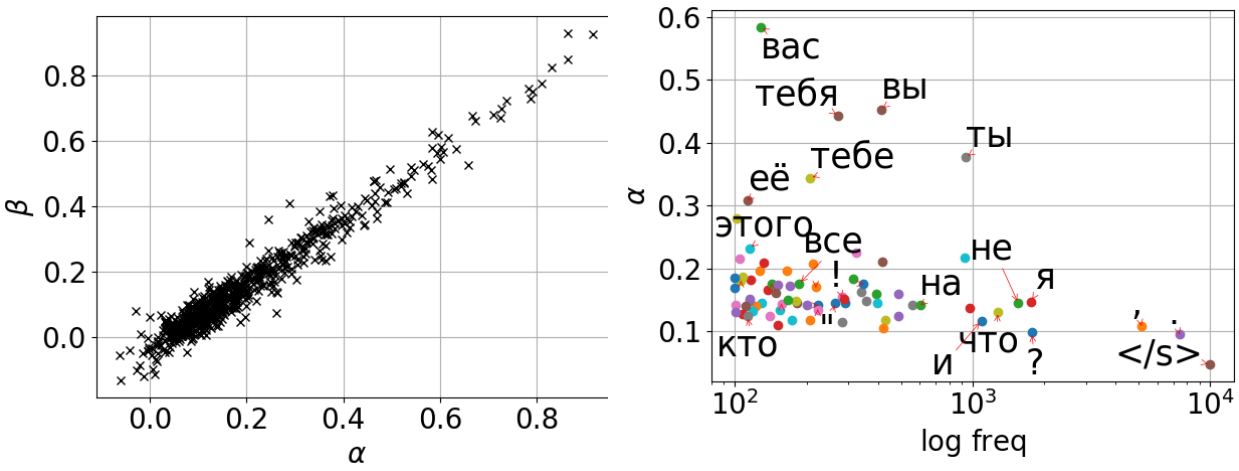

Figure 6.1: Scatter plot of α and β for tokens appearing at least 100 times over the validation set (left). Average dynamic coefficient α for frequent words over the validation set (right).

	D	LC	I	VP
LM difference	95.5	91.7	71.8	85.6
Baseline	50.0	45.9	53.4	26.6
Concat	84.9	47.7	84.2	78.6
Static	66.6	65.5	56.6	40.2
Dynamic	74.2	51.1	57.8	56.8
CADec	81.6	58.1	72.2	**80.0**
DocRepair	**91.8**	**80.6**	**86.4**	75.2

Table 6.3: Deixis (D), lexical cohesion (LC), inflection ellipsis (I) and VP ellipsis (VP) accuracy (%). Best scores from translation models only are highlighted.

depend on plurality and formality, are assigned high weights.

Challenge sets While static and dynamic coefficients lead to similar BLEU, using dynamic coefficients often results in better performance on multiple-choice scoring-based challenge sets targeting specific translation phenomena (Table 6.3) [Voita et al. 2019b].[1] We conjecture this likely happens because dynamic coefficients can more narrowly focus on particular subsets of target sentences that benefit from document-level context.

[1]Using the difference of language models scores gives higher accuracy, but they cannot be used in isolation to generate relevant translations.

6.5 Related work

Document-level NMT Neural machine translation may be extended to include extra-sentential information in many ways, as surveyed by Maruf et al. [2021]. The model architecture may be modified, for example by encoding previous source sentences or generated translations and attending to them [Jean et al. 2017; Wang et al. 2017; Voita et al. 2018; Zhang et al. 2018; Miculicich et al. 2018; Maruf and Haffari 2018; Tu et al. 2018]. Otherwise, by simply concatenating multiple sentences together as input, existing model architectures may be used without additional changes [Tiedemann and Scherrer 2017; Junczys-Dowmunt 2019].

