Karlsruhe Institute of Technology

AI4LT

Unsupervised Quality Estimation in Spoken Language Translation with Glass-Box Features

Bachelor's Thesis of

Isabel Kraft

Artificial Intelligence for Language Technologies (AI4LT) Lab
Institut for Anthropomatics and Robotics (IAR)
KIT Department of Informatics

Reviewer:	Prof. Dr. Jan Niehues
Second reviewer:	Prof. Dr.-Ing. Rainer Stiefelhagen
Advisor:	M.Sc. Tu Anh Dinh

10. September 2024 – 10. January 2025

Karlsruher Institut für Technologie
Fakultät für Informatik
Postfach 6980
76128 Karlsruhe

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

PLACE, DATE Karlsruhe, der 9. Januar 2025

.....

(Isabel Kraft)

Abstract

This thesis explores unsupervised quality estimation for spoken language translation (SLT), leveraging glass-box features from neural models. Estimation of the quality is crucial for knowing when a translation or transcription is of a bad quality and has to be human edited. This is especially important on machine translation and automatic speech recognition (ASR) technologies used in a cascaded SLT model.

The thesis adapts established unsupervised metrics from text-based machine translation to evaluate SLT systems, focusing on cascaded models but also having a look at an end-to-end model. Methods such as transcription probability, softmax entropy, standard deviation of token probabilities, and uncertainty quantification with dropout are employed to estimate the quality of transcription and translation. Additionally, different unified scoring approaches are proposed to evaluate the overall performance of cascaded SLT systems. Experiments are conducted using state-of-the-art models like Whisper, SeamlessM4T, and DeltaLM, demonstrating the efficacy of the proposed metrics in correlating with evaluation measures.

We conclude that the transcription mean, the translation probability and standard deviation of token probabilities are good metrics to estimate the quality of Spoken Language translation ins cascaded model and translation probability and standard deviation are good metrics for end-to-end models.

Zusammenfassung

Diese Bachelorarbeit untersucht die unüberwachte Qualitätsschätzung für gesprochene Sprachübersetzung (Spoken Language Translation, SLT) unter Verwendung von Glasbox-Merkmalen aus neuronalen Modellen. Angesichts der zunehmenden Abhängigkeit von maschineller Übersetzung und Technologien zur automatischen Spracherkennung (Automatic Speech Recognition, ASR) wird eine präzise Qualitätseinschätzung entscheidend, um den menschlichen Nachbearbeitungsaufwand zu minimieren.

Die Studie passt etablierte, unüberwachte Metriken aus textbasierter maschineller Übersetzung an, um SLT-Systeme zu bewerten, und konzentriert sich dabei auf kaskadierte Modelle wobei ein End-to-End-Modell betrachtet wird. Methoden wie Transkriptionswahrscheinlichkeit, Softmax-Entropie und Unsicherheit Quantifikation durch Dropout werden verwendet, um die Qualität von Transkriptionen und Übersetzungen zu schätzen. Darüber hinaus wird ein einheitlicher Bewertungsansatz vorgeschlagen, um die Gesamtleistung kaskadierter SLT-Systeme zu bewerten. Experimente mit modernen Modellen wie Whisper, SeamlessM4T und DeltaLM zeigen die Wirksamkeit der vorgeschlagenen Metriken in der Korrelation mit Evaluationsmethoden.

Wir kommen zu dem Schluss, dass der Transkriptionsmittelwert, die Übersetzungswahrscheinlichkeit und die Standardabweichung der Token-Wahrscheinlichkeiten gute Metriken zur Schätzung der Qualität von gesprochenen Sprachübersetzungen in kaskadierten Modellen sind, während Übersetzungswahrscheinlichkeit und Standardabweichung gute Metriken für End-to-End-Modelle sind.

Contents

Abstract i

Zusammenfassung iii

1. Introduction 1

2. Background 3
2.1. Basic Knowledge . . . 3
2.1.1. Sequence Generation . . . 3
2.1.2. Automatic Speech Recognition . . . 4
2.1.3. Translation . . . 4
2.1.4. Speech translation . . . 5
2.1.5. Softmax . . . 7
2.1.6. Entropy . . . 7
2.1.7. Dropout . . . 7
2.2. Models . . . 8
2.2.1. Decoder-Encoder . . . 8
2.2.2. Transformer . . . 9
2.2.3. Whisper . . . 10
2.2.4. Seamless . . . 11
2.2.5. SentencePiece . . . 12
2.2.6. DeltaLM . . . 13
2.3. Evaluation metrics . . . 13
2.3.1. WER . . . 13
2.3.2. Comet . . . 14
2.3.3. Pearson-correlation . . . 14

3. Related works 17

4. Methodology 19
4.1. Previously proposed . . . 19
4.1.1. Translation probability . . . 19
4.1.2. Softmax Entropy . . . 20
4.2. Dropout . . . 20
4.2.1. General Probability . . . 20
4.2.2. Variance . . . 21
4.2.3. Combo . . . 21

4.3. Proposed Methods . . . 21
4.3.1. Transcription probability . . . 21
4.3.2. Standard Deviation . . . 22
4.3.3. Dropout on ASR systems . . . 22
4.3.4. Unified Score . . . 23
4.3.5. Spoken language methods . . . 23

5. Experiments **25**
5.1. Dataset . . . 27
5.1.1. Segmentation . . . 27
5.1.2. Training . . . 27
5.2. Models . . . 28
5.2.1. Whisper . . . 28
5.2.2. DeltaLM . . . 28
5.2.3. Seamless . . . 28
5.3. Dropoutless Experiments . . . 29
5.4. Dropout . . . 29
5.5. Unified Scores . . . 29

6. Evaluation **31**
6.1. Transcription evaluation . . . 31
6.2. Translation evaluation . . . 32
6.3. Dropout evaluation . . . 35
6.4. One unified score . . . 40

7. Conclusion **45**

Bibliography **47**

A. Appendix **53**
A.1. Implementation information . . . 53
A.1.1. Transcription probability . . . 53
A.1.2. Translation probability . . . 53
A.1.3. Softmax entropy . . . 53
A.1.4. Standard deviation . . . 54
A.2. Other interesting pearson correlation scores . . . 54
A.3. References graphs . . . 54

List of Figures

2.1. Schematic architecture for an encoder-decoder speech recognizer from [21] 5
2.2. Basic overview of a cascaded speech translation model 6
2.3. basic overview of a end-to-end model architecture 6
2.4. dropout in a basic Neural Network model 8
2.5. basic overview of an encoder-decoder model architecture 9
2.6. Example of a single sentence translation in a basic RNN encoder-decoder approach. The source and target sentence are concatenated with a separator token. The decoder then uses the context information from the last hidden state of the encoder. Taken from [21] 9
2.7. Transformer model architecture Vaswani et. al 2017 10
2.8. Overview of the Whisper architecture Radford et. al 2022 11
2.9. Overview of SeamlessM4T. (1) shows the pre-trained models used when finetuning multitasking UnitY. (2) outlines multitasking UnitY with its two encoders, text decoder, T2U encoder-decoder, and the supporting vocoders for synthesizing output speech in S2ST. [8](figure 4) 12
2.10. Vanilla Transformer decoder (left) compared to the interleaved Transformer decoder(right) from Ma et. al 2021 13
2.11. overview of the comet estimator model architecture from [39] 14
2.12. graphical representation of different Pearson correlation scores [35] . . . 15

6.1. plot over the transcription probabilities, the transcription means, and the WER scores 32
6.2. Model translation scores over the corresponding comet scores. The left side shows the seamless scores, the right side shows the DeltaLM scores. 33
6.3. Model softmax entropy scores over the corresponding comet scores. The left side shows the seamless scores, the right side shows the DeltaLM scores 34
6.4. Model standard deviation scores over the corresponding comet scores. The left side shows the seamless scores, the right side shows the DeltaLM scores 35
6.5. transcription dropout scores plotted over the WER; left is the base scores, right is the mean scores 36
6.6. transcription dropout variance scores plotted over the wer, left is the base scores, right half is the mean scores 37
6.7. transcription dropout scores plotted over the wer, left is the base scores, right half is the mean scores 37
6.8. dropout probability scores plotted over the comet scores, left is the seamless scores, right half is the DeltaLM scores 38
6.9. dropout variance scores plotted over the comet scores, left is the seamless scores, right half is the DeltaLM scores 39

6.10. Dropout combo scores plotted over the comet scores; left is the seamless scores, right is the DeltaLM scores 39
6.11. Plot of the unified scored calculated by adding the translation and transcription scores together, a) and b) are calculated with the non normalized transcription scores, c) and d) with the normalized transcription scores . 43
6.12. Changes in correlation with different alpha values: the red line is for the unified score taken with the transcription mean, the blue line is the unified score gathered with the the non-normalised transcription score. 44

A.1. Reference scores over dataset lines 55

List of Tables

5.1. Example of differing transcript results, the top is an example where the transcript with dropout (first line) that has the highest quality estimation score is the same as the transcript without dropout. Compared to an example where the transcription withe highest score is vastly different from the non dropout transcription, 2. to last line. The last line is a different result from the dropout runs that is essentially the non dropout result. The left column is the corresponding quality score; the quality score shown is the transcription mean. 26

6.1. result from the transcription part of the cascaded model, correlated with WER scores, with the reference and model transcript normalized, the sign on the left denotes what sign a row should have 32
6.2. Correlation scores for the separate models and calculated quality scores. The sign on the left denotes whether the expected correlation value is supposed to positive or negative. If no signs are added to the values in a row, then all of the values have the expected sign. 33
6.3. pearson correlations of multiplicative scores for various other metrics. The columns are separated by whether the transcript probability, denoted with base, or the transcription mean was used to calculate the score, as well as the translation model. 40
6.4. pearson correlation scores that are added together from the different scores in the translation transcription categories. The columns are separated by whether the transcript probability, denoted with base, or the transcription mean was used to calculate the score, as well as the translation model. . 41
6.5. correlation scores of addition unified scores where the absolute of input scores has been taken 42

1. Introduction

Spoken language is everywhere, from talking to people in person, radio transmissions or videos a big part of communication is spoken language. But not every person speaks the same language. Take for example a look at the internet, a very large section of the internet is in English. This includes many websites, hours upon hours of audio in for example the form of podcasts or radio and videos like the many hours of video material uploaded to sites like YouTube. All of this is only accessible if a person speaks English or someone translates it into a different language.

But translation is a time intense problem, that with the help of machine translation has gotten a lot easier. Since the machine translation is not perfect, humans still need to double check the translations. This process of double checking the translations still takes a lot of time as the human that does the quality check has to read over the entire translation and the source material to be able to gauge if the translation is good.

Spoken language translation only adds more potential problems to this. If a human would have to translate spoken language, like a video or a speech, they would have two options, firstly transcribing the spoken language into text in the source language and then translating it, or secondly translating it on the go. Both of these take a lot of time as depending on the language used finding a good translation can be hard. Fortunately Natural Language Processing, NLP for short, has made a lot of advancements in the past years and the technology to make such tasks easier exist already. Automatic Speech Recognition, or in short ASR, has gotten quite good and quite fast at transcribing audio in many different languages and machine translation can translate almost anything.

But this alone does not help speeding up the process of spoken language translation, as a human would still have to listen to the entire audio to make sure the transcript is right and the translation based off of this transcript also doesn't make mistakes. To remedy this Quality Estimation can be employed to estimate the quality of the transcript and the translation.

Quality Estimation, like the name implies, has the goal of to estimate the quality of a system, a product or a result. This thesis takes a closer look at quality estimation of such spoken language translation, more specifically it takes a look at cascaded model spoken language translation and the quality estimation of this. To estimate the quality of the spoken language translation, different features from both the ASR and machine translation (MT) systems are used to create a so called white or glass box approach. In addition to that the aimed for quality estimation techniques should be usable in an unsupervised manner, this means that no references are needed.

The basis for the approach used in this thesis was proposed on text translation by Fomicheva et al. [15] and has been adapted to Spoken language translation.

2. Background

This chapter explains the methods and concepts. These are separated in 3 parts: one for basic knowledge, one for the evaluation methods used, and one for the used models.

2.1. Basic Knowledge

Important knowledge for understanding the contents of the thesis: this includes how an encoder-decoder model can generate an output sequence; what machine translation, automatic Speech recognition, and dropout are; and a section about the Models used, the different basic model architectures, and an explanation of the model architectures of the specific models used in the experiments.

2.1.1. Sequence Generation

An encoder-decoder model generates a sequence $y = (y_1 \dots y_m)$ from an input sequence $x = (x_1 \dots x_n)$, by maximizing the Probability

$$P(y|x)$$

This probability is a conditional probability, which is used in Bayesian inference and uses Bayes rule to break such a conditional probability down into three different probabilities. Bayes rule is described as $P(a|b) = \frac{P(b|a)P(a)}{P(b)}$. Because of Bayes rule the Sequence Probability can be written as

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

Since we know that the input sequence is constant the Probability can be simplified to $P(x|y)P(y)$. So if only the first token y_1 of the sequence y is supposed to be generated, this would be done by

$$P(y_1) = argmax_{y_1 \in V} P(y_1|x)$$

as the first token of the sequence only depends on the input sequence x and is the determined by the highest probability, which is what the argmax stands for, in the vocabulary. The second token in the sequence in turn depends on the input sequence and the first token, so the probability is

$$argmax_{y_2 \in V} P(y_2|y_1, x)$$

This pattern continues, and because of the chain rule of probability we can write the probability for generating the whole sequence as

$$P(y|x) = \prod_{i=1}^{m} P(y_i|x, y_{<i})$$

[21, chapters 3, 9.5, 10, 13] This behaviour of being able to generate the next token in the sequence just from the previous elements is also called autoregressive generation.

