NEURAL MACHINE TRANSLATION

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Minh-Thang Luong
July 2016

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Christopher D. Manning) Principal Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Dan Jurafsky)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Andrew Ng)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Quoc V. Le)

Approved for the Stanford University Committee on Graduate Studies

Contents

List of Tables

List of Figures

Chapter 1

Introduction

> The Babel fish is small, yellow, leech-like, and probably the oddest thing in the universe. It feeds on brainwave energy ... if you stick a Babel fish in your ear, you can instantly understand anything in any form of language.
>
> *The Hitchhiker's Guide to the Galaxy*. Douglas Adams.

Human languages are diverse and rich in categories with about 6000 to 7000 languages spoken worldwide.[1] As civilization advances, the need for seamless communication and understanding across languages becomes more and more crucial. Machine translation (MT), the task of teaching machines to learn to translate automatically across languages, as a result, is an important research area. MT has a long history [17] from the original phiosophical ideas of universal languages in the seventeen century to the first practical instances of MT in the twentieth century, e.g., one proposal by Weaver [50]. Despite several excitement moments that led to hopes that MT will be solved "very soon", e.g., the 701 translator[2] developed by scientists at George Town and IBM in the 1950s or a simple vector-space transformation technique[3] proposed by Google researchers at the beginning of the twenty-first century, MT remains to be an extremely challenging problem.[4] To understand why MT is difficult, let us trace through one "evolution" path of MT which crosses

[1] http://www.linguisticsociety.org/content/how-many-languages-are-there-world
[2] http://www-03.ibm.com/ibm/history/exhibits/701/701_translator.html
[3] https://www.technologyreview.com/s/519581/how-google-converted-language-transl
[4] http://www.huffingtonpost.com/nataly-kelly/why-machines-alone-cannot-translati

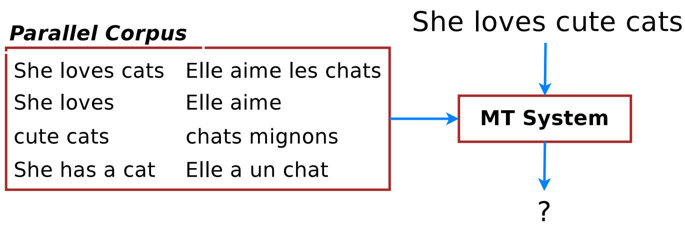

Figure 1.1: **Machine translation** (MT) – a general setup of MT. Systems build translation models from parallel corpora to translate new unseen sentences, e.g., "She loves cute cats".

through techniques that are used extensively in commercial MT systems.

1.1 Machine Translation Development

Modern statistical MT started out with a seminal work by IBM scientists [5]. The proposed technique requires minimal linguistic content and only needs a *parallel corpus*, i.e., a set of pairs of sentences that are translations of one another, to train machine learning algorithms to tackle the translation problem. Such a language-independent setup is illustrated in Figure 1.1 and remains to be the general approach for nowadays MT systems. For over twenty years since the IBM seminal paper, approaches in MT such as [6, 7, 10, 21, 22, 23, 37], are, by and large, similar according to the following two-stage process (see Figure 1.2). First, source sentences are broken into chunks which can be translated in isolation by looking up a "dictionary", or more formally a *translation model*. Translated target words and phrases are then put together to form coherent and natural-sounding sentences by consulting a *language model* (LM) on which sequences of words, i.e., n-*grams*, are likely to go with one another.

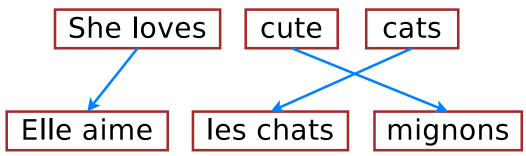

Figure 1.2: **Phrase-based machine translation** (MT) – example of how phrase-based MT systems translate a source sentence "She loves cute cats" into a target sentence "Elle aime les chats mignons": sentences are split into chunks and phrases are translated.

The aforementioned approach, while has been successfully deployed in many commercial systems, does not work very well and suffers from the following two major drawbacks. First, translation decisions are *locally determined* as we translate phrase-by-phrase and long-distance dependencies are often ignored. Second, it is slightly "strange" that language models (LMs), despite being a key component in the MT pipeline, utilize context information that is both short, consisting of only a handful of previous words, and target-only, never looking at the source words. These shortcomings in LMs gives rise to a new wave of *hybrid* systems which aim to empower phrase-based MT with neural network components, most notably neural probabilistic language models (NPLMs).