Voita et al. [2019b] and Voita et al. [2019a] propose refining the output of a context-agnostic baseline, using a new model trained from either document-level parallel data or from round-trip translated monolingual data. The noisy channel approach similarly uses large-scale monolingual data [Yu et al. 2020] to refine translations, while using arbitrary, and potentially pre-trained, translation or language models, as discussed in Sec. 6.2.

Our approach shares many similarities with the above, but admits a more straightforward generation process. If desired, we could still rerank the beam search output with a channel model, which might improve general translation quality for reasons not necessarily related to context.

Language modelling Language model probabilities have been used to rerank NMT hypotheses [see, e.g., Stahlberg et al. 2019]. Additionally, direct integration of a language model into a translation model, using various fusion techniques, improves generation quality and admits the use of single-pass search algorithms [Gulcehre et al. 2015]. To promote diversity in dialogue systems, model scores may be adjusted by negatively weighing a language model [Li et al. 2015].

6.6 Conclusion

In this chapter, we set to use heterogeneous data sources in an effective and efficient manner for document-level NMT. We reformulated the noisy channel approach [Yu et al. 2020] and end up with a left-to-right log-linear model combining a baseline machine translation model with sentence-level and document-level language models.

To modulate the impact of the language models, we dynamically adapt their coefficients at each time step with a *merging module* taking into account the translation and language models. We observe improvements over a context-agnostic baseline and using dynamic coefficients helps capture document-level linguistic phenomena better.

Future directions include combining our approach with MT models trained on back-translated documents, exploring its applicability to other modalities such as vision and speech, and considering deeper fusion of the models.

6.7 Since the release of this chapter

A similar approach, with fixed coefficients, has been independently proposed by Sugiyama and Yoshinaga [2020]. They interpret the competing language model log-probabilities as the pointwise mutual information between the target sentence and the context. They smooth the language model log-probabilities to improve generation quality.

7 | Conclusion

This dissertation explores the usefulness of neural networks to improve translation in context, where context refers to the nearby source sentences or their translations. Chapter 2 provides background information that is useful to the understanding of the dissertation. We discuss various approaches to machine translation, describe why translating in context is necessary, present previous methods to integrate it, and mention potential advantages of neural machine translation (NMT). Chapter 3, where we present our submission to WMT'15, establishes the potential of neural machine translation, which is used throughout the remaining chapters. Chapter 4 presents the first attempts to integrate context into NMT systems, in particular with architectures where context is encoded and attended to. We evaluate models both in terms of general performance and on a pronoun prediction task. Given the promising, but still not fully satisfactory results of chapter 4, chapter 5 encourages NMT models to put more emphasis on context. We explore different approaches, from either a learning or data perspective. In the second part of this chapter, we transition from using source-side context only to also integrate the previous translations. While we had mostly examined end-to-end systems for context-aware neural machine translation up to now, chapter 6 instead examines the efficient integration of large-scale language models into NMT systems. We reformulate a noisy channel approach for document-level neural machine translation, allowing the use of standard decoding algorithms such as beam search. Overall, this dissertation demonstrates the efficacy of neural networks for larger-context translation.

Our work, in parallel with [Wang et al. 2017], has helped establish the topic of document-

level neural machine translation. Alongside our contributions, many others have pushed the field further. Some of these improvements came from new architectures, especially adapted to the Transformer model [Voita et al. 2018; Zhang et al. 2018; Miculicich et al. 2018]. The development of specific evaluation metrics, also helped improve the understanding and development of larger-context NMT models [Bawden et al. 2018; Voita et al. 2019b; Jwalapuram et al. 2020]. Apart from end-to-end approaches, other strategies, such as post-editing or the use of a noisy channel model, have been proposed to leverage monolingual data [Voita et al. 2019a; Yu et al. 2020]. Additional works are surveyed by Maruf et al. [2021].