For the actual models, this probability of generating the i-th token is the probability after applying the softmax to the probability distribution over the vocabulary at the last layer of the model, and the model parameters Θ are also part of the conditional part of the probability $P(y_i|y_{<i}, x, \Theta)$. This means the mathematical description of the sequence probability is

$$p(y|x, \Theta) = \prod_{i=1}^{T} p(y_i|y_{<i}, x, \Theta)$$

2.1.2. Automatic Speech Recognition

Automatic speech recognition systems, or in short ASR systems, are systems that recognise and transcribe spoken language into written text. Historically this was done with 2 different models. One is called the acoustic model, which models the relationship between the audio signal and phonetic units, usually by means of classification. The second one is called the language model, which assigns probability estimates to word sequences and thus defines what might be said in the audio and the vocabulary that is used. To do this it tries to differentiate between different sequences. [14]

In more recent approaches the same has been done with a single end-to-end encoder-decoder model with great success. To do this an ASR system takes the audio data, which is in waveform, and transforms it to extract a sequence of acoustic feature vectors. Each of those acoustic feature vectors in the sequence contains a small time window of the signal. This transformation is usually done with the log mel spectrum to get a mel-log spectrogram. [21, chapter 16] This mel-log spectrogram encodes the audio frequencies onto the mel-scale which is based on human perception, as humans do not perceive frequencies on a linear scale [49] and then represents these transformed frequencies on a logarithmic scale. The input features are then put into the encoder after being subsampled down to a shorter sequence, as the input feature sequence is quite long. As per how an encoder-decoder architecture works the decoder then takes the encoder representation and decodes it into a text sequence. This architecture is shown in Figure 2.1.

2.1.3. Translation

Translation is the practice of translating text or language from one language into another language. This can be done by hand by a human or in a very statistical approach where a dictionary is used to directly translate the text. Another way is with a machine learning approach where a Neural network learns the representations of one language and how this translates in a different language. The current standard Machine translation (MT) architecture is an encoder-decoder transformer architecture, which is explained more in depth later. As most sentences can be translated independently from each other, this is what is usually done in machine translation. To translate a sentence from the source language into the target language the MT system has to generate the corresponding target sentence. [21, chapter 13.2]

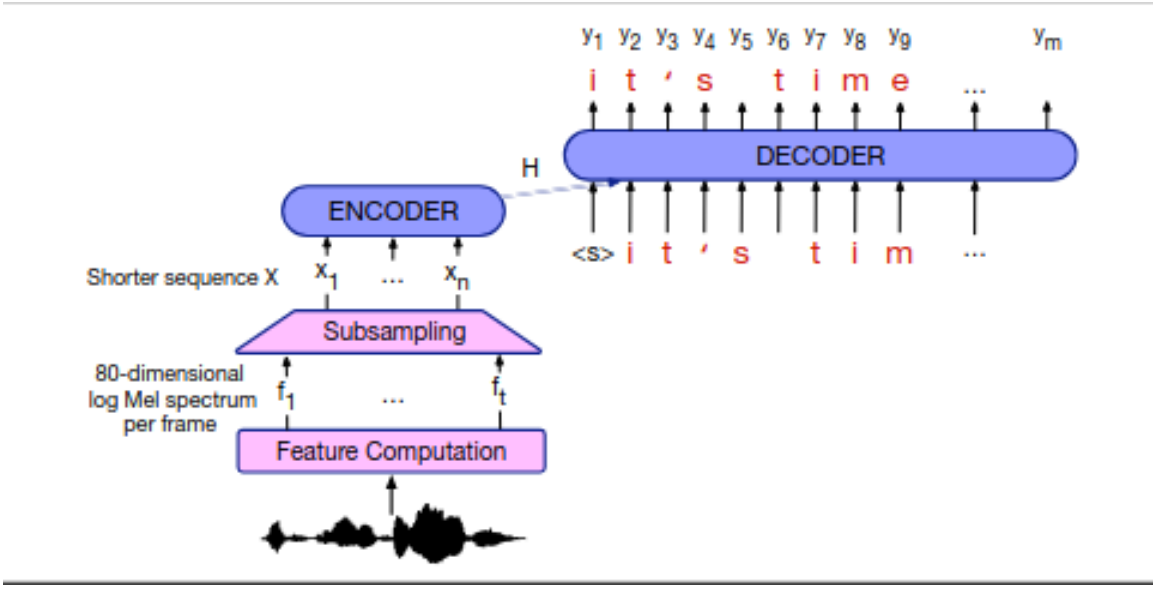


Figure 2.1.: Schematic architecture for an encoder-decoder speech recognizer from [21]

For example the German source sentence: "Der Apfel ist grün." would have to be translated into the English sentence "The apple is green".

For a MT model to be able to do this translation it has to be trained on on data. This data is a large amount of matched sentences in both the source and target languages. The training itself is supervised machine learning; this means the system has to learn how to map the source sentences to the target sentences. To do this the input is encoded into tokens. These can be words, subwords, so parts of words, or characters. It is more useful to do this with units that are smaller than words. These are called subword tokens and they have the advantage that they make it easier to generate the vocabulary of the MT system, which is fixed in advance, usually by the training. In an encoder-decoder architecture the encoder will take the input sentence in the source language and produce the context, which is passed to the decoder which produces the decoded target language sentence. [21, chapter 13.2]

2.1.4. Speech translation

Speech translation or spoken language translation is similar to regular translation but it has, like the name says, spoken language as the basis for the translation instead of text. There are 2 main approaches to Speech Translation (ST): cascaded models, which consist of a ASR model and a MT model, and end-to-end models, which have are only one model, most of the time a encoder-decoder architecture. Cascaded models have been used for a long time as they are easy to implement. But end-to-end models have gained quite a lot more popularity in the last years as they promise better results and ideas that have proven to work well in text-to-text MT have been tested for ST, with good results.

2.1.4.1. Cascaded Models

Cascaded Speech translation Models consist of 2 parts: a part that is responsible for transcribing the audio, which is usually done with an ASR model, and a part that is responsible for translating the resulting transcription, which is done with neural machine translation or statistical machine translation. A basic overview of this can be seen in Figure 2.2.

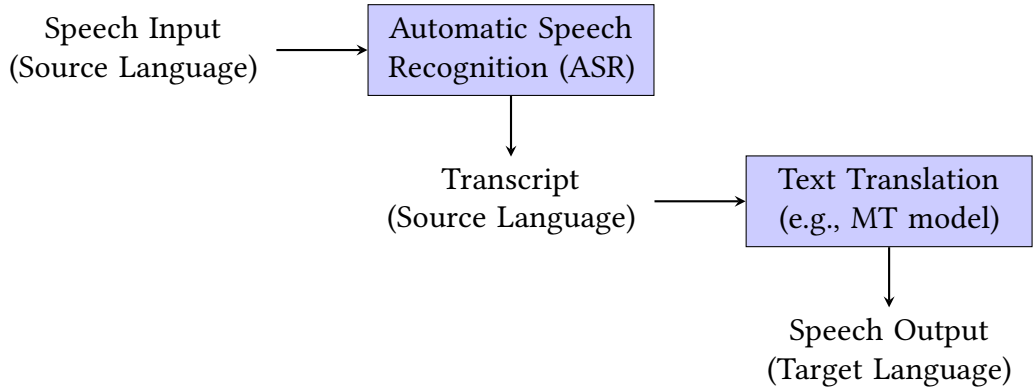


Figure 2.2.: Basic overview of a cascaded speech translation model

To train a cascaded ST model one would have to train the ASR system and the MT system. This can be done separately as long as the ASR system is trained for the source language and the MT system is trained to translate form the source to the target language. This makes cascaded systems quite convenient to train as different datasets can be used to train both parts of the model as long as they have vocabulary overlap.

2.1.4.2. End-to-End Models

End-to-End Speech translation models do not have the explicit split between the Automatic Speech Recognition model and the translation. This means that such a model gets audio as an input and outputs the text in the target language. End-to-End models are trained to perform a task from the raw input, which in the case of Speech translation case is audio, to the output, in this case the corresponding translation, without any intermediate processing from outside the model or feature-engineering in sub-models. This can be seen in Figure 2.3.

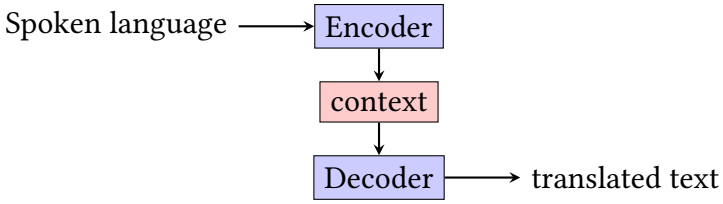


Figure 2.3.: basic overview of a end-to-end model architecture

To train an end to end ST system a dataset with audio and corresponding translations in the target language is needed, as the system has to learn the mapping between the

audio sequences and the translated sentences. This process works very similar to the MT training process. However it is significantly more time intensive to create such datasets, especially if a more conversational and casual setting of spoken language is looked for.

2.1.5. Softmax

Taking the softmax of a vector or probability distribution maps each entry in that probability distribution to a value between 0 and 1 by setting all values below 0 to 0, scaling all values that are bigger than 0 to fall between 0 and 1, as well as making sure that the sum of all the values is 1. Mathematically this can be described as

$$softmax(z)_i = \frac{exp(z_i)}{\sum_j exp(z_i)}$$

where $z = (z_1 \dots z_K) \in R^K$.

So an example of applying the softmax to a probability distribution follows. Let $x = (0, -1, 2, 1, -5, 5)$ be a vector or probability distribution that the softmax is to be applied on. The sum of all non 0 values is $2+1+5 = 8$. Applying the softmax to each element goes like this: the first element, the 0, stays a 0; the second element, the -1, is negative so the the value is set to 0. The third element, 2, is positive, so it gets scaled down by the sum of all positive values $2/8 = 0.25$; the fourth element 1 also gets scaled down to 0.125. The fifth element is -5 so it gets set to 0 as well, and the last element in the array is 5 which gets scaled down to be $5/8 = 0.625$. This results in the softmaxed vector $softmax(x) = (0, 0, 0.25, 0.125, 0, 0.625)$. and this fulfills the requirement that $\sum_i x_i = 0 + 0 + 0.25 + 0.125 + 0 + 0.625 = 1$. [21, chapter 5.3.1]

2.1.6. Entropy

In Information theory the entropy is the average amount of uncertainty or information that is found in a discrete probability distribution. More specifically the entropy of a random variable measures the average amount of uncertainty that is connected to that random variable. The concept of Entropy was proposed by Shannon for the context of measuring the information in transmitted data [41]. Entropy is also used in thermodynamics where is is used to denote the randomness, disorder, or uncertainty of systems and it is the central point of the 2. law of thermodynamic.

The entropy is mathematically defined as $-\sum_{x \in \chi} p(x) log p(x)$,where p is a probability, χ is the probability distribution, and x is an element from that probability distribution.

2.1.7. Dropout

Dropout in the context of machine learning usually refers to the process of dropping out, so omitting, units in the neural network. One variation of these dropout methods is Monte-Carlo dropout [17], in short MC dropout, which is the process of masking neurons in a Deep Neural Network randomly, based on a Bayesian probability to 0. A graphic example of this masking values to 0 can be seen in Figure 2.4. This is usually done during training

to reduce the chance of the model overfitting on the training data, and this application was first proposed by Srivastava et al. [44]. Nowadays it is very common to use dropout in general and in particular MC dropout during training for this exact reason.

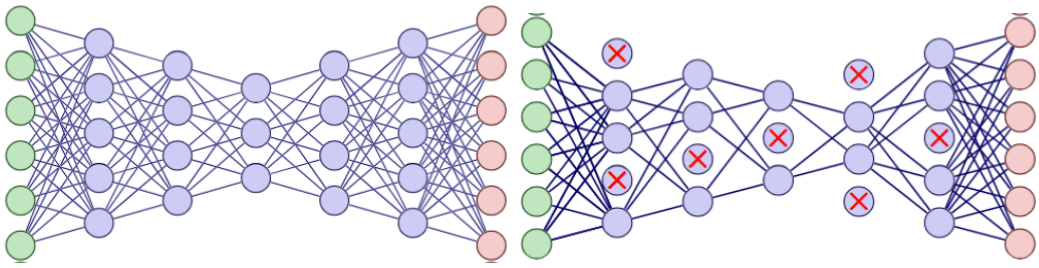


Figure 2.4.: dropout in a basic Neural Network model

Monte Carlo Dropout has also been utilized in Deep Neural Networks to measure the uncertainty of the Network. This application of it has been shown in [17]. Gal et. al also demonstrate in this that using dropout can be interpreted as a Bayesian approximation of Gaussian processes. To obtain the model uncertainty from this Bayesian approximation they only use the predictive mean and uncertainty of stochastic forward passes. Both the predictive mean and the uncertainty that this delivers are part of the methods used to retrieve quality estimation scores in chapter 4 and thus are explained more there.

This use of Monte Carlo dropout is mostly used in deep architectures but is also used to measure the uncertainty in Auto-encoders. [18] The use of dropout to measure the model's uncertainty in Automatic Speech recognition has been tried by Vyas et al. [52] before. They have found that most uncertainty in ASR stems from noisy input data, especially since non-noisy data produces good transcriptions a lot of the time if the model knows all used words and names.

2.2. Models

All the models used in this thesis use an encoder-decoder architecture, and more specifically transformer architecture models, which is a subsection of encoder-decoder models. Encoder-decoder models are also sometimes called sequence-to-sequence models as they are able to generate sequences of arbitrary length based on the input sequence. All of the models used are such encoder-decoder models.

2.2.1. Decoder-Encoder

Encoder-decoder Models are Models that contain an encoder and a decoder. The encoder creates an embedding of the input; this embedding is also called the context or context representation. While embedding the encoder can also add additional context to the representation. This context representation is then input into the decoder, which decodes it into a form that is for example a human readable text. The basic idea of this can be seen in Figure 2.5.