NPLMs were first proposed by Bengio et al. [4] as a way to combat the "curse" of dimensionality suffered by traditional LMs. In traditional LMs, one has to explicitly store and handle all possible n-grams occurred in a training corpus, the number of which quickly becomes enormous. As a result, existing MT systems often limit themselves to use only short, e.g., 5-gram, LMs [14], which capture little context and cannot generalize well to unseen n-grams. NPLMs address these concerns by using distributed representations of words and not having to explicitly store all enumerations of words. As a result, many MT systems, [26, 40, 49], inter alia, start adopting NPLMs alongside with traditional LMs. To make NPLMs even more powerful, recent work [1, 9, 41, 43] propose to condition on source words beside the target context to lower uncertainty in predicting next words (see Figure 1.3).[5]

These hybrid MT systems with NPLM components, while having addressed shortcomings of traditional phrase-based MT, still translate locally and fail to capture long-range dependencies. For example, in Figure 1.3, the source-conditioned NPLM does not see the word "stroll", or any other words outside of its fixed context windows, which can be useful in deciding that the next word should be "bank" as in "river bank" rather "financial bank". More problematically, the entire MT pipeline is already complex with different components needed to be tuned separatedly, e.g., translation models, language models, reordering models, etc.; now, it becomes even worse as different neural components are incorporated. Neural Machine Translation to the rescue!

[5] In [9], the authors have constructed a model that conditions on 3 target words and 11 source words, effectively building a 15-gram LM.

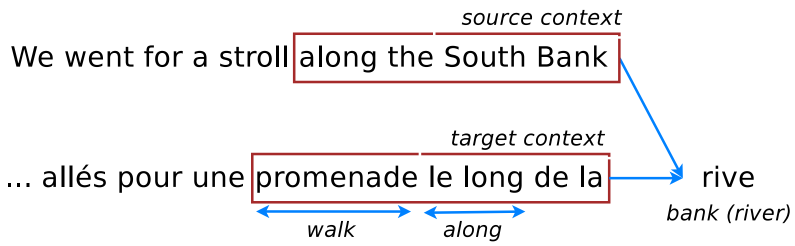

Figure 1.3: **Source-conditioned neural probabilistic language models** (NPLMs) – example of a source-conditioned NPLM proposed by Devlin et al. [9]. To evaluate a how likely a next word "rive" is, the model not only relies on previous target words (context) "promenade le long de la" as in traditional NPLMs [4], but also utilizes source context "along the South Bank" to lower uncertainty in its prediction.

Neural Machine Translation (NMT) is a new approach to translating text from one language into another that captures long-range dependencies in sentences and generalizes better to unseen texts. The core of NMT is a single deep neural network with hundreds of millions of neurons that learn to directly map source sentences to target sentences [8, 19, 47]. This is often referred as the sequence-to-sequence or encoder-decoder approach.[6] NMT is appealing since it is conceptually simple and can be trained end-to-end. NMT translates as follows: an *encoder* reads through the given source words one by one until the end, and then, a *decoder* starts emitting one target word at a time until a special end-of-sentence symbol is produced. We illustrate this process in Figure 1.4.

Such simplicity leads to several advantages. NMT requires minimal domain knowledge: it only assumes access to sequences of source and target words as training data and learns to directly map one into another. NMT beam-search decoders that generate words from left to right can be easily implemented, unlike the highly intricate decoders in standard MT [21]. Lastly, the use of recurrent neural networks (RNNs) allow NMT to generalize well to very long word sequences while not having to explicitly store any gigantic phrase tables or language models as in the case of standard MT.

[6]Forcada and Neco [13] wrote the very first paper on sequence-to-sequence models for translation!

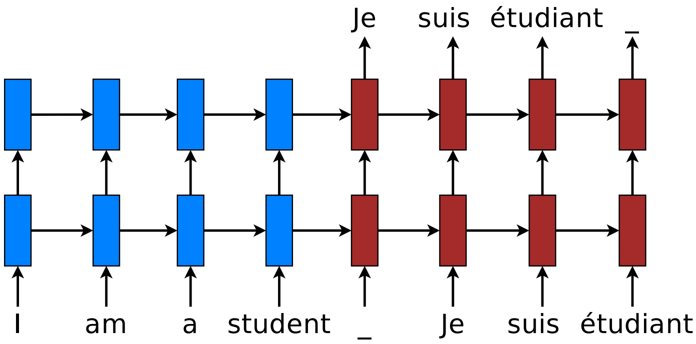

Figure 1.4: **Neural machine translation** – example of a deep recurrent architecture proposed by Sutskever et al. [47] for translating a source sentence "I am a student" into a target sentence "Je suis étudiant". Here, "_" marks the end of a sentence.