Our contributions and those of other researchers have advanced larger-context neural machine translation, but many challenges remain. We have considered a mostly local context spanning a few sentences. To translate longer documents such as books well, it might be necessary to consider longer passages. Efforts are already under way to reduce the quadratic complexity of attention, which could likely help tackle this problem [Tay et al. 2020]. Improved cache models or memory networks could allow retrieving relevant information, and reasoning about it, within a very long document [Tu et al. 2018; Maruf and Haffari 2018]. Appropriate data would also need to be collected. If such available data is scarce, domain adaptation techniques might prove helpful [Chu and Wang 2018].

Given the dependencies between sentences, efficient document translation is also a challenge, in particular for long documents. This problem shares many similarities with non-autoregressive translation, which relaxes the left-to-right generation process within sentences [Gu et al. 2018]. In particular, latent variables [Lee et al. 2020a] could control style, topic, or other document-level properties.

Larger-context neural machine translation also has a lot in common with other tasks, such as summarizing documents or building dialogue systems [Zhang et al. 2019; Lei et al. 2018]. All rely on information beyond the current sentence, although the signal-to-noise ratio within the context might differ. Advances on such tasks may provide insights to make progress on the others.

Bibliography

Alonso, J. A. and Thurmair, G. (2003). The comprendium translator system. In *Ninth Machine Translation Summit*.

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *ICLR'2015, arXiv:1409.0473*.

Barrault, L., Bojar, O., Costa-jussà, M. R., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Koehn, P., Malmasi, S., Monz, C., Müller, M., Pal, S., Post, M., and Zampieri, M. (2019). Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Barron, A. R. (1994). Approximation and estimation bounds for artificial neural networks. *Machine Learning*, 14(1):115–133.

Bawden, R., Sennrich, R., Birch, A., and Haddow, B. (2018). Evaluating discourse phenomena in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1304–1313.

Bengio, Y., Ducharme, R., and Vincent, P. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Bengio, Y. and Sénécal, J.-S. (2008). Adaptive importance sampling to accelerate training of a neural probabilistic language model. *IEEE Trans. Neural Networks*, 19(4):713–722.

Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., Jimeno Yepes, A., Koehn, P., Logacheva, V., Monz, C., Negri, M., Neveol, A., Neves, M., Popel, M., Post, M., Rubino, R., Scarton, C., Specia, L., Turchi, M., Verspoor, K., and Zampieri, M. (2016). Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Boulanger-Lewandowski, N., Bengio, Y., and Vincent, P. (2013). Audio chord recognition with recurrent neural networks. In *ISMIR*.

Brown, P. F., Cocke, J., Pietra, S. A. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., Mercer, R. L., and Roossin, P. S. (1990). A statistical approach to machine translation. *Computational linguistics*, 16(2):79–85.

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Caglayan, O., Aransa, W., Bardet, A., García-Martínez, M., Bougares, F., Barrault, L., Masana, M., Herranz, L., and van de Weijer, J. (2017). Lium-cvc submissions for wmt17 multimodal translation task. In *Proceedings of the Second Conference on Machine Translation*, pages 432–439.

Caglayan, O., Barrault, L., and Bougares, F. (2016). Multimodal attention for neural machine translation. *arXiv preprint arXiv:1609.03976*.

Calixto, I., Liu, Q., and Campbell, N. (2017). Doubly-attentive decoder for multi-modal neural machine translation. *arXiv preprint arXiv:1702.01287*.

Caswell, I., Chelba, C., and Grangier, D. (2019). Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation*.

Chen, S. F. and Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–394.

Chiang, D. (2005). A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, Michigan. Association for Computational Linguistics.

Cho, K. (2015). Natural language understanding with distributed representation. *arXiv preprint arXiv:1511.07916*.

Cho, K., van Merrienboer, B., Gulcehre, C., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*.

Chu, C. and Wang, R. (2018). A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1304–1319, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Crego, J., Kim, J., Klein, G., Rebollo, A., Yang, K., Senellart, J., Akhanov, E., Brunelle, P., Coquard, A., Deng, Y., et al. (2016). Systran's pure neural machine translation systems. *arXiv preprint arXiv:1610.05540*.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314.

