Notes on Neural Machine Translation

compiled by D.Gueorguiev 7/19/2025

# Introduction to Statistical Machine Translation (SMT)

## A Little Bit of History with IBM Models for SMT

The IBM translation models go back to late 1980s and early 1990s. These models still form the basis for many SMT now in use today.

Let us assume that the task is to translate from *French* (the *source* language) into *English* (the *target* language).

We will use the symbol to refer to a sentence in *French*. is a sequence of words where is the length of the sentence, and for is the -th word in the sentence. We will use to refer to an *English* sentence: is equal to where is the length of the English sentence.

We have a set of example translations (our training data) - for , where is the -th *French* sentence , is the k-th *English* sentence, and is the translation of . Each is equal to where is the length of the -th *French* sentence. Each is equal to where is the length of the -th *English* sentence. We will estimate the parameters of our model from these training examples.

### The Noisy-Channel Approach

The IBM models are an instance of a noisy-channel model and as such they have two components :

1 A *language model* that assigns a probability for any sentence in English.

//TODO: mainly [16] and [17]

## Alignment in Neural Machine Translation (NMT)

In NMT, *alignment* is the process of identifying word or phrase correspondence between source and target sentences. While early NMT models did not explicitly model alignment, modern approaches, particularly those using attention mechanisms, have incorporated ways to learn these relationships implicitly or explicitly.

Soft Implicit Alignment achieved with the Attention Mechanism

NMT especially those based on the Transformer architecture, utilize attention mechanisms to weight the importance of different parts of the source sentence when generating each word of the target sentence.

These attention weights effectively create a *soft alignment*, indicating the degree to which each source word influences the target word. Higher attention weights suggest stronger alignment. The soft alignment through Attention represents a model of *implicit alignment*.

For example in translating *“The cat sat on the mat”* to French, the attention mechanism may assign higher weights to *“cat”* when generating the French word *“chat”*, and to “mat” when generating *“tapis”*.

Explicit Alignment Models

While Attention provides implicit alignment, some research explores explicit alignment for better interpretability, constant imposition, and handling of specific translation phenomena. Explicit Alignment Models coexist with the main NMT model.

# Sequence Transduction (ST)

Sequence Transduction is the transformation of input sequences into output sequences.

Examples of sequence transduction:

Speech recognition

Machine translation

Protein secondary structure prediction

Challenge in sequence transduction:

Learning to represent both the input and output sequences in a way that is invariant to sequential distortions such as *shrinking*, *stretching*, and *translating*.

RNNs are powerful sequence learning architecture that has proven

//TODO: Finish this section (mainly [1])

## Continuous Translation Models (CTM)

//TODO: Finish this section (mainly using [2], [3], [4])

## Recurrent Continuous Translation Models (RCTM)

RCTM are a class of probabilistic

//TODO: Finish this section (mainly using [5])

# References

[1] [Sequence Transduction with Recurrent Neural Networks, Alex Graves, 2012](https://github.com/dimitarpg13/nlp_concepts/blob/main/literature/articles/neural_machine_translation/Sequence_Transduction_with_Recurrent_Neural_Networks_Graves_2012.pdf)

[2] [Continuous Space Language Models for Statistical Machine Translation, Holger Schwenk et al, 2006](https://github.com/dimitarpg13/nlp_concepts/blob/main/literature/articles/neural_machine_translation/Continuous_Space_Language_Models_for_Statistical_Machine_Translation_Schwenk_2006.pdf)

[3] [Continuous Space Translation Models for Phrase-Based Statistical Machine Translation, Holger Schwenk, 2012](https://github.com/dimitarpg13/nlp_concepts/blob/main/literature/articles/neural_machine_translation/Continuous_Space_Translation_Models_for_Phrase-Based_Statistical_Machine_Translation_Schwenk_2012.pdf)

[4] [Continuous Space Translation Models with Neural Networks, Le Hai Son et al, 2012](https://github.com/dimitarpg13/nlp_concepts/blob/main/literature/articles/neural_machine_translation/Continuous_Space_Translation_Models_with_Neural_Networks_Le-Hai-Son_2012.pdf)

[5] [Recurrent Continuous Translation Models, Nal Kalchbrenner, Phil Blunsom, Oxford U., 2013](https://github.com/dimitarpg13/transformers_intro/blob/main/articles_and_books/Recurrent_continuous_translation_models_Kalchbrenner_Blunsson_OxfordU_2013.pdf)

[6] [Neural Machine Translation by Jointly Learning To Align and Translate, Dzmitry Bahdanau, K. Cho, Yoshua Bengio, 2016](https://github.com/dimitarpg13/transformers_intro/blob/main/articles_and_books/NeuralMachineTranslationByJointlyLearningToAlignAndTranslateBahdanau2015.pdf)

[7] [On The Properties of Neural Machine Translation: Encoder-Decoder Approaches, K. Cho, B. van Merrienboer, Dzmitry Bahdanau, 2014](https://github.com/dimitarpg13/transformers_intro/blob/main/articles_and_books/OnthePropertiesOfNeuralMachineTranslationEncoderDecoderApproaches.pdf)