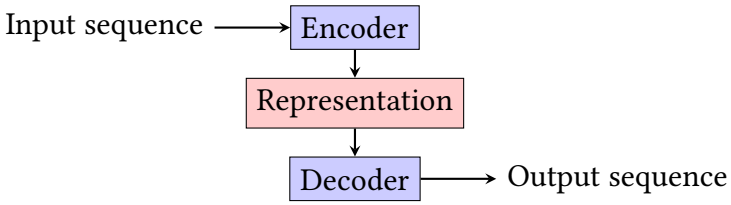


Figure 2.5.: basic overview of an encoder-decoder model architecture

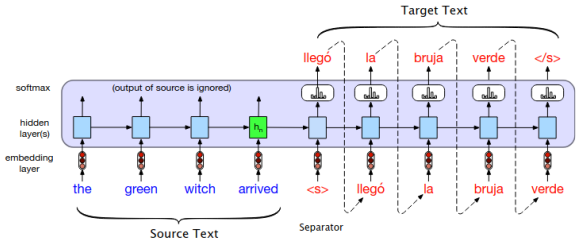


Figure 2.6.: Example of a single sentence translation in a basic RNN encoder-decoder approach. The source and target sentence are concatenated with a separator token. The decoder then uses the context information from the last hidden state of the encoder. Taken from [21]

As mentioned before the encoder-decoder Architecture is used for Sequence-to-Sequence models, since it allows inputting a sequence and outputting a sequence that can be a different length than the input sequence. [21, chapter 8.7]

Using an encoder-decoder architecture as a sequence-to-sequence translation model was first introduced by Sutskever et. al [46]. They proposed it using LSTMs, which stands for Long Short Term Memory networks and are a kind of Recurrent neural network (RNN) that can remember information longer than regular RNNs by having a more complex internal structure than RNNs, as the makeup of both the encoder and decoder. A simplified example of the translation process that was proposed is pictured in Figure 2.6.

2.2.2. Transformer

The Transformer architecture is a Neural Network architecture that was first introduced in the paper Attention is all you need [50] that makes use of self-attention mechanisms.

The attention mechanism is responsible for weighing and combining the representations from other relevant tokens in the context of the layer k-1 to create the representation for tokens in layer k. This creates contextual embeddings for words in a sequence by adding meaning from contextually relevant words to the embedding. Self-attention mechanisms make use of this attention and also use information from all previous representations in the context window. Depending on the model the transformer's self-attention mechanism can also make use of the representations of later tokens. This all describes a single attention head; the transformer has several of these which attend to different purposes in the context.

The Transformer has an encoder-decoder structure with N encoder blocks that are made up out of Multi-Head Attention blocks and feed forward networks. Both of these have an

add and normalization layer behind that, which each take the input token x_i, where i is the position of the token, put it through the Multi-Head Attention add it back to x_i. This is then put into the feed forward network and added back onto the previous result. This process can be seen in Figure 2.7 along with the general architecture of the transformer. The N blocks can map an entire context window of input vectors $(x_1 \dots x_n)$ to a of the same length window of output vectors $(h_1 \dots h_n)$. The decoder part consists of N blocks that contain a Masked multi-head attention block, a Multi-layer attention block, and a feed forward neural network. In the end there is a linear layer and a softmax layer. [21, chapter 9]

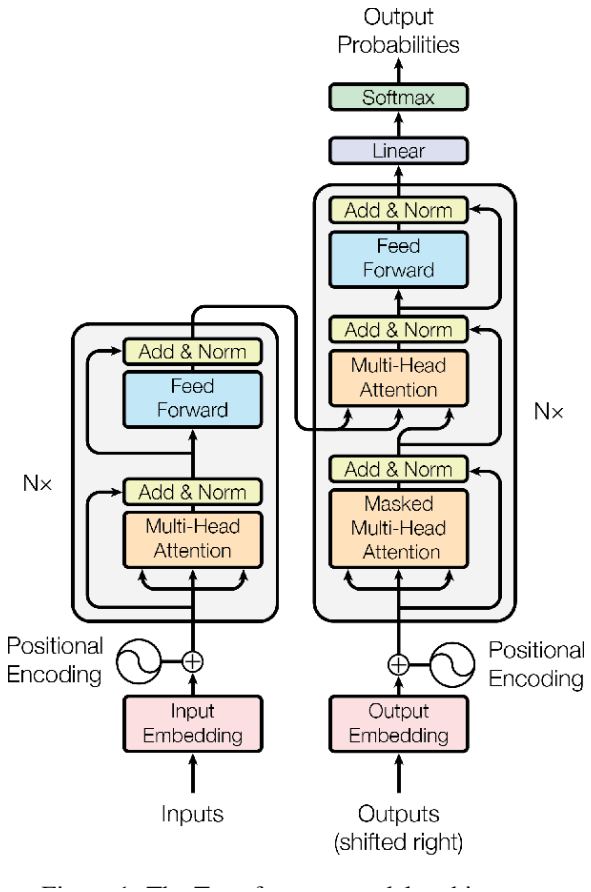


Figure 2.7.: Transformer model architecture Vaswani et. al 2017

2.2.3. Whisper

Whisper is a multilingual multitask Model that is focused on speech processing and was proposed in the Robust Speech Recognition via Large-Scale Weak Supervision paper [37]. Its architecture is based on a classical Transformer Architecture where the Transformer encoder Blocks consist of self attention blocks and Multilayer perception blocks. The Transformer decoder blocks use the learned position embeddings and tied input-output token representations. The encoder and decoder have the same number of transformer blocks. The audio pre-processing is done by making sure the audio chunks that are given to

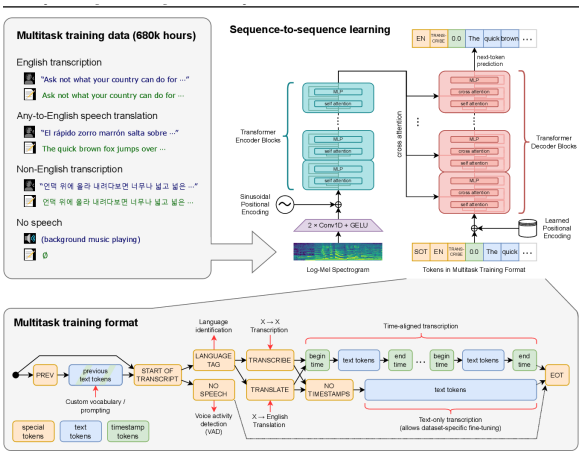


Figure 2.8.: Overview of the Whisper architecture Radford et. al 2022

the model are 30 seconds long, have ben sampled to 16,000 Hz, and have a 80-channel log-magnitude Mel spectrogram representation. This log-mel spectrogram is then computed with 25-millisecond windows and a stride of 10 milliseconds. If an audio chunk is shorter than 30 seconds it gets padded up to 30 seconds; if it is longer the audio gets split up. This input is then scaled to be between -1 and 1 and is put through 2 convolutional layers that have a filter width of 3 and use the GELU function as activation function. The second of those 2 layers has a stride of 2. The resulting data is then added to a sinusoidal position embedding, which is a positional encoding embedding that uses sinusoidal functions, so the sine and cosine [50], and then forwarded into the encoder blocks.

2.2.4. Seamless

Seamless [8] is a multilingual, multimodal model that uses a transformer architecture for the text translation part and, in the v2 version which is used in the experiments, it uses a w2v-Bert speech encoder which was pretrained on unlabelled audio data. The general architecture of Seamless is shown in Figure 2.9, for this thesis only the left half up to the Transformer Text decoder of the figure is relevant, the other parts of it are used for speech synthesis.

Seamless v2 improves upon the original model by introducing non-autoregressive text-to-unit decoding. However, this modification has minimal impact on this thesis since it primarily focuses on text-to-text (T2TT) and speech-to-text (S2TT) translation. The other change for v2 is that it uses a w2v-Bert 2.0 [7] encoder that is trained self-supervised on 4.5 million hours of unlabled audio, as compared to the previous 1 million hours of unlabeled data. SeamlessM4T has been trained on unlabeled, human-labeled, pseudo-labled, and automatically aligned data, where the text-to-text-translation (T2TT) was done on NLLB data [47]. This is a method of creating low-resource language datasets with a combination of using the Flores [20] and No Language Left Behind (NLLB) Seed dataset, which is a set of professionally translated sentences in the Wikipedia domain. The anything-to-text model, which can do T2TT, ASR, and Speech to Text translation (S2TT), is trained on different sources of S2TT data that is human-labeled, pseudo-labeled, and automatically

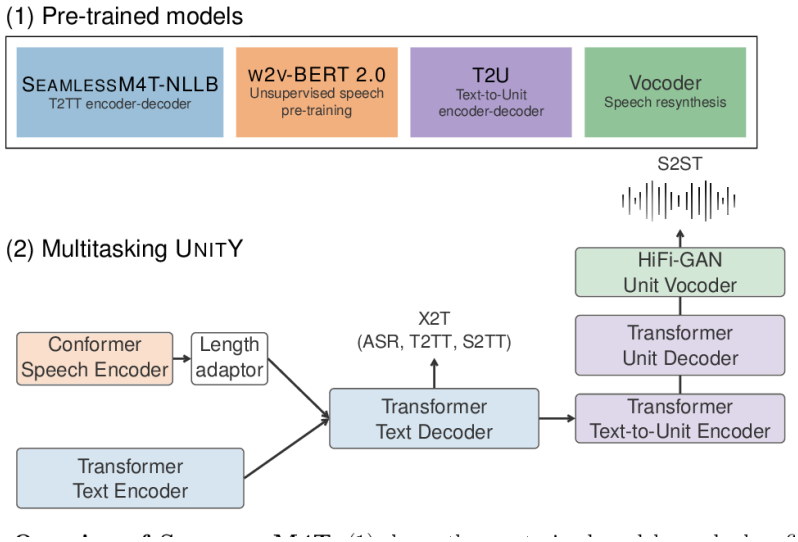


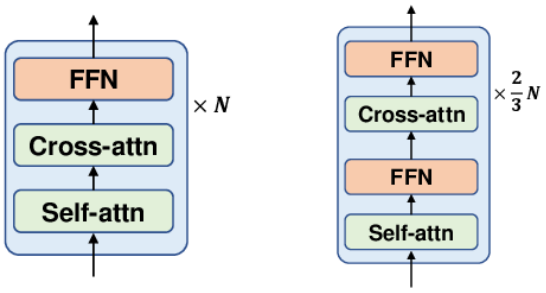
Figure 2.9.: Overview of SeamlessM4T. (1) shows the pre-trained models used when finetuning multitasking UnitY. (2) outlines multitasking UnitY with its two encoders, text decoder, T2U encoder-decoder, and the supporting vocoders for synthesizing output speech in S2ST. [8](figure 4)

aligned, and is a combines the v2w-Bert model and the Text encoder from the NLLB T2TT model and the corresponding decoder. It was trained in 2 steps on this data; the first one focuses on supervised English ASR and S2TT data where the target language is English. The 2. step in training then focuses on English to X S2TT and non-English ASR data.

The experiments were run on the v2 large version, both for the cascaded part and the end-to-end part of the experiments, no additional training was used.

2.2.5. SentencePiece

SentencePiece [23] is a tokenizer and detokenizer that allows for subword units, especially byte pair-encoding [40], that are language independent, as the sentences are treated as unicode character sequences and preprocessing is not always needed. It's comprised of a Normalizer, a Trainer, an encoder, and a decoder. The encoder uses the Normalizer to normalize the Test and then tokenizes the sentence. In the SentencePiece implementation the Decoding is considered the inverse operation of Encoding of normalized text. This results in a lossless tokenization, so there is no information loss over the process of encoding and decoding. To achieve this SentencePiece encodes white spaces with a meta-symbol that can be reverted. SentencePiece also manages the Vocabulary that is used in preprocessing as it also outputs a dictionary and can output a ID sequence to text and vice versa mapping. As the SentencePiece model is self-contained it also leads to better reproducibility as only the model file is needed, which is publicly available.



(a) Vanilla decoder (b) Interleaved decoder

Figure 2.10.: Vanilla Transformer decoder (left) compared to the interleaved Transformer decoder(right) from Ma et. al 2021

2.2.6. DeltaLM

DeltaLM [28] is one of the current state of the art Neural Machine Translation models. It is based off of the classical encoder-decoder structure but both the encoder and decoder are initialised with the pretrained multilingual encoder and then trained in a self-supervised manner. In classical pretrained encoder-decoder architectures only the encoder is initialized with a pretrained encoder and the decoder is initialized with random values.

In addition to this is the decoder a Interleaved Transformer decoder, which is not the same architecture as the encoder and differs from the standard Transformer decoder in that the Transformer blocks now consist of a self-attention layer, two feed-forward networks, and a cross-attention layer which are arranged as seen in Figure 2.10. This way of building the decoder is more similar to the structure of the encoder and makes it easier to leverage the pretrained encoder.

The interleaved decoder is then initialised with the layers from the pretrained encoder, which is the InfoXLM [5], in the following way: the self-attention and the bottom FFN layers are initialised with the odd layers of the InfoXLM encoder and the cross-attention and top FFN layers are initialised with the even layers. The leftover components of the decoder are also initialised the same as the pretrained encoder. This means that all of the sublayers are initialised with the pretrained weights and none of them use randomised values.

2.3. Evaluation metrics

For the evaluation of the results a couple of terms, algorithms, and methods are used. The explanation of them is in this section.

2.3.1. WER

The Word Error Rate, in short WER, has been proposed in [54] and [29]. It's based on the Levenshtein distance [26] but instead of working on phonemes it operates on words. The

WER can be computed as

$$WER = \frac{S + D + I}{N} = \frac{S + D + I}{S + D + C}$$

where S is the number of substitutions, D is the number of deletions, I is the number of insertions, N is the number of words in the reference, and C is the number of correct words.

2.3.2. Comet

Comet, in full Cross-lingual Optimized Metric for Evaluation of Translation, is a neural framework for machine translation evaluation that was proposed in [39] and improved in [38]. For this thesis the Estimator model is relevant, which is used to estimate a quality score based off of a source sentence and a translation reference. The estimator model architecture is pictured in Figure 2.11.

To do this the hypothesis translation, the source sentence, and the reference translation are encoded independently using a pre-trained encoder, that is a cross-lingual model like XLM [24], a multilingual BERT [11], or XLM-RoBERTa [9]. The resulting embeddings are padded into a pooling layer where they create a sentence embedding for each segment. The Embeddings are then combined and concatenated into a single vector that is passed into a regression layer, that then regresses on reference scores from Direct Assesment (DA) [19], MQM [27], and HTER [42]. This model is trained to minimize the Mean Squared error.

In the original implementation this does not give a score from 0 to 1 based on how good or bad the translation is, but the improved version does this, where a score of 1 means a perfect translation and a score of 0 means a bad translation. Besides this, only hyper-parameters have been changed in comparison to the original version, and more training has been done.