Dabre, R., Puzikov, Y., Cromieres, F., and Kurohashi, S. (2016). The kyoto university cross-lingual pronoun translation system. In *Proceedings of the First Conference on Machine Translation*, pages 571–575, Berlin, Germany. Association for Computational Linguistics.

Denkowski, M. and Lavie, A. (2014). Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.

Denkowski, M. and Neubig, G. (2017). Stronger baselines for trustable results in neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 18–27.

Elliott, D. (2018). Adversarial evaluation of multimodal machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2974–2978. Association for Computational Linguistics.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2):179–211.

Eriguchi, A., Hashimoto, K., and Tsuruoka, Y. (2016). Tree-to-sequence attentional neural machine translation. In *ACL*.

Eriguchi, A., Tsuruoka, Y., and Cho, K. (2017). Learning to parse and translate improves neural machine translation. *arXiv preprint arXiv:1702.03525*.

Firat, O., Cho, K., and Bengio, Y. (2016). Multi-way, multilingual neural machine translation with a shared attention mechanism. In *NAACL*.

Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. In *International Conference on Machine Learning*, pages 1243–1252.

Goodfellow, I., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y. (2013). Maxout networks. In *Proceedings of The 30th International Conference on Machine Learning*, pages 1319–1327.

Graves, A. (2012). Sequence transduction with recurrent neural networks. In *Workshop on Representation Learning at ICML*.

Graves, A. (2013). Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850.

Grönroos, S.-A., Huet, B., Kurimo, M., Laaksonen, J., Merialdo, B., Pham, P., Sjöberg, M., Sulubacak, U., Tiedemann, J., Troncy, R., et al. (2018). The memad submission to the wmt18 multimodal translation task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 603–611.

Gu, J., Bradbury, J., Xiong, C., Li, V. O., and Socher, R. (2018). Non-autoregressive neural machine translation. In *ICLR*.

Guillou, L., Hardmeier, C., Nakov, P., Stymne, S., Tiedemann, J., Versley, Y., Cettolo, M., Webber, B., and Popescu-Belis, A. (2016). Findings of the 2016 wmt shared task on cross-lingual pronoun prediction. In *Proceedings of the First Conference on Machine Translation*, pages 525–542, Berlin, Germany. Association for Computational Linguistics.

Gulcehre, C., Firat, O., Xu, K., Cho, K., Barrault, L., Lin, H.-C., Bougares, F., Schwenk, H., and Bengio, Y. (2015). On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

Halliday, M., Hasan, R., Halliday, R., Longman, P., and Quirk, R. (1976). *Cohesion in English*. A Longman paperback. Longman.

Hardmeier, C. and Federico, M. (2010). Modelling pronominal anaphora in statistical machine translation. In *IWSLT (International Workshop on Spoken Language Translation); Paris, France; December 2nd and 3rd, 2010.*, pages 283–289.

Hardmeier, C., Nakov, P., Stymne, S., Tiedemann, J., Versley, Y., and Cettolo, M. (2015). Pronoun-focused mt and cross-lingual pronoun prediction: Findings of the 2015 discomt shared task on pronoun translation. In *Proceedings of the Second Workshop on Discourse in Machine Translation*, pages 1–16, Lisbon, Portugal. Association for Computational Linguistics.

Hardmeier, C., Nivre, J., and Tiedemann, J. (2012). Document-wide decoding for phrase-based statistical machine translation. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning; 12-14 July 2012; Jeju Island, Korea*, pages 1179–1190. Association for Computational Linguistics.

Hassan, H., Aue, A., Chen, C., Chowdhary, V., Clark, J., Federmann, C., Huang, X., Junczys-Dowmunt, M., Lewis, W., Li, M., et al. (2018). Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*.

Heafield, K. (2011). KenLM: faster and smaller language model queries. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, United Kingdom.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. (2020). The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Hsieh, Y.-Y. (2017). The matter of forking consequences: Translating saint-exupéry's little prince.