1.2 Thesis Outline

Despite all the aforementioned advantages and potentials, the early NMT architecture [8, 47] still has many drawbacks. In this thesis, I will highlight three problems pertaining to the existing NMT model, namely the *vocabulary size*, the *sentence length*, and the *language complexity* issues. Each chapter is devoted to solving each of these problems in which I will describe how I have pushed the limits of NMT, making it applicable to a wide variety of languages with state-of-the-art performance such as English-French [28], English-German [24, 27], and English-Czech [25]. Towards the *future* of NMT, I answer two questions: (1) whether we can improve translation by jointly learning from a wide variety of sequence-to-sequence tasks such as parsing, image caption generation, and auto-encoders or skip-thought vectors [29]; and (2) whether we can compress NMT for mobile devices [42]. In brief, this thesis is organized as follows. I start off by providing background knowledge on RNN and NMT in Chapter 2. The aforementioned three problems and approaches for NMT future are detailed in Chapters 3, 4, 5, and 6 respectively, which we will go through one by one next. Chapter 7 wraps up and discusses remaining challenges in NMT research.

Copy Mechanisms

A significant weakness in conventional NMT systems is their inability to correctly translate very rare words: end-to-end NMTs tend to have relatively small vocabularies with a single

<unk> symbol that represents every possible out-of-vocabulary (OOV) word. In Chapter 3, we propose simple and effective techniques to address this *vocabulary size* problem through teaching NMT to "copy" words from source to target. Specifically, we train an NMT system on data that is augmented by the output of a word alignment algorithm, allowing the NMT system to emit, for each OOV word in the target sentence, the position of its corresponding word in the source sentence. This information is later utilized in a post-processing step that translates every OOV word using a dictionary. Our experiments on the WMT'14 English to French translation task show that this method provides a substantial improvement of up to 2.8 BLEU points over an equivalent NMT system that does not use this technique. With 37.5 BLEU points, our NMT system is the first to surpass the best result achieved on a WMT'14 contest task.

Attention Mechanisms

While NMT can translate well for short- and medium-length sentences, it has a hard time dealing with long sentences. An attentional mechanism was proposed by Bahdanau et al. [2] to address that *sentence length* problem by selectively focusing on parts of the source sentence during translation. However, there has been little work exploring useful architectures for attention-based NMT. Chapter 4 examines two simple and effective classes of attentional mechanism: a *global* approach which always attends to all source words and a *local* one that only looks at a subset of source words at a time. We demonstrate the effectiveness of both approaches on the WMT translation tasks between English and German in both directions. With local attention, we achieve a significant gain of 5.0 BLEU points over non-attentional systems that already incorporate known techniques such as dropout. Our ensemble model using different attention architectures yields a new state-of-the-art result in the WMT'15 English to German translation task with 25.9 BLEU points, an improvement of 1.0 BLEU points over the existing best system backed by NMT and an n-gram reranker.

Hybrid Models

Nearly all previous NMT work has used quite restricted vocabularies, perhaps with a subsequent method to patch in unknown words such as the copy mechanisms mentioned earlier.

While effective, the copy mechanims cannot deal with all the complexity of human languages such as rich morphology, neologisms, and informal spellings. Chapter 5 presents a novel word-character solution to that *language complexity* problem towards achieving open vocabulary NMT. We build hybrid systems that translate mostly at the *word* level and consult the *character* components for rare words. Our character-level recurrent neural networks compute source word representations and recover unknown target words when needed. The twofold advantage of such a hybrid approach is that it is much faster and easier to train than character-based ones; at the same time, it never produces unknown words as in the case of word-based models. On the WMT'15 English to Czech translation task, this hybrid approach offers an addition boost of +$2.1-11.4$ BLEU points over models that already handle unknown words. Our best system achieves a new state-of-the-art result with 20.7 BLEU score. We demonstrate that our character models can successfully learn to not only generate well-formed words for Czech, a highly-inflected language with a very complex vocabulary, but also build correct representations for English source words.

NMT Future

Chapter 6 answers the two aforementioned questions for the future of NMT: whether we can utilize other tasks to improve translation and whether we can compress NMT models.