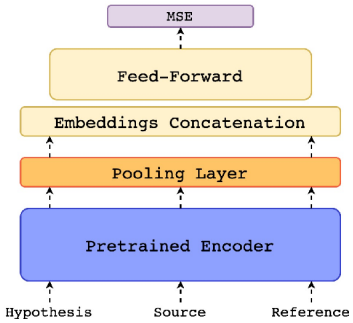


Figure 2.11.: overview of the comet estimator model architecture from [39]

2.3.3. Pearson-correlation

The Pearson correlation, or Pearson correlation coefficient (PCC), is a method to see how correlated 2 sets of values are. The r value is the covariance of the 2 sets divided by the

product of their standard deviations

$$PCC(x,y) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}}$$

where $\overline{x}, \overline{y}$ are the means of the sets of values, x_i, y_i are individual elements of the set of values, and n is the sample size. Due to this definition the possible values for the PCC are between -1 and 1, where a value of 1 means the sets are correlated, so a linear relation between the 2 value sets exists. A value of -1 is the result of the PCC if it's inversely correlated. If the value is 0 calculated this means the sets are not correlated at all. A graphical representation of these scores can be found in Figure 2.12. [36]

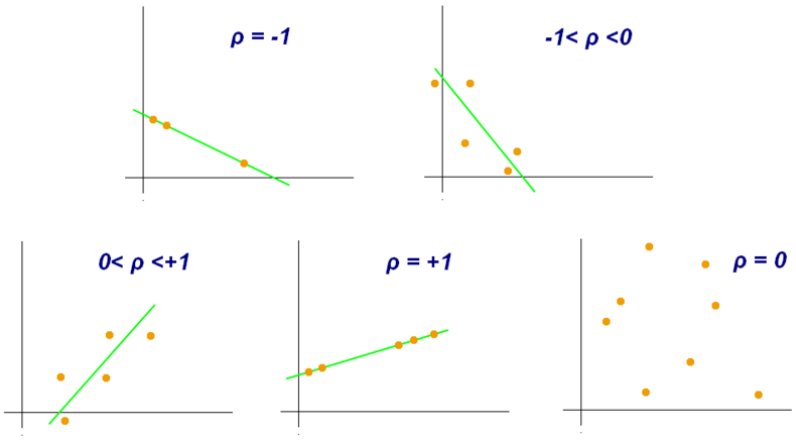


Figure 2.12.: graphical representation of different Pearson correlation scores [35]

The PCC has the advantage that it's symmetric, so switching the order of the inputs does not change the correlation score, but it has the drawback that it can be numerically unstable.

3. Related works

As far as I have found there hasn't been a lot of published work into estimating the quality of spoken language translation so far. The closest to what this thesis is doing has been done by Le et al. [25]. However they use a supervised approach that is based on Word confidence, so the classification of the confidence is on a word basis rather than sequence basis. Also, the translation step of the cascaded model they used is a statistical machine translation model instead of a neural machine translation model, which is no longer the state of the art machine translation, as neural machine translation models have gotten a lot better at translation in the years since the paper came out.

Negri et al.[30] take look at black box ASR quality estimation but have a small amount of white box quality estimation. Their white box methods are not designed for end-to-end ASR models, but they have used different features or information from both the acoustic and language models, and have combined information from both parts of the ASR models to be used in the regression part of the estimator. In their experiments they have shown that the regression based on white box features used to estimate the WER show quite the improvements over the baseline and the regression based on the black box features.

Analysing uncertainties with the help of dropout on Automatic Speech Recognition systems has been tries by Vyas et al. [52]. They look at uncertainty quantification, in particular WER estimation with the help of dropout. The experiments in this are done with ASR models that have an acoustic and a language model. They tried it with 2 different acoustic models. One is an HMM-DNN ASR model, another one is a Connectionist Temporal Classification acoustic model as the Language model they used a trigram LM. The dropout is only enabled on the acoustic models. The WER estimation metrics they propose in this make use of a non-dropout enabled transcript and N dropout enabled transcripts. They then calculate the pairwise edit distances between dropout transcripts, sort them in descending order, and take the mean of the top K of those edit distances, as well as the mean of the length of the decoding of those top K edit distances. As a reference they looked at N-best hypothesis of the decoding lattice to estimate the WER with the same estimation metrics. This has showed that this delivers good results for estimating WER on these kinds of of ASR models.

In a similar vein to Negri et al.[30] there have been works into predicting the WER. Fe-WER [34] is an improved version of the e-WER3 [6], which is a system to estimate the WER. It mainly improves computation performance without degrading the estimation performance. The e-WER3 system in turn is an improved version of the e-WER2 [3], which is a monolingual system, and e-WER [4], which compares both black and white box methods to estimate the WER. The e-WER3 is a framework for estimating multilingual ASR, which takes the audio and the automatic transcription of the ASR model to estimate the WER for that transcription. This works in a black box approach. A further improved

version of the Fe-WER has been proposed by Park et al. [33] where they also employed different Hypothesis generation strategies.

As mentioned in the introduction the Unsupervised Quality Estimation For Neural Machine translation paper [15] is the basis of most of this thesis. It takes a look at different ways to estimate quality estimation in machine translation, all of those in an unsupervised glass box manner. This means that all scores are based on data from within the model that can be gathered during inference and they do not need any reference data or other human input to create those scores in the first place. In addition to the metrics described in chapter 4, which are used in this thesis, they also looked at possible attention based metrics but found that this approach requires Direct Assessment to find the best head/layer which makes it not fully unsupervised.

Other works that have looked at quality estimation in translation include perturbation-based quality estimation of machine translation [12], which looks at word-level quality estimation of text translation on black box systems in an unsupervised manner by changing parts of the input text. Quality estimation without human-labeled data [48] explores generating synthetic data for quality estimation that can be used for word and sentence level estimation. Unbabels' entry for the WMT19 Translation Quality Estimation task [22], several submissions to the WMT19s Quality Estimation task (which can be found in [16]), as well as Sun et al.'s Exploratory Study on Multilingual Quality Estimation [45] use Direct Assessment or post-edit distance during training to estimate the quality. However such DA annotations are rare and really resource intensive to produce for spoken language, which is why this thesis focuses on unsupervised approaches.

4. Methodology

This chapter goes more in depth on how the Quality estimation scores are derived from data retrieved from the models in general, whereas chapter chapter 5 goes more in depth on how this was done for the specific models and some of the potential errors that were made in the experiments.

4.1. Previously proposed

The following methods have been proposed by Fomicheva et al [15] and are also employed here on the machine translation part of the cascaded models.

4.1.1. Translation probability

The translation probability is the probability a Machine translation model outputs the sequence $y = y_1, y_2 \dots y_n$ for the input $x = x_1, x_2 \dots x_n$. The probability is calculated by formula

$$TP = -\frac{1}{T}\sum_{t=1}^{T} log\ p(y_t)$$

where the probability of generating the sequence y is defined as

$$p(y) = p(y|x, \Theta) = \prod_{t=1}^{T} p(y_t|y_{<t}, x, \Theta)$$

where Θ is the model parameters. The probability $p(y_t|y_{<t}, x, \Theta)$ is the probability distribution after the decoding step of the t-th decoding step after applying the softmax. The $\frac{1}{T}$ is there to normalise the translation probability over the length of the translation sequence T as to minimize the effect of longer sequences getting a higher score when they shouldn't.

The general formula for the log-probability of the model generating a sequence is

$$TP = -\frac{1}{T}\sum_{t=1}^{T} log\ p(y_t)$$

where p is the log-probability of generating the t-th token in the output sequence. So to get the probability of the whole sequence log-probabilities the log-probabilities of each token are added together, and then normalized with the length of the sequence T. This formula can be directly derived from the formula given in subsection 2.1.1 by applying the logarithm. These log-probabilities are the probabilities after applying the softmax to the decoding probability distribution and then applying the logarithm to the highest probability.

4.1.2. Softmax Entropy

The Softmax entropy is the entropy of each element in the vocabulary at decoding step. This is a way to measure the uncertainty in the vocabulary at each decoding step. To compute the softmax entropy of a decoding step, the entropy (see subsection 2.1.6) for each element in the Vocabulary together is summed together. Then the sum of the entropy of all decoding steps is taken and normalised over the sequence length.

This results in the Formula:

$$\text{Softmax-Entropy} = -\frac{1}{T}\sum_{t=1}^{T}\sum_{v=1}^{V} p(y_t^v) log\ p(y_t^v)$$

where V is the Vocabulary size and T is the length of the generated sequence. The minus comes from the entropy and is only moved in front of the sums for ease of computation. Due to how entropy works a lower value, so one closer to 0, in the score is better. So if there are fewer entries in the vocabulary that have similar probabilities during a single decoding step, then the entropy of that decoding step will be lower.

4.2. Dropout

Dropout, as explained in subsection 2.1.7, aims to measure the uncertainty in a Deep Neural network. For this the same input is run N times through the model; due to the potential masking of neurons different results can be observed from the model. Based on how much these results differ from each other, and a reference that was obtained without dropout, conclusions can be made as to how certain the original result is. The following measures have been used in the past to minimize the effect of low quality outputs on neural machine translation training with back translation [53]. The dropout measures are used on the transcription and translation part of the cascaded models, as well as the end-to-end model.

4.2.1. General Probability

The dropout probability is the mean of the regular probabilities, as done in Table 6.1. For this the method described above is run on the model with dropout several times to get the Probability scores for each run. This results in the formula:

$$\text{D-TP} = \frac{1}{N}\sum_{n=1}^{N} TP_{\hat{\theta}n}$$

This method works to estimate the quality because if the masked neurons affect the result sequence and resulting probability will change, especially if the model is very uncertain about the resulting sequence. If the model is certain about the resulting sequence then masking neurons to 0 will not affect the resulting sequence and probability as much.

4.2.2. Variance

The dropout variance is the variance of the different probabilities gathered during the N runs. Mathematically this can be described as:

$$\text{D-Var} = E[TP_{\hat{\theta}}^2] - (E[TP_{\hat{\theta}}])^2$$

Where $TP_{\hat{\theta}}$ is the probability (see subsection 4.1.1) of the runs. If the Dropout Variance is high then the model is uncertain about the resulting sequence, and if the variance is closer to 0 it is quite certain about the sequence. So a low variance is to be considered better than a high variance.

4.2.3. Combo

As the variance does not take into account the probability of the sequence, a combination of the dropout Probability and dropout variance is proposed by Fomicheva et al [15]. The combination of the results from the probability and the variance is done by calculating

$$D - Combo = (1 - \frac{D - TP}{D - Var})$$

where $D - TP$ and $D - Var$ are the Translation probability mean (subsection 4.2.1) and the Dropout variance (subsection 4.2.2).

4.3. Proposed Methods

The following methods are derived from related works, changed from Fomicheva et al., or simply applied on different models than has been proposed in the past.

4.3.1. Transcription probability

The transcription probability is the probability that the ASR component transcribes the audio to the sequence of text $y_1 \dots y_n$. In encoder-decoder models this is most commonly done by encoding the audio signal in the encoder, using attention mechanisms to get the context for the current next output token, and then using previous predicted tokens to decode the current token. That next output token has a certain probability that, after applying the softmax to the whole probability distribution of the Vocabulary, is between 0 and 1. This probability is on the last layer of the decoder and retrieved from the model. This probability can be mathematically described in the formula

$$p(y|x, \Theta) = \prod_{t=1}^{T} p(y_t|y_{<t}, x, \Theta)$$

where Θ is the model parameters, x is the audio input sequence, and the softmax is used after every decoding step t on the resulting probability distribution $p(y_t|y_{<t}, x, \Theta)$. It is

also common practice to use the log probability instead of the raw probabilities after applying the softmax. This gives the total probability formula the form:

$$TP = -\frac{1}{T}\sum_{t=1}^{T} log\ p(y_t)$$

where p is the log-probability of generating the t-th token in the output sequence after applying the softmax. So to get the probability of the whole sequence log-probabilities, the log-probabilities of each token are added together, and then normalized with the length of the sequence T. These log-probabilities are the probabilities after applying the softmax to the decoding probability distribution and then applying the logarithm to the highest probability.

4.3.2. Standard Deviation

The standard deviation, similarly to the Softmax entropy, aims to measure the uncertainty by looking at the dispersion of the probabilities in the sequence. The idea of using the standard deviation was proposed by Fomicheva et al., however they use it on a word level whereas the experiments in this thesis employ it on a token level instead. This is done for ease of gathering the scores during the inference on some models, as it is easier to only work on the probabilities gathered, which are for tokens and therefore subwords, without having to recalculate with information from the sequence.

The mean that is the probability score does not account for the different behaviour that, for example [0.1,0.9] and [0.5,0.5] have, even though the have the same mean. To obtain this quality estimator the standard deviation over the top token at each decoding step of a sequence is computed. This means the mathematical formula is:

$$\text{Seq-Std} = \sqrt{\mathbf{E}[P^2] - (\mathbf{E}[P])^2}$$

where $P = p(y_1), \ldots p(y_T)$ is the token-level log-probabilities for the sequence.

4.3.3. Dropout on ASR systems

The Dropout metrics that have been proposed by Fomicheva et al. have also been used on the ASR system, exactly the same as it is used on the MT systems. This is possible because the structure of the model used is that of a Deep Neural Network.

This means that the Dropout probability for ASR can also be described as

$$\text{D-TP} = \frac{1}{N}\sum_{n=1}^{N} TP_{\hat{\theta}n}$$

The Variance of the transcription is described as

$$\text{D-Var} = E[TP_{\hat{\theta}}^2] - (E[TP_{\hat{\theta}}])^2$$

And the Combination of the 2 dropout metrics is also used, which is described by the formula

$$D - Combo = (1 - \frac{D - TP}{D - Var})$$

4.3.4. Unified Score

The unified score is a combination score for the transcription and translation part as an attempt to approximate the quality of the whole cascaded model. For this the translation probability and transcription probability are multiplied together as both the translation and transcription probabilities fall between 0 and 1.

$$\text{unified score}_{prod} = TP_{transcript} \cdot TP_{translation}$$

Another really naive variant would be to simply add the scores together

$$\text{unified score}_{sum} = TP_{transcript} + TP_{translation}$$

An alternative option for a unified sum score is weighing the translation and transcription probabilities differently. This is the case since transcription or translation errors might have a different impact on the quality and a weighted score should be able to represent this better. One way to do this with linear interpolation which is described the formula

$$unifiedscore_{\alpha} = (1-\alpha)TP_{transcript} \cdot (\alpha)TP_{translation}$$

.

