CONVOLUTIONAL NEURAL NETWORK LANGUAGE MODELS

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

Ngoc-Quan Pham
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 Master of Science.

(Marco Baroni) 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 Master of Science.

(Gemma Boleda)

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 Master of Science.

(German Kruzhevsky)

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 [9] 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 [30]. 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-translati

[4] http://www.huffingtonpost.com/nataly-kelly/why-machines-alone-cannot-translation_b_4570018.html

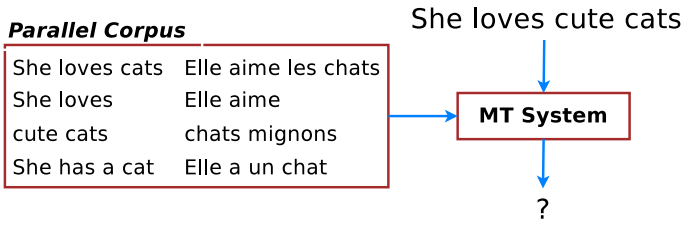

Figure 1.1: **Machine translation** (MT) – a general setup of MT. Systems build translation models from parallel corpora to translate new unseen sentences, e.g., "She loves cute cats".

through techniques that are used extensively in commercial MT systems.

Modern statistical MT started out with a seminal work by IBM scientists [2]. 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 [3, 4, 7, 15, 16, 17, 22], 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.

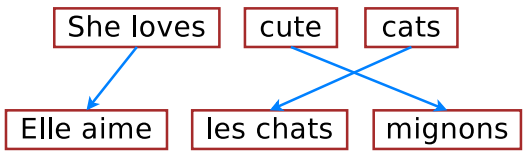

Figure 1.2: **Phrase-based machine translation** (MT) – example of how phrase-based MT systems translate a source sentence "She loves cute cats" into a target sentence "Elle aime les chats mignons": sentences are split into chunks and phrases are translated.

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 language models (NLMs).

NLMs were first proposed by Bengio et al. [1] 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 [8], which capture little context and cannot generalize well to unseen n-grams. NLMs 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, [19, 25, 29], inter alia, start adopting NLMs alongside with traditional LMs. To make NLMs even more powerful, recent work [6, 26] propose to condition on source words beside the target context to lower uncertainty in predicting next words (see Figure 1.3).[5]

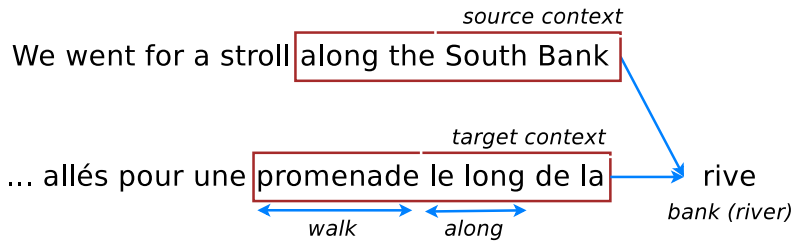

Figure 1.3: **Source-conditioned neural language model** (NLM) – example of a source-conditioned NLM proposed by Devlin et al. [6]. To evaluate a how likely a next word "rive" is, the model not only relies on previous target words (context) "promenade le long de la" as in traditional NLMs [1], but also utilizes source context "along the South Bank" to lower uncertainty in its prediction.

These hybrid MT systems with NLM components, while having addressed shortcomings of traditional phrase-based MT, still translate locally and fail to capture long-range

[5] In [6], the authors have constructed a model that conditions on 3 target words and 11 source words, effectively building a 15-gram LM.

dependencies. For example, in Figure 1.3, the source-conditioned NLM 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!

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. Despite being relatively new [5, 12, 28], NMT has already shown promising results, achieving state-of-the-art performances for several language pairs such as English-French [21], English-German [10, 20, 27], and English-Czech [11, 18]. NMT is appealing since it is conceptually simple and can be trained end-to-end. NMT translates as follows: it reads through the given source words one by one until the end, and then, 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.