Huang, Y. (2000). Shalom lappin & elabbas benmamoun (eds.), fragments: studies in ellipsis and gapping. new york: Oxford university press, 1999. pp. xiii 298. *Journal of Linguistics*, 36(3):589–644.

Isozaki, H., Hirao, T., Duh, K., Sudoh, K., and Tsukada, H. (2010). Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952. Association for Computational Linguistics.

Jain, S. and Wallace, B. C. (2019). Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Jean, S., Bapna, A., and Firat, O. (2019). Fill in the blanks: Imputing missing sentences for larger-context neural machine translation. *arXiv preprint arXiv:1910.14075*.

Jean, S. and Cho, K. (2019). Context-aware learning for neural machine translation. *arXiv preprint arXiv:1903.04715*.

Jean, S. and Cho, K. (2020). Log-linear reformulation of the noisy channel model for document-level neural machine translation. In *Proceedings of the Fourth Workshop on Structured Prediction for NLP*, pages 95–101.

Jean, S., Cho, K., Memisevic, R., and Bengio, Y. (2015a). On using very large target vocabulary for neural machine translation. In *Proceedings of ACL*.

Jean, S., Firat, O., Cho, K., Memisevic, R., and Bengio, Y. (2015b). Montreal neural machine translation systems for wmt'15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 134–140, Lisbon, Portugal. Association for Computational Linguistics.

Jean, S., Lauly, S., Firat, O., and Cho, K. (2016). Neural machine translation for cross-lingual pronoun prediction. In *DiscoMT workshop*.

Jean, S., Lauly, S., Firat, O., and Cho, K. (2017). Does neural machine translation benefit from larger context? *arXiv preprint arXiv:1704.05135*.

Ji, Y., Cohn, T., Kong, L., Dyer, C., and Eisenstein, J. (2015). Document context language models. *arXiv preprint arXiv:1511.03962*.

Ji, Y., Haffari, G., and Eisenstein, J. (2016). A latent variable recurrent neural network for discourse relation language models. *arXiv preprint arXiv:1603.01913*.

Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., Boyle, R., luc Cantin, P., Chao, C., Clark, C., Coriell, J., Daley, M., Dau, M., Dean, J., Gelb, B., Ghaemmaghami, T. V., Gottipati, R., Gulland, W., Hagmann, R., Ho, C. R., Hogberg, D., Hu, J., Hundt, R., Hurt, D., Ibarz, J., Jaffey, A., Jaworski, A., Kaplan, A., Khaitan, H., Koch, A., Kumar, N., Lacy, S., Laudon, J., Law, J., Le, D., Leary, C., Liu, Z., Lucke, K., Lundin, A., MacKean, G., Maggiore, A., Mahony, M., Miller, K., Nagarajan, R., Narayanaswami, R., Ni, R., Nix, K., Norrie, T., Omernick, M., Penukonda, N., Phelps, A., and Ross, J. (2017). In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th annual international symposium on computer architecture*.

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 (Volume 2: Shared Task Papers, Day 1)*, pages 225–233, Florence, Italy. Association for Computational Linguistics.

Jwalapuram, P., Rychalska, B., Joty, S., and Basaj, D. (2020). Can your context-aware mt system pass the dip benchmark tests? : Evaluation benchmarks for discourse phenomena in machine translation.

Kalchbrenner, N. and Blunsom, P. (2013). Recurrent continuous translation models. In *Proceedings*

of the ACL Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1700–1709. Association for Computational Linguistics.

Kiefer, J. and Wolfowitz, J. (1952). Stochastic Estimation of the Maximum of a Regression Function. *The Annals of Mathematical Statistics*, 23(3):462 – 466.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Skip-thought vectors. In *NIPS*, pages 3276–3284.

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *1995 international conference on acoustics, speech, and signal processing*, volume 1, pages 181–184. IEEE.