4.3.5. Spoken language methods

The methods used for the end-to-end model are all the same as the Machine translation model. This means the employed methods are the translation Probability, calculated by the formula

$$TP = -\frac{1}{T}\sum_{t=1}^{T} log\ p(y_t)$$

The softmax entropy calculated by this formula:

$$\text{Softmax-Entropy} = -\frac{1}{T}\sum_{t=1}^{T}\sum_{v=1}^{V} p(y_t^v) log\ p(y_t^v)$$

The Standard deviation of the top token probabilities:

$$\text{Seq-Std} = \sqrt{\mathbf{E}[P^2] - (\mathbf{E}[P])^2}$$

As well as all of the dropout related methods like the Dropout Probability:

$$D - TP = \frac{1}{N}\sum_{n=1}^{N} TP_{\hat{\theta}n}$$

the Dropout variance:

$$\text{D-Var} = E[TP_{\hat{\theta}}^2] - (E[TP_{\hat{\theta}}])^2$$

and the combination out of those 2 scores:

$$D - Combo = (1 - \frac{D - TP}{D - Var})$$

5. Experiments

This chapter details the dataset and the experiments that have been run on Whisper [37], Seamless [8] and DeltaLM [28]. The experiments on whisper and seamless have been made with the help of the huggingface [1][13][1] models and frameworks, whereas DeltaLM[2] has been used with the fairseq toolkit [32][3].

For the cascaded models the transcription from the ASR model is passed into the translation model. In the case of the dropout quality estimators the decision of which transcript to put into translation model has been made based on the quality estimations of those transcriptions. One option is taking the transcript with the highest transcription probability mean to be the basis for the dropout of the translation. This is not the best way of obtaining the best transcription as the basis for the translations but it is a very good method if only the dropout part of the quality estimation is run. An example of this can be seen in Table 5.1. This is the case as there are some transcriptions with dropout that have a higher score at the end than the regular transcript would have, but have obvious signs of the dropout being enabled, like a lot of repeated letters. So taking the transcript with the highest score can propagate unwanted errors in the transcription to the translation section. The other and better method of obtaining a good transcript is running the transcription once without the dropout turned on, as that would result in the regular transcript that the model would output, but this method is fairly likely to return such a transcription due to the way of how the dropout is used. More on that is described in section 5.4. Running the transcription once more without using dropout should be considered as compared to the number of runs that is done for the dropout it does not add a lot more time to the runtime of a single sequence.

[1] huggingface can be found here: `https://huggingface.co/`, the whisper model documentation here: `https://huggingface.co/docs/transformers/model_doc/whisper` and the Seamless M4T v2 documentation here: `https://huggingface.co/docs/transformers/main/model_doc/seamless_m4t_v2`

[2] the code for DeltaLM can be found here: `https://github.com/microsoft/unilm/tree/master/deltalm`

[3] the documentation for fairseq can be found here: `https://fairseq.readthedocs.io/en/v0.10.2` and the github repo is found here: `https://github.com/facebookresearch/fairseq`

qe	transcript
-0.30165	Because that’s it.
-0.12921	Because that’s it.
-0.33879	No, no„ no, no, no, no, no, no, no, no, no, no, no, no, no, no, no, no, no, no, no, no, no, no, no, no, no, no, no„ no„ no, no, no„ no, no, no, no, no, no, no, no, no, no, no, no, no, no, no, no, no, no,. no, no, no, no, no, no, no, no, no
-0.094421	Not a model, not a replica.
-0.644232	Not a model, not a replica.

Table 5.1.: Example of differing transcript results, the top is an example where the transcript with dropout (first line) that has the highest quality estimation score is the same as the transcript without dropout. Compared to an example where the transcription withe highest score is vastly different from the non dropout transcription, 2. to last line. The last line is a different result from the dropout runs that is essentially the non dropout result. The left column is the corresponding quality score; the quality score shown is the transcription mean.

5.1. Dataset

The used dataset for all the scores is the benchmark section of the IWSLT2023 [43], which is a parallel dataset that consists of TED talks and the English transcriptions of those TED talks. It contains reference translations, as well as a segmentation for the transcriptions, which matches roughly the translation reference segmentation, and thus can be used in the evaluation without needing to resegment those before evaluating.

The benchmark section of the Dataset consists of 42 TED talks, each of which is the wav file that corresponds to the youtube videos, that result in 2255 segments.

5.1.1. Segmentation

Since most ASR models and end-to-end speech translation models only take audio up to a certain length as input, the given audio files have to be segemented into smaller chunks. The segmentation of the audio from the dataset will be done with the timestamps given in the dataset itself. Those timestamps segment the talks into segments that are less than 35 seconds long. This is especially useful since Whisper takes audio segments of 30 seconds. If a segment is longer than 30 seconds, Whisper uses a moving window and padding to transcribe the audio in 30 second chunks, the more precise splitting up and padding is taken care of by the preprocessor or tokenizer. The timestamps given in the dataset correspond to the source language transcription reference and the target language translation also correspond mostly well with the reference segments given in the dataset. The segments in the dataset were generated with the help of mwerSegmenter [4] for the IWSLT 2023, but it can also be used to align the model outputs to the references as it minimizes the word error rate between model translation and one or more references.

5.1.2. Training

The seamless and Whisper models are not trained or fine tuned for the experiments besides the pretraining that has been already done for the base models. The DeltaLM model however has been fine tuned using the training dataset from the IWSLT 2023 constrained category, specifically the English-German part of this.

[4] https: //www-i6.informatik.rwth-aachen.de/web/Software/mwerSegmenter.tar.gz

5.2. Models

More details on the models used in the experiments. More information on the implementations, like how the scores are retrieved in specific, can be found in section A.1. For Whisper and seamless the scores are retrieved with the help of functionality from huggingface or pytorch; for DeltaLM it's functionality from fairseq where available. More on that is found in the models section.

5.2.1. Whisper

The Transcription step is only done on the Whisper model. Open AI gives several different sizes for Whisper models. In this thesis the medium model is used, specifically the pretrained model that is available on huggingface, which provides a processor and a few different Whisper models that have a different head on top of it. The basic Whisper model on huggingface outputs the raw hidden states without a specific head on top of it. The specific model version used in the experiments is the WhisperForConditionalGeneration, as it has a language modelling head, and is recommended for automatic speech recognition. There are also ones that have heads for audio classification or a language modelling head that is a linear layer with weights tied to the input embeddings.

5.2.2. DeltaLM

The experiments [28] were done on a fine tuned large version of DeltaLM that was fine tuned using the training data from the IWSLT 2023 constrained category. For this only the English-German part was used.

The text was preprocessed with the pretrained SenctencePiece model and dictionary that has been provided on the DeltaLM github page [10]. After that it is preprocessed with faireq preprocess. The result of this is then put into fairseq generate with batch size and beam size 1.

Running the experiments was done with the help of the fairseq toolkit [32] which returns the probability of the specific translation along with the translation hypothesis, and prints the softmax probabilities of the top token after each decoding step. By default these probabilities are in the base 2 logarithm.

5.2.3. Seamless

There are a couple of difference model sizes of seamless that are provided by Meta. The one that was used to run the experiments is the seamless M4T v2 large model that is available on huggingface. Similarly to the Whisper model, the seamless models on huggingface also have several more specialized models. For the text translation the seamless v2 large model used is the SeamlessM4Tv2TextToText model. For the end-to-end translation the SeamlessM4Tv2SpeechToText model is used.

5.3. Dropoutless Experiments

The first of the experiments is to retrieve the non dropout scores from the different models. For the cascaded models this means retrieving or calculating both of the transcription scores, the translation score, the softmax entropy, and the standard deviation for the token probabilities. For the spoken language translation end-to-end model this means calculating the translation probability, the softmax entropy, and the standard deviation of the token probabilities.

To get the softmax entropy from the models the batch and beam size are set to 1 as otherwise the resulting tensors make it more difficult to pick out which is the right entropy for the resulting batch.

5.4. Dropout

To run the the dropout based experiments 30 forward passes with the same input are used on the models. Each time, neurons in the model are masked to 0 by some probability, which is usually the same probability as was used in training, or set to 0.1 when no information about that was available. The 0.1 dropout probability is used on both Whisper and seamless during training, so using the same probability on DeltaLM only makes sense.

Due to the nature of pytorch and huggingface models, the dropout has to be done in training mode, as the evaluation mode turns off any dropout layers that are in the model. Due to this and a bug in the implementation for caching during forward passes in the seamless models on huggingface, which leads to a tuple index out of range error that only appears in training mode with dropout turned on, the caching was turned off in the seamless configurations for the dropout based experiments. This means caching is turned off on both the text translation and speech translation system.

5.5. Unified Scores

Different unified scores are calculated with the scores that have been retrieved in the previous 2 experiments for the cascaded models. To calculate the scores the unified score methods proposed in subsection 4.3.4 are used. For this both the transcript probability score and the transcript mean score are paired with each of the regular translation scores. The gathered dropout scores are also combined, where the dropout transcription probability score and the dropout transcription mean are combined with the dropout translation probability, the dropout transcription variance scores are combined with the dropout translation variance score, and the dropout transcription combination scores are combined with the dropout translation combination score.

6. Evaluation

The evaluation of results of the experiments that were conducted on the benchmark section of the IWSLT 2023 dataset [43] with regards to section 5.1 as the relevant changes and preprocessing that was done.

The resulting scores are then pearson correlated [51] to comet[1] scores as well as word error rates and compare those. The used pearson correlation implementation is the huggingface implementation [2].

The comet scores are retrieved by using the regular translation from the MT models as well as the reference source transcription and the reference translation. These scores are used across all evaluation metrics in the MT category.

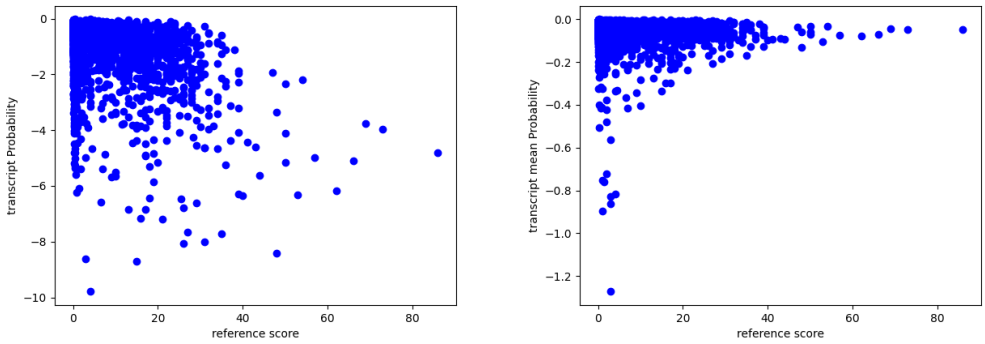
6.1. Transcription evaluation

To evaluate how good the transcription quality estimation is, the Word error rate[2] (see subsection 2.3.1) is used as reference score to compare the transcription quality estimation metric by correlating the WER scores with the help of the pearson correlation [51]. As WER is case sensitive, both the model result and the reference are normalised to be all lowercase, with single spaces, no leading or trailing blank spaces, and no punctuation.

The resulting transcription probabilities plotted over the according WER values are shown in Figure 6.1 and the pearson correlation can be found in Table 6.1. The plots in Figure 6.1 demonstrate quite nicely the impact of normalizing the scores with the sequence length. Figure 6.1a shows less of a trend with the high model scores and low WER scores than Figure 6.1b, which shows a more focused trend of higher WER scores and lower probability mean scores. However both methods show outlier values. For the transcription mean these are mostly low probability scores that have a low WER score as well, which means they are good transcripts. This also overlaps with the pearson correlation score, where the correlation score for the transcription mean is rounded 0.605, whereas the non mean correlation score for the transcription probability is 0.331.

This shows that taking the mean transcription probability is a good metric for estimating the transcription quality, as there are only a few outliers, and a transcription probability mean score close to 0 means it is highly likely that the WER score will also be low.

[1] the used comet implementation can be found here: https://github.com/Unbabel/COMET
[2] the used WER implementation can be found here: https://github.com/analyticsinmotion/werpy



(a) transcript scores over WER reference score (b) transcript mean over WER reference score

Figure 6.1.: plot over the transcription probabilities, the transcription means, and the WER scores

	Whisper
Transcription Probability (-)	−0.33164
Transcription mean (-)	−0.60496
Dropout transcription (-)	−0.2760
Dropout mean (-)	−0.3672
Dropout variance (+)	+0.1233
Dropout mean variance (+)	−0.0873
Dropout combo (+)	−0.1624
Dropout mean combo (+)	+0.2684

Table 6.1.: result from the transcription part of the cascaded model, correlated with WER scores, with the reference and model transcript normalized, the sign on the left denotes what sign a row should have

6.2. Translation evaluation

For the evaluation of the translation scores, the reference score is generated with the help of comet [39]. This way of generating the reference score means that a score close to 1 is a good translation and a score close to 0 is a bad translation that is no better than random chance.

Those reference scores are then also pearson correlated with the scores from the model. The correlated scores for the translation part of the cascaded model and the speech translation part of the end-to-end model are found in Table 6.2.

As can be seen in Table 6.2 the scores retrieved from the model correlate well on the cascaded models, with correlation scores of 0.376 for seamless and 0.283 for DeltaLM, and really well on the end to end model with a correlation score of 0.656.