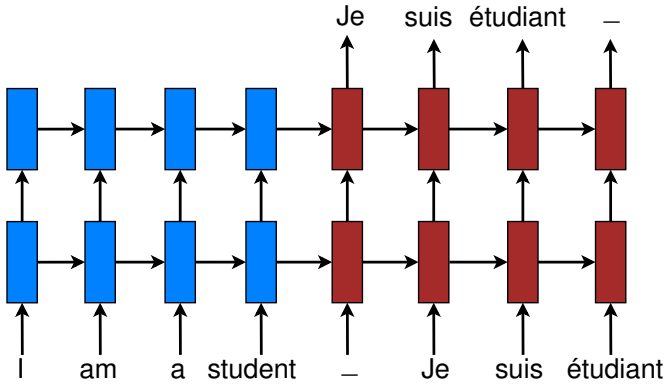

Figure 1.4: **Neural machine translation** – example of a deep recurrent architecture proposed by Sutskever et al. [28] for translating a source sentence "I am a student" into a target sentence "Je suis étudiant". Here, "_" marks the end of a sentence.

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 [15]. 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.

In this thesis, 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. We start off by providing background knowledge on RNN and NMT in Chapter **??**. The following chapters detail my contributions. Chapter **??** discusses how the rare word problem in NMT is addressed with a mechanism to "copy" words from source to target; hence, extending the vocabulary coverage. Chapter **??** describes how the attention mechanism, a way to select local contexts in the source sentence as we transate, can be effectively used in NMT to better handle long sentences. Chapter **??** proposes a novel way of dealing with language complexity (rich morphology, neologisms, and informal spellings) by building a hybrid word and character level model which can gain from the flexibility of a character-level model while maintaining the speed and quality of the word-level model. Towards the future of NMT, I answer two questions in Chapter **??**: (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; and (2) whether we can compress NMT for mobile devices. Chapter **??** wraps up and discusses remaining challenges in NMT research.

Chapter 2

Literature Review

This chapter describes the basic knowledge of Statistical Language Modeling together with the prominent approaches. The research context is drawn in order to show the necessity of the neural network-based approaches.

2.1 Overview

In general, language models aim at measuring the fluency of any text, showing how much it makes sense. Artificial systems that generate text, therefore, requires the aid of language models in order to produce textual outputs that are comprehensible.

From the statistical point of view, a word string is considered as a stochastic process, thus the language model is formulated as the probability estimation over all possible sequences of words. The sequence length can be arbitrary, while the words are taken from a limited vocabulary. For example, the likeliness of "the end of our world" is much higher than "tea end of our word", because the latter string is much less likely to be found in available English text.

Since direction estimation for that probability distribution is intractable, the probability of a sentence $P(w_1^L)$ is factorised using the chain rule:

$$P(w_1^L) = P(w_1|<\text{s}>) \prod_{i=2}^{L} P(w_i|<\text{s}>w_1^{i-1}) = \prod_{i=l}^{L} P(w_l|h_l) \tag{2.1}$$

in which, <s> is used to denote beginning token of the sentence/string. The history h_i represents the string before the current word w_i. Instead of directly modeling the original distribution, the statistical models estimate the constituent probabilities, which usually results in browsing or storing the statistics of millions of possible word strings since each language may consist of several thousands to millions of words.

In terms of application, language models are often placed

2.2 Evaluation Metrics

Quality measurement of language models is often carried out using two different ways. First, statistical language models can be evaluated by the capability to predict a new corpus. The perplexity (PPL) of a word sequence w_1^L is computed as follows.

$$PPL = \exp(\frac{\sum_{l=1}^{L} - \ln P(w_i|h_i)}{L}) \tag{2.2}$$

It is notable that the exponential term is the average of the negative log-likelihood of every word in the data, while $\log_2$ PPL is the **Entropy** of the model. A low perplexity value corresponds to the fact that the language model is able to fits better the data, since the distribution of the model is closer to the unknown distribution of the test data.

Second, the quality of language models can also be evaluated through their impacts on other applications such as Automatic Speech Recognition (ASR) or Statistical Machine Translation (SMT), by reducing the errors in the output of such systems. Specifically, in ASR, the metric used to evaluate the contribution of language models is Word Error Rate (WER) which is the distance between the decoder hypothesis and the reference. Similarly in SMT, the effect of language models can be reflected by the BLEU score [23] or even human judgement.