Koehn, P. (2009). *Statistical machine translation*. Cambridge University Press.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (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, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Koehn, P. and Knight, K. (2003). Empirical methods for compound splitting. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*, pages 187–193. Association for Computational Linguistics.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Lin-*

guistics on Human Language Technology-Volume 1, pages 48–54. Association for Computational Linguistics.

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 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Lagarda, A.-L., Alabau, V., Casacuberta, F., Silva, R., and Díaz-de Liaño, E. (2009). Statistical post-editing of a rule-based machine translation system. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 217–220.

Landauer, T. K., Foltz, P. W., and Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, 25(2-3):259–284.

Läubli, S., Sennrich, R., and Volk, M. (2018). Has machine translation achieved human parity? a case for document-level evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4791–4796.

Le Nagard, R. and Koehn, P. (2010). Aiding pronoun translation with co-reference resolution. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 252–261.

Lee, J., Shu, R., and Cho, K. (2020a). Iterative refinement in the continuous space for non-autoregressive neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1006–1015, Online. Association for Computational Linguistics.

Lee, J., Tran, D., Firat, O., and Cho, K. (2020b). On the discrepancy between density estimation and sequence generation. *arXiv preprint arXiv:2002.07233*.

Lei, W., Jin, X., Kan, M.-Y., Ren, Z., He, X., and Yin, D. (2018). Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447, Melbourne, Australia. Association for Computational Linguistics.

Levinson, S. C. (1983). *Pragmatics*. Cambridge Textbooks in Linguistics. Cambridge University Press.

Li, G., Liu, L., Huang, G., Zhu, C., and Zhao, T. (2019). Understanding data augmentation in neural machine translation: Two perspectives towards generalization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5689–5695, Hong Kong, China. Association for Computational Linguistics.

Li, J., Galley, M., Brockett, C., Gao, J., and Dolan, B. (2015). A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Libovickỳ, J. and Helcl, J. (2017). Attention strategies for multi-source sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 196–202.

Lison, P., Tiedemann, J., and Kouylekov, M. (2018). Opensubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

Loáiciga, S., Stymne, S., Nakov, P., Hardmeier, C., Tiedemann, J., Cettolo, M., and Versley, Y. (2017). Findings of the 2017 DiscoMT shared task on cross-lingual pronoun prediction. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, DiscoMT-EMNLP17, Copenhagen, Denmark. Association for Computational Linguistics.

Luong, M.-T., Sutskever, I., Le, Q. V., Vinyals, O., and Zaremba, W. (2015). Addressing the rare word problem in neural machine translation. In *Proceedings of ACL*.

Luotolahti, J., Kanerva, J., and Ginter, F. (2016). Cross-lingual pronoun prediction with deep recurrent neural networks. In *Proceedings of the First Conference on Machine Translation*, pages 596–601, Berlin, Germany. Association for Computational Linguistics.

Ma, M., Li, D., Zhao, K., and Huang, L. (2017). Osu multimodal machine translation system report. In *Proceedings of the Second Conference on Machine Translation*, pages 465–469.

Ma, Z., Edunov, S., and Auli, M. (2021). A comparison of approaches to document-level machine translation. *arXiv preprint arXiv:2101.11040*.

Mansimov, E., Melis, G., and Yu, L. (2020). Capturing document context inside sentence-level neural machine translation models with self-training. *arXiv preprint arXiv:2003.05259*.

Maruf, S. and Haffari, G. (2018). Document context neural machine translation with memory networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1275–1284.

Maruf, S., Saleh, F., and Haffari, G. (2021). A survey on document-level neural machine translation: Methods and evaluation. *ACM Comput. Surv.*, 54(2).

Matthews, P. H. (1991). *Morphology*. Cambridge Textbooks in Linguistics. Cambridge University Press, 2 edition.

Meyer, T. (2015). Discourse-level features for statistical machine translation. Technical report, EPFL.