For the first question, we examine three multi-task learning (MTL) settings for sequence to sequence models: (a) the *one-to-many* setting – where the encoder is shared between several tasks such as machine translation and syntactic parsing, (b) the *many-to-one* setting – useful when only the decoder can be shared, as in the case of translation and image caption generation, and (c) the *many-to-many* setting – where multiple encoders and decoders are shared, which is the case with unsupervised objectives and translation. Our results show that training on a small amount of parsing and image caption data can improve the translation quality between English and German by up to 1.5 BLEU points over strong single-task baselines on the WMT benchmarks. Rather surprisingly, we have established a new *state-of-the-art* result in constituent parsing with 93.0 F_1 by utilizing translation data. Lastly, we reveal interesting properties of the two unsupervised learning objectives, autoencoder and skip-thought, in the MTL context: autoencoder helps less in terms of perplexities but more on BLEU scores compared to skip-thought.

For the second question, we examine three simple magnitude-based pruning schemes to compress NMT models, namely *class-blind*, *class-uniform*, and *class-distribution*, which differ in terms of how pruning thresholds are computed for the different classes of weights in the NMT architecture. We demonstrate the efficacy of weight pruning as a compression technique for a state-of-the-art NMT system. We show that an NMT model with over 200 million parameters can be pruned by 40% with very little performance loss as measured on the WMT'14 English-German translation task. This sheds light on the distribution of redundancy in the NMT architecture. Our main result is that with *retraining*, we can recover and even surpass the original performance with an 80%-pruned model.

Chapter 2

Background

For Neural Machine Translation, it all started from language modeling.

Thang Luong.

Language modeling plays an indispensable role in ensuring that machine translation systems produce fluent target sentences and has always been an active area of research. Despite much effort in improving traditional n-gram language models [12, 14, 15, 38, 39, 44, 48], traditional LMs inherently can only handle short contexts of a few words. Approaches to building neural probabilistic language models (NPLMs) using feed-forward networks such as those initiated by Bengio et al. [4] and enhanced by others [3, 34, 35, 36] have addressed that drawback to model longer contexts. Still, NPLMs can only capture fixed-length contexts and is incapable of handling variable-length sequences, which is the case for sentences. Recurrent neural networks (RNNs) come in handy as a powerful and expressive architecture to handle sequential data and have successfully been applied to the language modeling task [31, 32, 33]. By viewing RNNs as generative models [46] that can produce texts and by pushing another step towards conditioning RNNs on source sentences, recent works [8, 19, 47] have started a new line of resesearch in machine translation, namely Neural Machine Translation (NMT). NMT is technically a source-conditioned NPLM that can be trained end-to-end.

In this chapter, we provide background knowledge on two main topics, RNN and NMT.

We first go through the basics of RNNs, explaining how they can be used to model sentences. Then we delve into details of one particular type of RNNs, the Long Short-term Memory, that makes training RNNs easier. Given RNNs as a building block, we discuss NMT together with tips and tricks for better training and testing NMT.

2.1 Recurrent Neural Network

Recurrent Neural Network (RNN) [11] is the model that takes as input a sequence of vectors $x_1, x_2, ..., x_n \in \mathbb{R}^d$ and processes them in the order they are given. RNN maintains a sequence of hidden states $h_1, h_2, ..., h_n$, which it computes by a recurrent function.

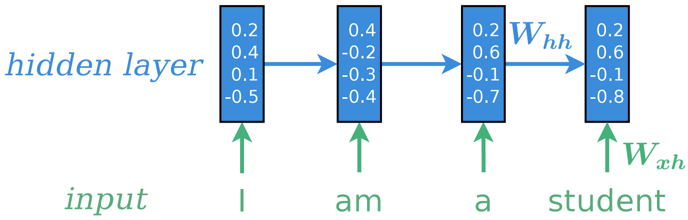

Figure 2.1: **Recurrent neural networks** – example of a recurrent neural network that processes a sequence of input words “I am a student” to build up hidden representations as input symbols are consumed. The recurrent $\boldsymbol{W_{hh}}$ and feed-forward $\boldsymbol{W_{xh}}$ parameters are shared across timesteps.

$$\boldsymbol{h_t} = \sigma\left(\boldsymbol{T}_{d\times 2d}\begin{bmatrix}\boldsymbol{x_t}\\ \boldsymbol{h_{t-1}}\end{bmatrix}\right) \tag{2.1}$$

Here, $\boldsymbol{T}_{d\times 2d}$ is the RNN parameter matrix, which decomposes into “input” connections $\boldsymbol{W}_x h$ and “recurrent” connections $\boldsymbol{W}_h h$.

$$\boldsymbol{T}_{d\times 2d} = [\boldsymbol{W_{xh}} \boldsymbol{W_{hh}}] \tag{2.2}$$

For more details on RNNs, we refer readers to the following resources [20, 30, 45].