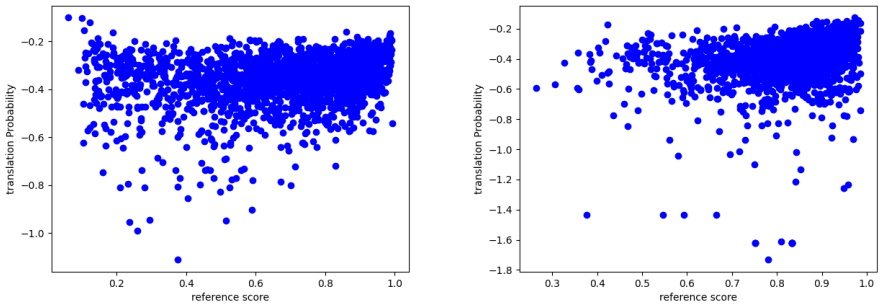
The drop on the DeltaLM model scores could be because of model differences with the seamless text-to-text translation model or due to the different score retrieval method; for this, more experiments would have to be run with different models and the different

	Seamless	DeltaLM	Seamless e2e
Translation	0.37592	0.28284	0.656299
Softmax Entropy (-)	0.30604	0.18071	0.60334
Standard deviation (-)	0.32905	0.25363	0.67148
Dropout translation	0.150755	0.282556	0.14194
Dropout Variance (-)	−0.106986	−0.16285	0.13080
Dropout combo (-)	−0.163593	0.179621	−0.20624
unified score$_{prod}$ (-)	0.32965	0.05348	
unified score mean$_{prod}$(-)	−0.50055	0.12587	
unified score$_{add}$	0.31837	0.06600	
unified score mean$_{add}$	0.56032	0.168710	

Table 6.2.: Correlation scores for the separate models and calculated quality scores. The sign on the left denotes whether the expected correlation value is supposed to positive or negative. If no signs are added to the values in a row, then all of the values have the expected sign.

toolkits. The drop could also be because even if the seamless model used for text-to-text translation only uses text-to-text translation, it might have learned different things during training that might help with translating. However, this can be put down to model differences in the end.

Another reason why the DeltaLM score on the translation probability is lower could be because it has more outliers with smaller translation probabilities but reference scores that are greater than 0.5, which can be seen in Figure 6.2. However what can also be seen from the scatter plots is that DeltaLM has fewer low reference scores than seamless has. The end-to-end scores are that much higher, with 0.665 to 0.377 or 0.283, since there are no potential errors from the transcription part, but it could also be that the model simply translates a lot better when using seamless for spoken language translation.



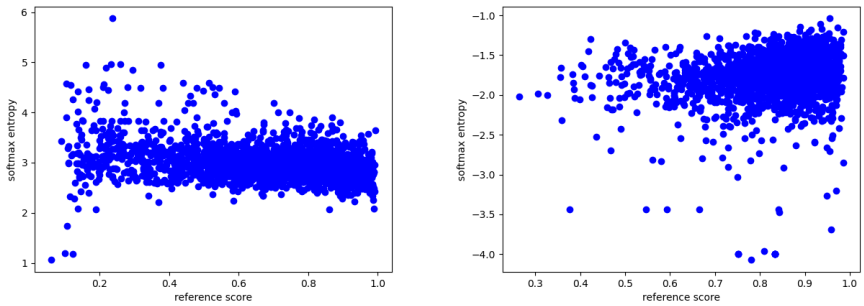
(a) spread of the translation probability scores on seamless over the reference scores

(b) spread of the translation probability scores on DeltaLM over the reference scores

Figure 6.2.: Model translation scores over the corresponding comet scores. The left side shows the seamless scores, the right side shows the DeltaLM scores.

The entropy scores anti-correlate with the reference scores. This is logical since the higher the entropy is, the worse the translation. When looking at the absolute value of the correlation, so ignoring the sign, it can be seen that they correlate less with the references than the translation scores, as there is a 0.07 difference on seamless for text and 0.053 difference for end-to-end translation, as well as a 0.1 difference on DeltaLM. They still correlate well with it, with correlation scores of 0.305 for t2t seamless, 0.180 for DeltaLM, and 0.603 for e2e seamless.

The sofmax entropy scores plotted over the reference comet scores are shown in Figure 6.3, which also show quite well why the seamless pearson correlation scores are so much higher than the DeltaLM correlation scores. This again might be because of the score retrieval method or it might be due to model differences. The end-to-end translation softmax entropy scores are once again higher than the text-to-text translation models. This is most likely again due to fewer possible errors from the transcription step.



(a) seamless entropy scores over comet scores

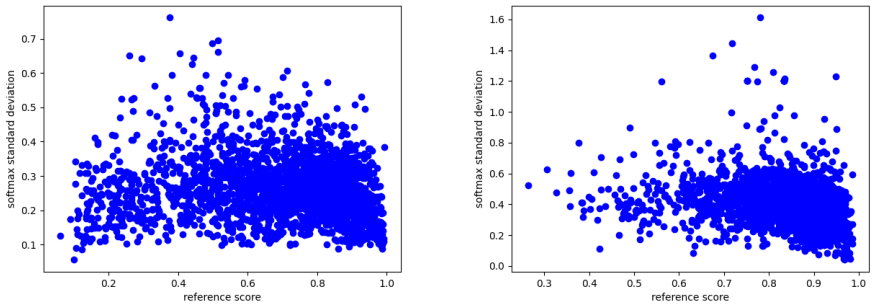
(b) DeltaLM entropy scores over comet scores

Figure 6.3.: Model softmax entropy scores over the corresponding comet scores. The left side shows the seamless scores, the right side shows the DeltaLM scores

The standard deviation scores anti-correlate, just like the entropy scores. This is also due to how the standard deviation works, as a lower standard deviation means less spread in the probabilities from the mean of those probabilities. Similarly to the softmax entropy, the absolute standard deviation correlation scores are also smaller than the translation scores, but they correlate more to the reference scores than the entropy scores, with correlations scores of 0.329 for t2t seamless, 0.254 for DeltaLM and 0.671.

Curiously enough the seamless end-to-end scores for the standard deviation have a higher correlation score than the translation score correlation with a correlation score of 0.671, which is most likely from the end-to-end nature of the model. This means the model does not have to use an intermediate transcript and thus generating each top token can have different behaviour than the text to text models. It could also stem from the multilingual part of seamless.

The standard deviation scores are plotted over the comet scores in Figure 6.4 which also shows well how the scores are anti-correlated. It also shows that the scores are very similar to how the regular translation probability scores look when plotted.



(a) seamless standard deviation scores over comet scores (b) DeltaLM standard deviation scores over comet scores

Figure 6.4.: Model standard deviation scores over the corresponding comet scores. The left side shows the seamless scores, the right side shows the DeltaLM scores

Based off of these results and possible error causes, it can be gathered that the best metric for quality estimation in text and spoken language translation is the translation probability. This is both because it is easy to implement in most models and frameworks or toolkits, where most of the time it is already included, and because it delivers good correlation results with reference scores, as explained above.

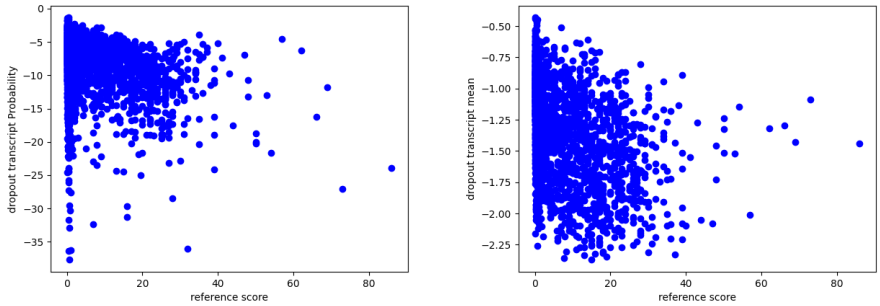
The next best metric would be the standard deviation of the decoding step probabilities. This is the case because it delivers good results in terms of correlation, which might also have more potential for new scores on end-to-end models. It is also fairly easy to implement the standard deviation, since getting the probability of the top token is what some toolkits or frameworks already allow. Compiling the different scores during translation is also not difficult, and applying the standard deviation to these scores, which is part of any maths library, is also simple.

The worst method of these 3 methods would be the softmax entropy. This is because if the toolkit does not have it implemented already or a way to get the probability distribution over the vocabulary at each decoding step, it can be difficult to implement, and it does not deliver scores that are better than any of the other scores proposed in this section. However the softmax entropy might be an interesting metric if used during dropout on text-to-text translation; more on that can be found in section A.2.

6.3. Dropout evaluation

The dropout score is calculated by taking the mean of the dropout probabilities of the model, the variance of the dropout probabilities, and a combination of both. The dropout score is then Pearson correlated with the comet scores or the word error scores in the case of the transcription. The reference scores are computed with the non dropout transcriptions and translations, since the dropped out sequences can differ a lot from the non-dropout sequences, which would impact the reference score and not accurately represent the reference. The correlation results are listed in Table 6.2 for the translation part and Table 6.1 for the transcription dropout.

Figure 6.5 shows the differences between the transcription probabilities and transcription mean in dropout. As can be seen, if the scores are not normalized there is a lot less clear correlation between the low word error scores and high scores, as there are a lot more scores that have a low WER score but a broad spectrum of probability scores. This reflects in the correlation scores; the scores gathered with the mean of the transcription have a correlation score of 0.367 where the not normalised scores only have a correlation of 0.276.



(a) dropout transcription probability scores over the reference scores

(b) dropout transcription mean scores over the reference scores

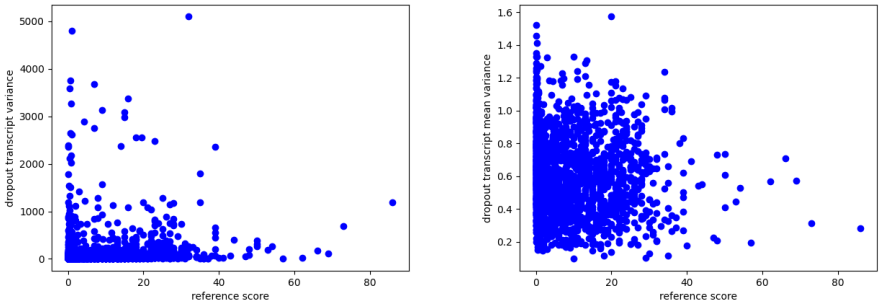
Figure 6.5.: transcription dropout scores plotted over the WER; left is the base scores, right is the mean scores

The variance correlation is a lot higher in the non mean dropout with a correlation score of 0.123 than it is for the mean probability dropout with a score of 0.087. This is due to a lot more variance in the length of the resulting transcripts and the impact of this variation in length on the score. The variance of the mean scores in dropout is a lot smaller; this is most likely due to the normalisation and a resulting smaller impact on the scores. It should also be noted that the transcription mean variance correlation score is quite close to 0 and because of this not really statistically significant. A plot of these values can be found in Figure 6.6, which also shows how not correlated the transcription mean dropout variance scores are to the WER.

The combination score, which consists of the dropout transcription score and the dropout variance score, once again show decent correlation with the WER scores. There are correlation scores of 0.162 for the non normalized scores and 0.268 for the normalised scores. These are lower than the dropout transcription score correlation by about 0.1 for each, but with how much lower the variance correlations are this is to be expected.

The sign difference in the correlation scores comes most likely from the sign difference in the variance and the formula of how the combination score is calculated at each step, as well as the fact that the non normalized transcription variance scores are significantly higher than the variance values of the mean which leads to smaller values in the score.

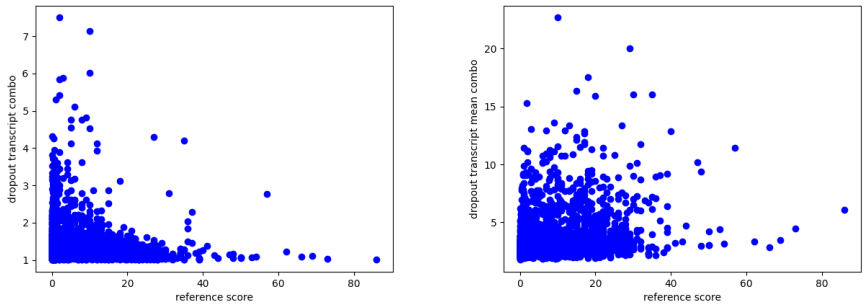
The plotted combo scores both for the transcription dropout scores and transcription mean dropout scores can found in Figure 6.7. They once again show that the mean displays a bit better distinction around the WER score of 0 than the transcription score.



(a) spread of the dropout transcription variance scores over the reference scores

(b) spread of the dropout transcription mean variance scores over the reference scores

Figure 6.6.: transcription dropout variance scores plotted over the wer, left is the base scores, right half is the mean scores



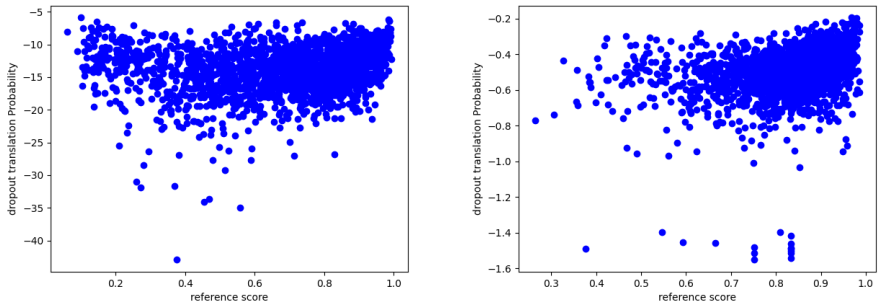
(a) spread of the dropout transcription combination scores over the reference scores

(b) spread of the dropout transcription mean combination scores over the reference scores

Figure 6.7.: transcription dropout scores plotted over the wer, left is the base scores, right half is the mean scores

The baseline for the dropout is the dropout translation, found under that name in Table Table 6.2. As can be seen the correlation between the seamless results, both the translation only with a score of 0.151 and the end to end versions with a correlation score of 0.142, is lower than the ones from DeltaLM, which has a correlation score of 0.282.

This is most likely due to how the dropout is implemented or used with the different toolsets/frameworks, but it could also be due to model differences. To properly distinguish between those possibilities more experiments would be needed where dropout is enabled only on different parts of the model, so for example just the decoder, just the encoder, including attention and the like. Alternatively it might also stem from different versions of the input into the dropout part of seamless, but that would not explain the drastically lower score on the end to end version of seamless, since that takes the audio as an input. A plot of these scores can be seen in Figure 6.8.