The main advantage of perplexity is that it is fast to perform and independent to other complex systems. The crediablilty of perplexity, however, depends on the validation/test data as well as the underlying vocabulary. It is also unusable for models that provide unnormalised distributions (sum of the distributions does not equal to 1). More importantly, an improvement in terms of perplexity does not always result in the application improvement.

For example, the improvement is required to be at least 10% to be noteworthy for an ASR system [24]

2.3 Prominent approaches in Language Models

2.3.1 N-gram statistical language models

The statistical methods are revised from equation 2.1, in which the probability of a sentence is factored into constituent conditioned probabilities. There are many approaches proposed to estimate those probabilities, however most of them revolve around **maximum likelihood** (MLE) of the training data together with **smoothing techniques** that help the models generalise better on unseen data. Such estimation is often done by collecting the word co-occurence frequencies, with the important Markovian assumption that the history h_i is limited to only $n-1$ words (thus called n-gram models):

$$P(w_i|h_i) \approx P(w_i|w_{i-n+1}^{i-1}) \tag{2.3}$$

In order to estimate each conditional probability by MLE, we simply count the number of occurence of the n-grams and the history:

$$P_{MLE}(w_i|w_{i-n+1}^{i-1}) = \frac{C(w_1^n)}{C(w_1^{n-1})} \tag{2.4}$$

where $C(X)$ is the number of times that the string X appears in the training data.

Problem arises when we need to estimate the probability distribution of rare word strings or even n-grams that do not occur in the corpus. According to Zipf law [13], the frequency distribution of a word is reversely proportional to its rank in the frequency table. The n-gram models, consequently, underestimate or even give null probabilities for unseen n-grams which actually make sense in natural language. As a result, smoothing techniques are employed in order to alleviate estimation of rare and unseen n-grams. In this section, we describe the most successful method which has been considered the standard in n-gram language models: the interpolated Knesey-Ney smoothing technique [14].

2.3.2 Interpolated Knesey-Ney smoothing

2.3.3 Structured language models

Originally, statistical language models were not warmly welcomed by the linguistic community, as can be seen from the statement of Chomsky: the notion of probability of a sentence is completely useless one. Essentially, n-gram models, which will be described in subsequent sections, and their finite state machine variations are not able to represent linguistic patterns and long term dependency between words in a sentence, or in a broad context.

Structured language models involve using **Context free grammars** to generate a syntax tree for the word string, in which the leaves represent the words and the other nodes are non-terminal symbols. The generation process is statistically learned from a training corpus, so that the final score of the sentence is decided from the probabilistic derivations of the grammar.

Despite the fact that structured language models are much more linguistically related than the statistical counterpart, they were not able to prove their practicality. The approach itself seems to be questionable when applied to speech content where the speakers do not strictly follow grammatical rules. Moreover, grammar construction often requires the participation of expert linguists and native speakers of the language, thus the method is expensive when applied to other languages.

2.3.4 Class based language models

2.3.5 Maximum Entropy Language models

2.4 Neural Network Language Models

2.4.1 Feed-forward variations

2.4.2 Recurrent Neural Network variations

Chapter 3

Neural Network Language Models

Chapter 4

Convolutional Neural Network

Chapter 5

Experiments

Chapter 6

Inside the Convolutional layer

Chapter 7

Conclusion

Bibliography

[1] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. 3:1137–1155, 2003.

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

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

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

[5] 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.

[6] J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, and J. Makhoul. Fast and robust neural network joint models for statistical machine translation. In *ACL*, 2014.

[7] 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.

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

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

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

[11] Sébastien Jean, Orhan Firat, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. Montreal neural machine translation systems for WMT'15. In *WMT*, 2015.

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

[13] Zipf George Kingsley. Selective studies and the principle of relative frequency in language, 1932.

[14] Reinhard Kneser and Hermann Ney. Improved backing-off for m-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 181–184. IEEE, 1995.

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

[16] 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 *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics, 2007.

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

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

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

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

[21] 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.

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

[23] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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, 2002.

[24] Ronald Rosenfeld. Two decades of statistical language modeling: Where do we go from here. In *Proceedings of the IEEE*, page 2000, 2000.

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

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

[27] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *ACL*, 2016.

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

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

[30] 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.