2.1.1 Vanishing Gradient Problem

$$\boldsymbol{h}_t = \sigma\left(\boldsymbol{W_{xh}}\boldsymbol{x}_t + \boldsymbol{W_{hh}}\boldsymbol{h}_{t-1}\right) \tag{2.3}$$

$$\frac{\partial \boldsymbol{h}_t}{\partial \boldsymbol{h}_{t-1}} = \operatorname{diag}\left(\sigma'(\ldots)\right)\boldsymbol{W_{hh}}^\top \tag{2.4}$$

$$\begin{aligned} \left\|\frac{\partial \boldsymbol{h}_t}{\partial \boldsymbol{h}_{t-1}}\right\| &\leq \left\|\operatorname{diag}\left(\sigma'(\ldots)\right)\right\| \left\|\boldsymbol{W_{hh}}^\top\right\| &(2.5)\\ &\leq \gamma\lambda_1 &(2.6) \end{aligned}$$

$$\left\|\frac{\partial \boldsymbol{h}_t}{\partial \boldsymbol{h}_{t-k}}\right\| \leq (\gamma\lambda_1)^k \to 0 \quad \text{if } \lambda_1 < \frac{1}{\gamma} \tag{2.7}$$

$$\frac{\partial \boldsymbol{c}_t}{\partial \boldsymbol{c}_{t-1}} = \boldsymbol{I} \tag{2.8}$$

2.1.2 Long Short-Term Memory

We use the formulation of [51]. For a single LSTM block at layer l and time t, the new hidden state $\boldsymbol{h_t^l}$ and memory cell $\boldsymbol{c_t^l}$ are calculated from $\boldsymbol{h_t^{l-1}}$, $\boldsymbol{h_{t-1}^l}$ and $\boldsymbol{c_{t-1}^l}$ like so:

$$\begin{pmatrix} \boldsymbol{i_t^l} \\ \boldsymbol{f_t^l} \\ \boldsymbol{o_t^l} \\ \boldsymbol{\hat{h}_t^l} \end{pmatrix} = \begin{pmatrix} \mathrm{sigm} \\ \mathrm{sigm} \\ \mathrm{sigm} \\ \tanh \end{pmatrix} \boldsymbol{T}_{4n\times 2n} \begin{bmatrix} \boldsymbol{h_t^{l-1}} \\ \boldsymbol{h_{t-1}^l} \end{bmatrix} \tag{2.9}$$

$$\boldsymbol{c_t^l} = \boldsymbol{f_t^l} \circ \boldsymbol{c_{t-1}^l} + \boldsymbol{i_t^l} \circ \boldsymbol{\hat{h}_t^l} \tag{2.10}$$

$$\boldsymbol{h_t^l} = \boldsymbol{o_t^l} \circ \tanh(\boldsymbol{c_t^l}) \tag{2.11}$$

where sigm and $\tanh$ are applied element-wise, $\circ$ denotes element-wise multiplication, and $\boldsymbol{T}_{4n\times 2n}$ is a $4n \times 2n$ matrix of weights that depends on l but not t.[1] If $l = 1$ then $\boldsymbol{h_t^{l-1}}$ is the input vector x_t. If $t = 1$ then $\boldsymbol{h_{t-1}^l}$ and $\boldsymbol{c_{t-1}^l}$ are taken to be zero.

An LSTM block at layer $l \in \{1, \ldots L\}$ and time $t \in \{1, \ldots T\}$ consists of:

- The hidden state $\boldsymbol{h_t^l} \in \mathbb{R}^n$
- The memory cell $\boldsymbol{c_t^l} \in \mathbb{R}^n$
- The input gate $\boldsymbol{i_t^l} \in [0, 1]^n$
- The forget gate $\boldsymbol{f_t^l} \in [0, 1]^n$
- The output gate $\boldsymbol{o_t^l} \in [0, 1]^n$
- The input modulation gate $\boldsymbol{\hat{h}_t^l} \in [0, 1]^n$

We call n the LSTM block size.

2.1.3 LSTM Backpropation

$\boldsymbol{\delta_{c^{(2)}}}$

[1]Note: Sometimes these equations are written omitting the superscript l and writing $\boldsymbol{h_t^{l-1}}$ as x_t, but for the purposes of deriving the back-propagation equations, we need to refer to the layer l explicitly.