(a) spread of the dropout translation probability scores over the reference scores for seamless

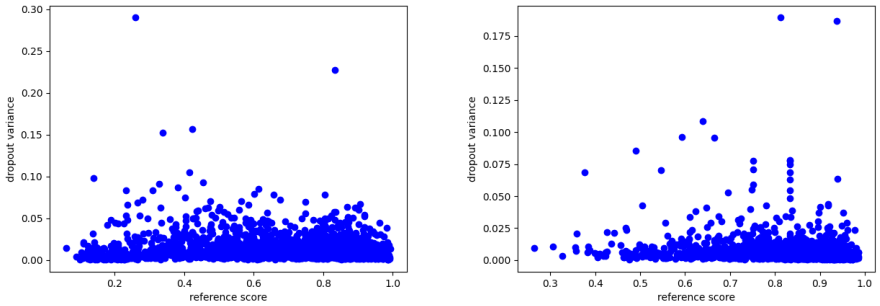
(b) spread of the dropout translation probability scores over the reference scores for DeltaLM

Figure 6.8.: dropout probability scores plotted over the comet scores, left is the seamless scores, right half is the DeltaLM scores

The variance scores are all smaller than the dropout probability scores, which as seen in the transcription part is to be expected. While the pearson correlation of the translation dropout variance scores are smaller than the pearson correlation of the dropout translation probability score, they are not that much smaller than them. The pearson correlation score can be found in Table 6.2 and a plot of the dropout variance scores over the reference scores can be seen in Figure 6.9.

The variance score is once again anti-correlated to the comet score. This is because of of how the variance works, where a smaller score denotes a better result. The end-to-end dropout variance is not anti correlated, which can very well be due to a difference in reference score, or can be a result of dropout. As using dropout with the end-to-end models sometimes produces long stings of nonsense, and because of how seamless is trained and what it can translate, those long strings do not always stick to a single language but rather switch through different ones and occasionally include characters from different alphabets.

The combination score made up out of the the other dropout scores shows a stronger correlation with the reference scores than the variance, but less than the translation



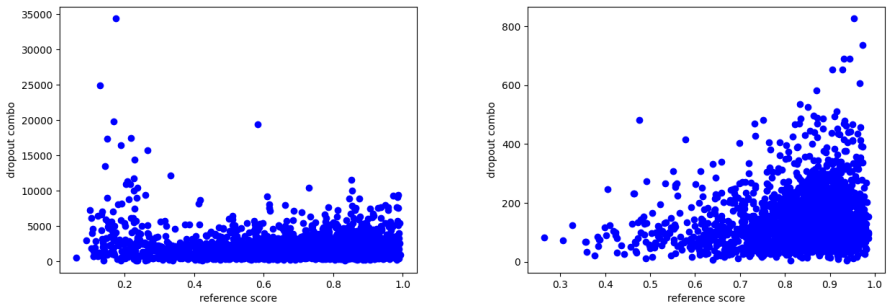
(a) spread of the dropout translation probability variance scores over the reference scores for seamless

(b) spread of the dropout translation probability variance scores over the reference scores for DeltaLM

Figure 6.9.: dropout variance scores plotted over the comet scores, left is the seamless scores, right half is the DeltaLM scores

probability for the cascaded models, with scores of -0.163 for seamless t2t and 0.180 for DeltaLM. Considering how it is defined this is to be expected.

The flipped sign on the correlation score for DeltaLM is most likely due to the incredibly small dropout variance scores and since the dropout translation score, which is negative, is divided by this variance score and then subtracted from 1. However it is stronger correlated in the case of the end to end model with a correlation score of -0.206. This is most likely due to the correlation instead of anti correlation in the variance and the very similar correlation score between the dropout translation and dropout variance scores. The overview of the pearson correlation scores again can be found in Table 6.2 and a plot of the scores for seamless t2t and DeltaLM over the reference scores can be found in Figure 6.10.



(a) spread of the dropout translation probability combo scores over the reference scores

(b) spread of the dropout translation probability combo scores over the reference scores

Figure 6.10.: Dropout combo scores plotted over the comet scores; left is the seamless scores, right is the DeltaLM scores

6.4. One unified score

Since several different metrics are used in the translation part of the cascaded model it is interesting which one might be the best choice to use in a unified score. From the text translation paper [15] we can gather that different metrics work better for different language groups. As this thesis only tested on English to German translation, no definite choice can be made without looking at other languages as well.

For a unified score the baselines, which are the translation and transcription probability, are multiplied. The resulting pearson score that can be found in Table 6.2. It is calculated by correlating this unified score, the product of translation probability and transcription probability, with the unified reference score. The unified reference score is calculated by multiplying the comet score with the WER mapped to be from 0 to 1. This is done by dividing all WER scores by the worst, so highest, WER score in the dataset. If finding the highest WER score for the dataset is not possible, using 100 as a value is a decent option, as most WER scores are below that, and subtracting it from 1. This is can be described mathematically as

$$reference = cometscore * (1 - \frac{WER}{max(WER)})$$

which was chosen since the transcription probability is anti-correlated to the WER, and since the best score of the WER is 0 that had to be taken into account as it ruled out just flat out dividing the comet score by the WER.

	Seamless+base	DeltaLM+Whisper	Seamless+mean	DeltaLM+mean
translation (-)	−0.32965	−0.05348	−0.50055	0.12587
softmax entropy (-)	0.3703422	−0.132959	0.558637	−0.129452
standard deviation	0.2670221	0.158974	0.4710036	0.1405853
dropout probability	−0.368084	−0.1526369	−0.465468	−0.227128
dropout variance	−0.134160	−0.048070	0.021499	−0.159075
combo	0.01635226	0.1689276	−0.296981	0.138246

Table 6.3.: pearson correlations of multiplicative scores for various other metrics. The columns are separated by whether the transcript probability, denoted with base, or the transcription mean was used to calculate the score, as well as the translation model.

The correlation scores for the multiplication version of the unified scores are shown shown in Table 6.3. The translation probability based score, which can be seen as a baseline for the other correlation scores, has a correlation of 0.329 for the non normalized transcription scores and a score of 0.501 with the normalized scores on seamless. Whereas the unified scores calculated with the translation probability scores from DeltaLM have a much lower correlation score with the reference score of only 0.053 with non normalized transcription scores and a correlation of 0.126 if the normalized transcription scores are used.

The softmax entropy based score has the highest correlation of 0.559 on seamless for the mean score and a correlation score of 0.37 for the transcription probability, whereas

the DeltaLM correlation scores are 0.133 for the non normalized score and 0.13 for the transcription mean score. The difference in correlation between the models is most likely due to the difference in correlation scores of the softmax entropy scores in the first place. The higher correlation than the baseline with the translation scores is most likely because of how the reference score is calculated but it could also be a difference in the entropy scores that the different models produce.

The standard deviation correlation scores are lower than the translation correlation for seamless with correlation scores of 0.267 and 0.471. In the case of DeltaLM the correlation scores are higher, such as the score that was calculated with the transcription probability going up by 0.1 in correlation and the transcription mean unified score going up by 0.02. These higher scores are most likely due to the anti-correlation property of the standard deviation score in the MT part of the model.

The dropout based scores show a very similar behaviour as the dropout of the different components, where the dropout probability has a correlation score that is similar to the non dropout correlation score, in this case scores of 0.36 and 0.46 for seamless and 0.15 and 0.23 for DeltaLM. The variance scores are worse or significantly worse scores.

The really interesting thing is the fact that dropout changes which transcription score gives a bad correlation score in combination with the translation model. The scores retrieved from seamless and multiplied with the transcription mean are significantly worse, with a correlation of 0.021, at predicting the reference score than if it was multiplied with the non normalised transcription probability, which has a correlation of 0.13. On DeltaLM this is the other way around, and the non mean score only has a correlation of 0.04, whereas with the variance of the mean transcription it has a correlation of 0.15. Because of how the combination score is calculated this also reflects in the correlations of those scores, where the bad correlation scores on the variance create better correlation scores in the combo score and the other way around for the good correlation scores.

This once again shows that the seamless scores are on average about 2 times more correlated than the DeltaLM scores. It also shows that using the transcription mean in scores like this gives better results all around, and that using the dropout probability, the translation probability, and the standard deviation of the token probability to estimate the quality are all good metrics that can be calculated in a reasonable time frame.

	Seamless+base	DeltaLM+base	Seamless+mean	DeltaLM+mean
translation	0.31837	0.06600	0.56032	0.168710
softmax entropy	0.171143	0.099865	0.1048499	0.205844
standard deviation	0.284749	−0.010293	0.497750	−0.057896
dropout probability	0.405961	0.042960	0.234955	0.107331
dropout variance	−0.242118	−0.028931	0.325615	0.019440
combo	−0.090351	0.189254	−0.090917	0.188630

Table 6.4.: pearson correlation scores that are added together from the different scores in the translation transcription categories. The columns are separated by whether the transcript probability, denoted with base, or the transcription mean was used to calculate the score, as well as the translation model.

	Seamless+base	DeltaLM+base	Seamless+mean	DeltaLM+mean
softmax entropy	−0.377987	−0.099865	−0.513472	−0.205843
standard deviation	−0.306851	−0.058635	−0.545197	−0.148497
dropout probability	−0.405961	−0.042960	−0.234955	−0.107331
dropout variance	−0.242118	−0.0289317	0.325615	0.019441
combo	−0.090351	0.189254	−0.090917	0.188630

Table 6.5.: correlation scores of addition unified scores where the absolute of input scores has been taken

The correlation scores for the sums of the scores are shown in Table 6.4. For the summed unified scores, very similar correlation scores are achieved when using the translation probability, with the main difference that it is correlated instead of anti correlated for all of them. Seamless also again has the higher correlation scores with 0.328 for the not normalized transcription unified score and a correlation of 0.560 for the score that uses the length normalized transcription score. Whereas DeltaLM has a correlation score of 0.066 for the non normalized score and a score of 0.169 for the normalized one, which is 0.04 higher than the multiplied score. A plot of these scores can be seen in Figure 6.11. This difference in scores could very well be because of a difference in the original scores or sign differences in the scores.

If the raw softmax entropy is used in the calculation of the unified score the scores look quite different. For seamless the non normalized transcription version of the score has a higher correlation than if the mean was used, whereas for DeltaLM the opposite is the case. However the fact that the correlation scores are quite similar between the translation and softmax based scores might indicate that this might be due to a difference in the model scores, or the signs of the scores. If the absolute of the scores is taken then the correlation scores are 0.378 with no transcript score normalization, 0.513 with transcript score normalisation on seamless, and the same for DeltaLM.

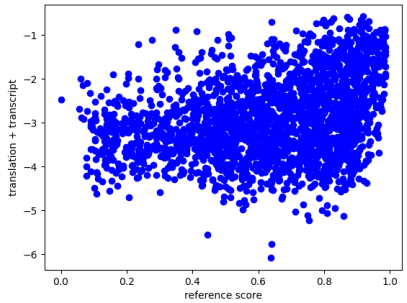
For the standard deviation based scores the correlation scores follow the same trend of the seamless scores being higher than the DeltaLM scores. Adding together the raw scores gives correlation scores of 0.28 and 0.49 for seamless, and on DeltaLM the scores are 0.01 and 0.05. However taking a look at the correlation of the absolute input scores this again gives better correlation results of 0.059 with transcription probability and 0.148 with the transcription mean. This is the case because the separate scores have different signs and the scores are in the same size range absolutely seen, especially the transcription mean. The seamless correlation scores also increase when taking the absolute to 0.306 and 0.545 respectively.

The dropout based scores show essentially the same effects as when multiplied where the seamless scores are a lot higher in correlation. In this case the non normalized dropout probability score is higher correlated than the mean dropout probability score, with correlation scores of 0.4 and 0.23 respectively. The correlation scores for DeltaLM are 0.04 on the non normalized scores and 0.107 on the normalized scores, which is most likely due to the same effects as on the translation probability version of the score, where they are simply not a good estimation and as has been seen before the dropout correlation is

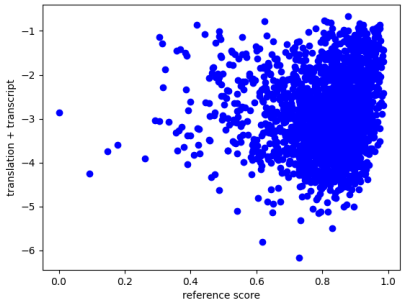
usually lower than the baseline anyway. The scores are also the same whether the raw scores are taken or the absolute. The dropout variance correlation scores show a similar behaviour but for seamless the mean score once again has the higher correlation whereas for DeltaLM the non normalized scores have a higher correlation. However it should be noted that the dropout variance scores for DeltaLM are very low with 0.029 and 0.019.

The correlation of dropout combination score shows that if both of the previous scores are well correlated the resulting scores are not correlated well and vice versa. This means that the seamless based scores are close to 0, with scores of 0.09 in both versions, whereas the DeltaLM based scores have a correlation scores of 0.189 each.

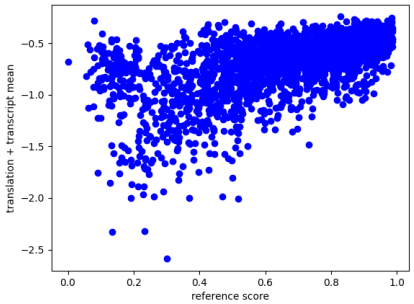
This shows that simply adding the scores gathered together is a possible way to get a unified score. However due to different signs in scores, the absolutes should be used instead of the raw scores. And similarly to the version above the translation and standard deviation based scores are among the best correlated and seem to be the most reliable when used with the transcription mean.



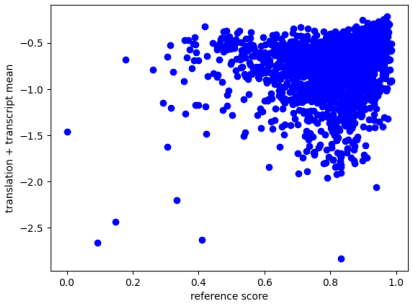
(a) seamless unified scores with non normalized transcription scores



(b) DeltaLM unified score with non normalized transcription scores



(c) seamless unified scores with normalized transcription scores



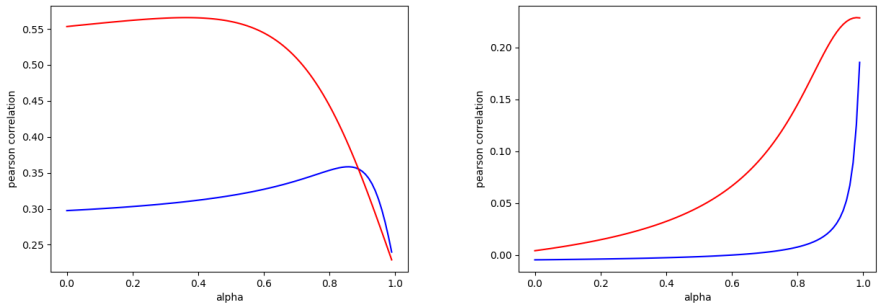
(d) DeltaLM unified scores with normalized transcription scores

Figure 6.11.: Plot of the unified scored calculated by adding the translation and transcription scores together, a) and b) are calculated with the non normalized transcription scores, c) and d) with the normalized transcription scores

As for the linear interpolated a unified score made up of the translation and transcription probabilities differently, which is given in the formula

$$unifiedscore_{\alpha} = \alpha TP_{transcript} + (1 - \alpha) TP_{translation}$$

. As can be seen in Figure 6.12 changing the weight between the transcription and the translation scores has a noticeable impact on the correlation scores. This also shows that the weighing for cascaded models is at least partially model dependent, if not framework dependent, but using a high α seems to yield good results on the non-normalised unified scores in either case. For the normalised unified score the correlations seem to differ on what the best α value would be; for seamless it would be in the 0.3 to 0.4 range, whereas for DeltaLM it is around 0.98.