[8] [Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, K. Cho, B. van Merrienboer, C. Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio, 2014](https://github.com/dimitarpg13/transformers_intro/blob/main/articles_and_books/LearningPhraseRepresentationsUsingRNNEncoderDecoderForStatisticalMachineTranslation.pdf)

[9] [Generating Sequences With Recurrent Neural Networks, Alex Graves, U of Toronto, 2014](https://github.com/dimitarpg13/deep_learning_and_neural_networks/blob/main/literature/articles/Generating_Sequences_With_Recurrent_Neural_Networks_Graves_2014.pdf)

[10] [Sequence to Sequence Learning with Neural Networks, Ilya Sutskever et al, 2014](https://github.com/dimitarpg13/nlp_concepts/blob/main/literature/articles/neural_machine_translation/Sequence_to_Sequence_Learning_with_Neural_Networks_Ilya_Sutskever_Google_2014.pdf)

[11] [Fast and Robust Neural Network Joint Models for Statistical Machine Translation, J. Devlin et al, Raytheon, 2014](https://github.com/dimitarpg13/nlp_concepts/blob/main/literature/articles/neural_machine_translation/Fast_and_Robust_Neural_Network_Joint_Models_for_Statistical_Machine_Translation_Devlin_2014.pdf)

[12] [Neural Machine Translation of Rare Words with Subword Units, Rico Sennrich, Barry Haddow, Alexandra Birch, 2015](https://github.com/dimitarpg13/nlp_concepts/blob/main/literature/articles/neural_machine_translation/Neural_machine_translation_of_rare_words_with_subword_units_UEdinsburgh_2016.pdf)

[13] [Bidirectional Recurrent Neural Networks, Mike Schuster, Kuldip Paliwal, IEEE Transactions on Signal Processing, 1997](https://github.com/dimitarpg13/transformers_intro/blob/main/articles_and_books/Bidirectional_Recurrent_Neural_Networks_Schuster_Paliwal_1997.pdf)

[14] [Neural Networks for Pattern Recognition, Christopher M. Bishop, 1995](https://github.com/dimitarpg13/transformers_intro/blob/main/articles_and_books/Neural_Networks_for_Pattern_Recognition-Bishop_1995.pdf)

[15] [Introduction to Finite-State Devices in Natural Language Processing, TR-96-13, Emmanuel Roche and Yves Schabes. 1996](https://github.com/dimitarpg13/nlp_concepts/blob/main/literature/articles/neural_machine_translation/Introduction_to_Finite-State_Devices_in_Natural_Language_Processing_roche_schabes_1996.pdf)

[16] [Statistical Machine Translation: IBM Models and word alignment, Patrik Lambert, 2010](https://github.com/dimitarpg13/nlp_concepts/blob/main/literature/articles/neural_machine_translation/MTMarathon-2010-Lambert-ppt.pdf)

[17] [Statistical Machine Translation: IBM Models 1 and 2 with annotations, Michael Collins, 2013](https://github.com/dimitarpg13/nlp_concepts/blob/main/literature/articles/neural_machine_translation/Collins_annotated_Statistical_Machine_Translation-IBM_Models_1_and_2.pdf)

[18] [Revisiting Optimal Decoding for Machine Translation IBM Model 4, Sebastian Riedel, James Clarke, 2009](https://github.com/dimitarpg13/nlp_concepts/blob/main/literature/articles/neural_machine_translation/Revisiting_Optimal_Decoding_for_Machine_Translation_IBM_Model_4_Riedel_2009.pdf)

[19] [N-Gram Language Models](https://github.com/dimitarpg13/nlp_concepts/blob/main/literature/articles/neural_machine_translation/N_gram_language_models_chapter_from_SaLP_Jurafsky_2025.pdf), a chapter of [Speech and Language Processing, Daniel Jurafsky and James Martin, Draft, 2025](https://github.com/dimitarpg13/nlp_concepts/blob/main/literature/books/Speech_and_Language_Processing_Jurafsky_3ed_2025.pdf)

# Appendix

## N-gram Language Models

We are given the task of computing , the probability of a word given some history . Suppose the history is *“The water of Walden Pond is so beautifully “* and we want to know the probability that the next word is *blue*:

(ngr.1)

One way to estimate this probability is directly from relative frequency counts: take a very large corpus, count the number of times we see *The water of Walden Pond is so beautifully*, and count the number of times each of those occurrences is followed by *blue*. This would be answering the question “Out of the times we saw the history , how many times was it followed by the word ”. This is expressed as:

(ngr.2)

The problem is we rarely have large enough corpus to compute all the probabilities of relevant parts of the sentences using (ngr.2).

Notation:

We will represent sequence of words either as or . Thus, the expression denotes the sequence but we will also use the equivalent notation .

For the joint probability of each word in sequence having a particular value we will use .

Question:

How do we compute ?

We can apply the chain rule to words as:

(ngr.3)

The problem with (ngr.3) is that it is not especially helpful. We do not know how to compute .

We need to make some assumption if we are to compute by using (ngr.3).

### The Markov assumption