$\boldsymbol{\delta}_{\boldsymbol{h}^{(2)}}$
$\boldsymbol{\delta}_{\boldsymbol{c}^{(1)}}$
$\boldsymbol{\delta}_{\boldsymbol{h}^{(1)}}$
$\boldsymbol{\delta}_{\boldsymbol{c}} \mathrel{+}= \boldsymbol{\delta}_{\boldsymbol{h}} \boldsymbol{o}_{\boldsymbol{t}} \tanh'(\boldsymbol{c}_{\boldsymbol{t}})$
$\boldsymbol{\delta}_{\boldsymbol{c}} = \boldsymbol{\delta}_{\boldsymbol{c}} \circ \boldsymbol{f}_{\boldsymbol{t}}$
$\boldsymbol{\delta}_{\boldsymbol{h}} \mathrel{+}=$ upper grad

2.2 Neural Machine Translation

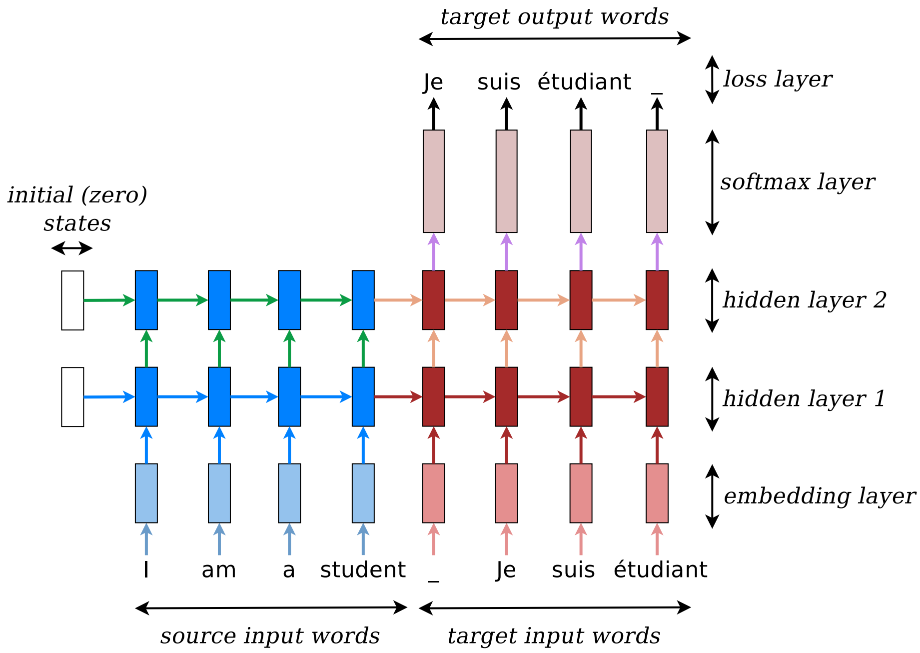

Figure 2.2: **Neural machine translation** – example of a deep recurrent architecture proposed by Sutskever et al. [47] for translating a source sentence "I am a student" into a target sentence "Je suis étudiant". Here, "_" marks the end of a sentence.

Neural machine translation aims to directly model the conditional probability $p(y|x)$ of translating a source sentence, $x_1, \ldots, x_n$, to a target sentence, $y_1, \ldots, y_m$. It accomplishes such goal through the *encoder-decoder* framework [8, 47]. The *encoder* computes a representation $\boldsymbol{s}$ for each source sentence. Based on that source representation, the *decoder* generates a translation, one target word at a time, and hence, decomposes the conditional probability as:

$$\log p(y|x) = \sum\nolimits_{j=1}^{m} \log p\left(y_j|y_{<j}, x, \boldsymbol{s}\right) \tag{2.12}$$

A natural choice to model such a decomposition in the decoder is to use a recurrent neural network (RNN) architecture, which most of the recent NMT work have in common. They, however, differ in terms of the RNN architectures used and how the encoder computes the source representation $\boldsymbol{s}$.

Kalchbrenner and Blunsom [19] used an RNN with the vanilla RNN unit for the decoder and a convolutional neural network for encoding the source. On the other hand, Sutskever et al. [47] and Luong et al. [27, 28] built deep RNNs with the Long Short-Term Memory (LSTM) unit [16] for both the encoder and the decoder. Cho et al., [8], Bahdanau et al., [2], and Jean et al. [18**?**] all adopted an LSTM-inspired hidden unit, the gated recurrent unit (GRU), and used bidirectional RNNs for the encoder.