Michel, P. and Neubig, G. (2018). Extreme adaptation for personalized neural machine translation. In *The 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, Melbourne, Australia.

Miculicich, L., Ram, D., Pappas, N., and Henderson, J. (2018). Document-level neural machine translation with hierarchical attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2947–2954.

Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., and Khudanpur, S. (2010). Recurrent neural network based language model. *INTERSPEECH*, 2:3.

Mitkov, R. (1999). Introduction: special issue on anaphora resolution in machine translation and multilingual nlp. *Machine translation*, pages 159–161.

Mitkov, R. (2002). *Anaphora Resolution*. Routledge.

Montúfar, G. F., Pascanu, R., Cho, K., and Bengio, Y. (2014). On the number of linear regions of deep neural networks. In *NIPS*, pages 2924–2932.

Moon, T. K. (1996). The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60.

Morris, J. and Hirst, G. (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48.

Müller, M., Rios, A., Voita, E., and Sennrich, R. (2018). A large-scale test set for the evaluation of context-aware pronoun translation in neural machine translation. *arXiv preprint arXiv:1810.02268*.

Nadejde, M., Reddy, S., Sennrich, R., Dwojak, T., Junczys-Dowmunt, M., Koehn, P., and Birch, A. (2017). Syntax-aware neural machine translation using ccg. *arXiv preprint arXiv:1702.01147*.

Nirenburg, S. (1989). Knowledge-based machine translation. *Machine Translation*, 4(1):5–24.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics.

Och, F. J., Gildea, D., Khudanpur, S., Sarkar, A., Yamada, K., Fraser, A., Kumar, S., Shen, L., Smith, D. A., Eng, K., et al. (2004). A smorgasbord of features for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 161–168.

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 annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *ICML13*.

Pham, V., Bluche, T., Kermorvant, C., and Louradour, J. (2014). Dropout improves recurrent neural networks for handwriting recognition. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pages 285–290. IEEE.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*.

Saunders, D., Stahlberg, F., and Byrne, B. (2020). Using context in neural machine translation training objectives. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7764–7770, Online. Association for Computational Linguistics.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on*, 45(11):2673–2681.

Sennrich, R., Haddow, B., and Birch, A. (2015). Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Sennrich, R., Haddow, B., and Birch, A. (2016). Improving neural machine translation models

with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96.

Shen, J., Nguyen, P., Wu, Y., Chen, Z., et al. (2019). Lingvo: a modular and scalable framework for sequence-to-sequence modeling.

Shuyo, N. (2010). Language detection library for java.

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 association for machine translation in the Americas*, pages 223–231.

Snover, M., Madnani, N., Dorr, B., and Schwartz, R. (2009). Fluency, adequacy, or HTER? Exploring different human judgments with a tunable MT metric. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 259–268, Athens, Greece. Association for Computational Linguistics.

Somers, H. (1999). Review article: Example-based machine translation. *Machine Translation*, 14(2):113–157.

Specia, L., Frank, S., Sima'an, K., and Elliott, D. (2016). A shared task on multimodal machine translation and crosslingual image description. In *Proceedings of the First Conference on Machine Translation, Berlin, Germany. Association for Computational Linguistics*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Stahlberg, F., Saunders, D., de Gispert, A., and Byrne, B. (2019). Cued@ wmt19: Ewc&lms. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 364–373.

Stymne, S. (2016). Feature exploration for cross-lingual pronoun prediction. In *Proceedings of the First Conference on Machine Translation*, pages 609–615, Berlin, Germany. Association for Computational Linguistics.

Sugiyama, A. and Yoshinaga, N. (2019). Data augmentation using back-translation for context-aware neural machine translation. In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 35–44.

Sugiyama, A. and Yoshinaga, N. (2020). Context-aware decoder for neural machine translation using a target-side document-level language model. *arXiv preprint arXiv:2010.12827*.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In Dasgupta, S. and McAllester, D., editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1139–1147, Atlanta, Georgia, USA. PMLR.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *NIPS*.

Tay, Y., Dehghani, M., Bahri, D., and Metzler, D. (2020). Efficient transformers: A survey.

Tiedemann, J. and Scherrer, Y. (2017). Neural machine translation with extended context. In *DiscoMT workshop*.

Toma, P. (1977). Systran as a multilingual machine translation system. In *Third European Congress on Information Systems and Networks, Overcoming the language barrier*.

Torregrosa, D., Pasricha, N., Masoud, M., Chakravarthi, B. R., Alonso, J., Casas, N., and Arcan, M. (2019). Leveraging rule-based machine translation knowledge for under-resourced neural machine translation models. In *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*, pages 125–133.

Tu, Z., Liu, Y., Shi, S., and Zhang, T. (2018). Learning to remember translation history with a continuous cache. *Transactions of the Association of Computational Linguistics*, 6:407–420.

Unanue, I. J., Esmaili, N., Haffari, G., and Piccardi, M. (2020). Leveraging discourse rewards for document-level neural machine translation. In *COLING*.

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*, pages 5998–6008.

Virpioja, S., Smit, P., Grönroos, S.-A., Kurimo, M., et al. (2013). Morfessor 2.0: Python implementation and extensions for morfessor baseline.

Voita, E., Sennrich, R., and Titov, I. (2019a). Context-aware monolingual repair for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing*, Hong Kong, China. Association for Computational Linguistics.

Voita, E., Sennrich, R., and Titov, I. (2019b). When a Good Translation is Wrong in Context: Context-Aware Machine Translation Improves on Deixis, Ellipsis, and Lexical Cohesion. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics.

Voita, E., Serdyukov, P., Sennrich, R., and Titov, I. (2018). Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1264–1274.

Wang, L., Tu, Z., Way, A., and Liu, Q. (2017). Exploiting cross-sentence context for neural machine translation. In *EMNLP*.

Wang, T. and Cho, K. (2016). Larger-context language modelling with recurrent neural network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1319–1329.

Webber, B., Carpuat, M., Popescu-Belis, A., and Hardmeier, C., editors (2015). *Proceedings of the Second Workshop on Discourse in Machine Translation*. Association for Computational Linguistics, Lisbon, Portugal.

Weissenborn, J. and Klein, W., editors (1982). *Here and There: Cross-linguistic Studies on Deixis and Demonstration*. John Benjamins.

Welleck, S., Kulikov, I., Roller, S., Dinan, E., Cho, K., and Weston, J. (2020). Neural text generation with unlikelihood training. In *International Conference on Learning Representations*.

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Xiong, D., Ben, G., Zhang, M., Lv, Y., and Liu, Q. (2013). Modeling lexical cohesion for document-level machine translation. In *IJCAI*, pages 2183–2189.

Xiong, H., He, Z., Wu, H., and Wang, H. (2019). Modeling coherence for discourse neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7338–7345.

Yamada, K. and Knight, K. (2001). A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 523–530, Toulouse, France. Association for Computational Linguistics.

Yu, L., Sartran, L., Stokowiec, W., Ling, W., Kong, L., Blunsom, P., and Dyer, C. (2020). Better document-level machine translation with bayes' rule. *Transactions of the Association for Computational Linguistics*, 8:346–360.

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zhang, J., Luan, H., Sun, M., Zhai, F., Xu, J., Zhang, M., and Liu, Y. (2018). Improving the transformer translation model with document-level context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 533–542.

Zhang, X., Wei, F., and Zhou, M. (2019). HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5059–5069, Florence, Italy. Association for Computational Linguistics.

Zheng, Z., Huang, S., Sun, Z., Weng, R., Dai, X.-Y., and Chen, J. (2018). Learning to discriminate noises for incorporating external information in neural machine translation. *arXiv preprint arXiv:1810.10317*.

Zoph, B. and Knight, K. (2016). Multi-source neural translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 30–34, San Diego, California. Association for Computational Linguistics.