In more details, considering the top recurrent layer in a deep RNN architecture, one can compute the probability of decoding each target word y_j as:

$$p\left(y_j|y_{<j}, x, \boldsymbol{s}\right) = \operatorname{softmax}\left(\boldsymbol{h}_j\right) \tag{2.13}$$

with $\boldsymbol{h}_j$ being the current target hidden state computed as:

$$\boldsymbol{h}_j = f(\boldsymbol{h}_{j-1}, y_{j-1}, \boldsymbol{s}) \tag{2.14}$$

Here, f derives the current state given the previous state $\boldsymbol{h}_{j-1}$, the current input (often the previous word y_{t-1}), and optionally, the source representation $\boldsymbol{s}$. f can be a vanilla RNN unit, a GRU, or an LSTM. The early NMT approach [8, 19, 28, 47] uses the last source hidden state $\boldsymbol{s} = \bar{\boldsymbol{h}}_n$ once to initialize the decoder hidden state and sets $\boldsymbol{s} = [\,]$ in Eq. (2.14).

2.2.1 Training

The training objective is formulated as follows:

$$J = \sum\nolimits_{(x,y)\in\mathbb{D}} -\log p(y|x) \tag{2.15}$$

with $\mathbb{D}$ being our parallel training corpus.

mention bucketing and batching

2.2.2 Testing

mention beam-search, ensemble

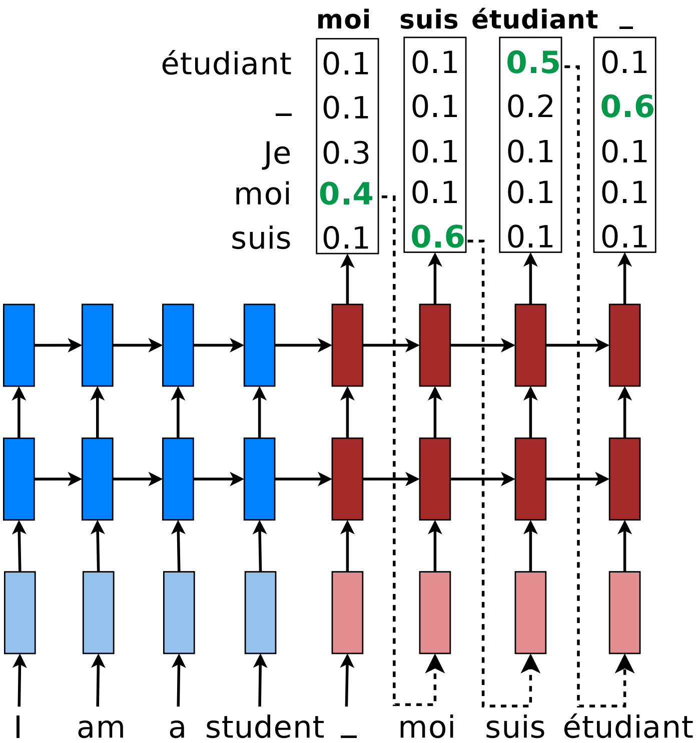

Figure 2.3: **Neural machine translation** – example of a deep recurrent architecture proposed by Sutskever et al. [47] for translating a source sentence “I am a student” into a target sentence “Je suis étudiant”. Here, “_” marks the end of a sentence.

Chapter 3

Copy Mechanisms

Chapter 4

Attention Mechanisms

Chapter 5

Hybrid Models

Chapter 6

NMT Future

Chapter 7

Conclusion

Bibliography

[1] Michael Auli, Michel Galley, Chris Quirk, and Geoffrey Zweig. Joint language and translation modeling with recurrent neural networks. In *ACL*, 2013.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

[3] Yoshua Bengio and Jean-Sébastien Senécal. Adaptive importance sampling to accelerate training of a neural probabilistic language model. *IEEE Trans. Neural Networks*, 19(4):713–722, 2008.

[4] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *JMLR*, 3:1137–1155, 2003.

[5] Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 06 1993.

[6] Daniel Cer, Michel Galley, Daniel Jurafsky, and Christopher D. Manning. Phrasal: A statistical machine translation toolkit for exploring new model features. In *ACL, Demonstration Session*, 2010.

[7] David Chiang. Hierarchical phrase-based translation. *Computational Linguistics*, 33 (2):201–228, 2007.

[8] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, 2014.

[9] Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. Fast and robust neural network joint models for statistical machine translation. In *ACL*, 2014.

[10] Chris Dyer, Jonathan Weese, Hendra Setiawan, Adam Lopez, Ferhan Ture, Vladimir Eidelman, Juri Ganitkevitch, Phil Blunsom, and Philip Resnik. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *ACL, Demonstration Session*, 2010.

[11] Jeffrey L. Elman. Finding structure in time. In *Cognitive Science*, 1990.

[12] Marcello Federico, Nicola Bertoldi, and Mauro Cettolo. IRSTLM: an open source toolkit for handling large scale language models. In *Interspeech*, 2008.

[13] Mikel L. Forcada and Ramón Neco. Recursive hetero-associative memories for translation. *Biological and Artificial Computation: From Neuroscience to Technology*, pages 453–462, 1997.

[14] Kenneth Heafield. KenLM: faster and smaller language model queries. In *WMT*, 2011.

[15] Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. Scalable modified Kneser-Ney language model estimation. In *ACL*, 2013.

[16] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[17] W. John Hutchins. Machine translation: A concise history, 2007.

[18] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation. In *ACL*, 2015.

[19] Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. In *EMNLP*, 2013.

[20] Andrej Karpathy. The unreasonable effectiveness of recurrent neural networks. `http://karpathy.github.io/2015/05/21/rnn-effectiveness/`, 2015. Accessed: 2016-07-05.

[21] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *NAACL*, 2003.

[22] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. Moses: Open source toolkit for statistical machine translation. In *ACL, Demonstration Session*, 2007.

[23] Percy Liang, Alexandre Bouchard-Côté, Dan Klein, and Ben Taskar. An end-to-end discriminative approach to machine translation. In *ACL*, 2006.

[24] Minh-Thang Luong and Christopher D. Manning. Stanford neural machine translation systems for spoken language domain. In *IWSLT*, 2015.

[25] Minh-Thang Luong and Christopher D. Manning. Achieving open vocabulary neural machine translation with hybrid word-character models. In *ACL*, 2016.

[26] Minh-Thang Luong, Michael Kayser, and Christopher D. Manning. Deep neural language models for machine translation. In *CoNLL*, 2015.

[27] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *EMNLP*, 2015.

[28] Minh-Thang Luong, Ilya Sutskever, Quoc V. Le, Oriol Vinyals, and Wojciech Zaremba. Addressing the rare word problem in neural machine translation. In *ACL*, 2015.

[29] Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. Multi-task sequence to sequence learning. In *ICLR*, 2016.

[30] Tomáš Mikolov. *Statistical Language Models Based on Neural Networks*. PhD thesis, Brno University of Technology, 2012.

[31] Tomáš Mikolov and Geoffrey Zweig. Context dependent recurrent neural network language model. In *SLT*, 2012.

[32] Tomáš Mikolov, Martin Karafit, Lukas Burget, Jan Cernock, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, 2010.

[33] Tomáš Mikolov, Stefan Kombrink, Lukas Burget, Jan Cernock, and Sanjeev Khudanpur. Extensions of recurrent neural network language model. In *ICASSP*, 2011.

[34] Andriy Mnih and Geoffrey Hinton. A scalable hierarchical distributed language model. In *NIPS*, 2009.

[35] Andriy Mnih and Yee Whye Teh. A fast and simple algorithm for training neural probabilistic language models. In *ICML*, 2012.

[36] Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. In *AISTATS*, 2005.

[37] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

[38] Adam Pauls and Dan Klein. Faster and smaller n-gram language models. In *ACL*, 2011.

[39] Ronald Rosenfeld. Two decades of statistical language modeling: Where do we go from here? In *IEEE*, volume 88, pages 1270–1278, 2000.

[40] Holger Schwenk. Continuous space language models. *Computer Speech and Languages*, 21(3):492–518, 2007.

[41] Holger Schwenk. Continuous space translation models for phrase-based statistical machine translation. In *COLING*, 2012.

[42] Abigail See, Minh-Thang Luong, and Christopher D. Manning. Compression of neural machine translation models via pruning. In *CoNLL*, 2016.

[43] Le Hai Son, Alexandre Allauzen, and Franois Yvon. Continuous space translation models with neural networks. In *NAACL-HLT*, 2012.

[44] Andreas Stolcke. SRILM – an extensible language modeling toolkit. In *ICSLP*, 2002.

[45] Ilya Sutskever. *Training Recurrent Neural Networks*. PhD thesis, University of Toronto, 2012.

[46] Ilya Sutskever, James Martens, and Geoffrey Hinton. Generating text with recurrent neural networks. In *ICML*, 2011.

[47] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

[48] Yee Whye Teh. A hierarchical Bayesian language model based on Pitman-Yor processes. In *ACL*, 2006.

[49] Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. Decoding with large-scale neural language models improves translation. In *EMNLP*, 2013.

[50] Warren Weaver. Translation. In William N. Locke and A. Donald Boothe, editors, *Machine Translation of Languages*, pages 15–23. MIT Press, Cambridge, MA, 1949. Reprinted from a memorandum written by Weaver in 1949.

[51] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.