(a) seamless unified scores (b) DeltaLM unified scores

Figure 6.12.: Changes in correlation with different alpha values: the red line is for the unified score taken with the transcription mean, the blue line is the unified score gathered with the the non-normalised transcription score.

This shows that using the translation and length normalized transcription probabilities from cascaded models is a viable quality estimation, no matter if the translation and transcription scores are added together, multiplied, or linearly interpolated for the score calculation. It also shows that finding the right weights for the transcription and translation parts can improve the correlation even more than simply adding or multiplying them together, but to find the right weights testing would have to be done, as it seems to differ on a model basis.

When compared to the end-to-end model these scores compare really well in terms of correlation, especially the added together scores where the transcription mean has been used.

7. Conclusion

The proposed metrics by Fomicheva et. al [15] also work in cascaded Speech translation systems. For this the translation quality estimation methods, like softmax entropy and standard deviation calculation of token probabilites, are used on the translation side of the cascaded model, and probability retrieval based quality estimation methods are used on the transcription part.

The dropout based metrics, like the dropout translation and transcription probability as well as the combination score of the probability and variance, also deliver good correlation results with the reference scores for both the transcription part of cascaded models as well as the translation part. It also works on end-to-end speech translation.

Two unified score versions for cascaded models has been proposed and tried with the separate metrics as the basis for the single score. Several of these have been shown to be a well correlated metric to reference scores, for which a formula also has been proposed. The main ones that are simple to implement and well correlated are the transcript probability mean on the ASR side and the translation probability, as well as the standard deviation of the token probabilities on the MT side of the model.

An additional version of the unified score has also been proposed where the different scores are weighed differently in the computation. However, finding a good value for a general value of this would require more testing with several more models and frameworks or toolkits. To explore the impact of the frameworks or toolkits used, as well as the impact of the models used on this score further experiments would be needed.

Future works could take a closer look at these metrics for different spoken language pairs, as it has been show in the past that these metrics perform even better on low and mid resource languages. Further work could take a look at combining multiple of these scores to see how well this estimates the quality both for MT and for ST, as well as explore the behaviour of methods like the softmax entropy and standard deviation of token probabilities during dropout and the quality estimation property of these.

Bibliography

[1] documentation for the huggingface seamless model. URL: https://huggingface.co/docs/transformers/main/model%5C_doc/seamless%5C_m4t%5C_v2 (visited on 12/21/2024).

[2] documentation of the huggingface pearson functionality. URL: ttps://huggingface.co/spaces/evaluate-metric/pearsonr (visited on 12/21/2024).

[3] Ahmed Ali and Steve Renals. *Word Error Rate Estimation Without ASR Output: e-WER2*. 2020. arXiv: 2008.03403 [eess.AS]. URL: https://arxiv.org/abs/2008.03403.

[4] Ahmed M. Ali and Steve Renals. "Word Error Rate Estimation for Speech Recognition: e-WER". In: *Annual Meeting of the Association for Computational Linguistics*. 2018. URL: https://api.semanticscholar.org/CorpusID:215542279.

[5] Zewen Chi et al. *InfoXLM: An Information-Theoretic Framework for Cross-Lingual Language Model Pre-Training*. 2021. arXiv: 2007.07834 [cs.CL]. URL: https://arxiv.org/abs/2007.07834.

[6] Shammur Absar Chowdhury and Ahmed Ali. "Multilingual Word Error Rate Estimation: E-Wer3". In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2023, pp. 1–5. DOI: 10.1109/ICASSP49357.2023.10095888.

[7] Yu-An Chung et al. *W2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training*. 2021. arXiv: 2108.06209 [cs.LG]. URL: https://arxiv.org/abs/2108.06209.

[8] Seamless Communication et al. *SeamlessM4T: Massively Multilingual & Multimodal Machine Translation*. 2023. arXiv: 2308.11596 [cs.CL]. URL: https://arxiv.org/abs/2308.11596.

[9] Alexis Conneau et al. *Unsupervised Cross-lingual Representation Learning at Scale*. 2020. arXiv: 1911.02116 [cs.CL]. URL: https://arxiv.org/abs/1911.02116.

[10] *deltalm github page*. URL: https://github.com/microsoft/unilm/tree/master/deltalm.

[11] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL]. URL: https://arxiv.org/abs/1810.04805.

[12] Tu Anh Dinh and Jan Niehues. *Perturbation-based QE: An Explainable, Unsupervised Word-level Quality Estimation Method for Blackbox Machine Translation*. 2023. arXiv: 2305.07457 [cs.CL]. URL: https://arxiv.org/abs/2305.07457.

[13] *documentation of the huggingface whisper model*. URL: ttps://huggingface.co/docs/transformers/model%5C_doc/whisper (visited on 12/21/2024).

[14] Mael Fabien. *Introduction to Automatic speech recogintion (ASR)*. URL: https://maelfabien.github.io/machinelearning/speech_reco/ (visited on 01/02/2025).

[15] Marina Fomicheva et al. *Unsupervised Quality Estimation for Neural Machine Translation*. 2020. arXiv: 2005.10608 [cs.CL].

[16] Erick Fonseca et al. "Findings of the WMT 2019 Shared Tasks on Quality Estimation". In: *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*. Ed. by Ondřej Bojar et al. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 1–10. DOI: 10.18653/v1/W19-5401. URL: https://aclanthology.org/W19-5401/.

[17] Yarin Gal and Zoubin Ghahramani. *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning*. 2016. arXiv: 1506.02142 [stat.ML]. URL: https://arxiv.org/abs/1506.02142.

[18] Jakob Gawlikowski et al. *A Survey of Uncertainty in Deep Neural Networks*. 2022. arXiv: 2107.03342 [cs.LG]. URL: https://arxiv.org/abs/2107.03342.

[19] Yvette Graham et al. "Continuous Measurement Scales in Human Evaluation of Machine Translation". In: *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*. Ed. by Antonio Pareja-Lora, Maria Liakata, and Stefanie Dipper. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 33–41. URL: https://aclanthology.org/W13-2305/.

[20] Francisco Guzmán et al. *The FLoRes Evaluation Datasets for Low-Resource Machine Translation: Nepali-English and Sinhala-English*. 2019. arXiv: 1902.01382 [cs.CL]. URL: https://arxiv.org/abs/1902.01382.

[21] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3rd. Online manuscript released August 20, 2024. 2024. URL: https://web.stanford.edu/~jurafsky/slp3/.

[22] Fabio Kepler et al. "Unbabel's Participation in the WMT19 Translation Quality Estimation Shared Task". In: *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*. Ed. by Ondřej Bojar et al. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 78–84. DOI: 10.18653/v1/W19-5406. URL: https://aclanthology.org/W19-5406/.

[23] Taku Kudo and John Richardson. "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*. Ed. by Eduardo Blanco and Wei Lu. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 66–71. DOI: 10.18653/v1/D18-2012. URL: https://aclanthology.org/D18-2012.

[24] Guillaume Lample and Alexis Conneau. *Cross-lingual Language Model Pretraining*. 2019. arXiv: 1901.07291 [cs.CL]. URL: https://arxiv.org/abs/1901.07291.

[25] Ngoc-Tien Le, Benjamin Lecouteux, and Laurent Besacier. *Automatic Quality Assessment for Speech Translation Using Joint ASR and MT Features*. 2016. arXiv: 1609.06049 [cs.CL].

[26] Vladimir I. Levenshtein. "Binary codes capable of correcting deletions, insertions, and reversals". In: *Soviet physics. Doklady* 10 (1965), pp. 707–710. URL: https://api.semanticscholar.org/CorpusID:60827152.

[27] Arle Lommel, Aljoscha Burchardt, and Hans Uszkoreit. "Multidimensional Quality Metrics (MQM): A Framework for Declaring and Describing Translation Quality Metrics". In: *Tradumàtica: tecnologies de la traducció* 0 (Dec. 2014), pp. 455–463. DOI: 10.5565/rev/tradumatica.77.

[28] Shuming Ma et al. *DeltaLM: Encoder-Decoder Pre-training for Language Generation and Translation by Augmenting Pretrained Multilingual Encoders*. 2021. arXiv: 2106.13736 [cs.CL].

[29] Andrew Morris, Viktoria Maier, and Phil Green. "From WER and RIL to MER and WIL: improved evaluation measures for connected speech recognition." In: Jan. 2004.

[30] Matteo Negri et al. "Quality Estimation for Automatic Speech Recognition". In: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Ed. by Junichi Tsujii and Jan Hajic. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, Aug. 2014, pp. 1813–1823. URL: https://aclanthology.org/C14-1171.

[31] *Numpy standard deviation reference*. URL: https://numpy.org/doc/stable/reference/generated/numpy.std.html (visited on 01/02/2025).

[32] Myle Ott et al. *fairseq: A Fast, Extensible Toolkit for Sequence Modeling*. 2019. arXiv: 1904.01038 [cs.CL]. URL: https://arxiv.org/abs/1904.01038.

[33] Chanho Park, Mingjie Chen, and Thomas Hain. *Automatic Speech Recognition System-Independent Word Error Rate Estimation*. 2024. arXiv: 2404.16743 [cs.CL]. URL: https://arxiv.org/abs/2404.16743.

[34] Chanho Park et al. *Fast Word Error Rate Estimation Using Self-Supervised Representations For Speech And Text*. 2023. arXiv: 2310.08225 [eess.AS]. URL: https://arxiv.org/abs/2310.08225.

[35] Kiatdd, CC BY-SA 3.0 <https://creativecommons.org/licenses/by-sa/3.0>, via Wikimedia Commons.

[36] *Pearson correlation explaination*. URL: https://online.stat.psu.edu/stat462/node/96/ (visited on 01/06/2025).

[37] Alec Radford et al. *Robust Speech Recognition via Large-Scale Weak Supervision*. 2022. arXiv: 2212.04356 [eess.AS].

[38] Ricardo Rei et al. "COMET-22: Unbabel-IST 2022 Submission for the Metrics Shared Task". In: *Proceedings of the Seventh Conference on Machine Translation (WMT)*. Ed. by Philipp Koehn et al. Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, Dec. 2022, pp. 578–585. URL: https://aclanthology.org/2022.wmt-1.52/.

[39] Ricardo Rei et al. "COMET: A Neural Framework for MT Evaluation". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Bonnie Webber et al. Online: Association for Computational Linguistics, Nov. 2020, pp. 2685–2702. DOI: 10.18653/v1/2020.emnlp-main.213. URL: https://aclanthology.org/2020.emnlp-main.213.

[40] Rico Sennrich, Barry Haddow, and Alexandra Birch. "Neural Machine Translation of Rare Words with Subword Units". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Katrin Erk and Noah A. Smith. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. DOI: 10.18653/v1/P16-1162. URL: https://aclanthology.org/P16-1162.

[41] C. E. Shannon. "A mathematical theory of communication". In: *The Bell System Technical Journal* 27.3 (1948), pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[42] Matthew Snover et al. "A Study of Translation Edit Rate with Targeted Human Annotation". In: *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*. Cambridge, Massachusetts, USA: Association for Machine Translation in the Americas, Aug. 2006, pp. 223–231. URL: https://aclanthology.org/2006.amta-papers.25/.

[43] Matthias Sperber et al. *Evaluating the IWSLT2023 Speech Translation Tasks: Human Annotations, Automatic Metrics, and Segmentation*. 2024. arXiv: 2406.03881 [cs.CL]. URL: https://arxiv.org/abs/2406.03881.

[44] Nitish Srivastava et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[45] Shuo Sun et al. "An Exploratory Study on Multilingual Quality Estimation". In: *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. Ed. by Kam-Fai Wong, Kevin Knight, and Hua Wu. Suzhou, China: Association for Computational Linguistics, Dec. 2020, pp. 366–377. DOI: 10.18653/v1/2020.aacl-main.39. URL: https://aclanthology.org/2020.aacl-main.39/.

[46] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. *Sequence to Sequence Learning with Neural Networks*. 2014. arXiv: 1409.3215 [cs.CL]. URL: https://arxiv.org/abs/1409.3215.

[47] NLLB Team et al. *No Language Left Behind: Scaling Human-Centered Machine Translation*. 2022. arXiv: 2207.04672 [cs.CL]. URL: https://arxiv.org/abs/2207.04672.

[48] Yi-Lin Tuan et al. "Quality Estimation without Human-labeled Data". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Ed. by Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty. Online: Association for Computational Linguistics, Apr. 2021, pp. 619–625. DOI:

10.18653/v1/2021.eacl-main.50. URL: https://aclanthology.org/2021.eacl-main.50/.

[49] *understanding of the mel spectrogram*. URL: https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53 (visited on 01/02/2025).

[50] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[51] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[52] Apoorv Vyas et al. "Analyzing Uncertainties in Speech Recognition Using Dropout". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019, pp. 6730–6734. DOI: 10.1109/ICASSP.2019.8683086.

[53] Jiayi Wang et al. "Alibaba Submission for WMT18 Quality Estimation Task". In: *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*. Ed. by Ondřej Bojar et al. Belgium, Brussels: Association for Computational Linguistics, Oct. 2018, pp. 809–815. DOI: 10.18653/v1/W18-6465. URL: https://aclanthology.org/W18-6465.

[54] J.P. Woodard and J.T. Nelson. "An information theoretic measure of speech recognition performance". In: 1982.

A. Appendix

A.1. Implementation information

The following is more in depth information about how exactly the probabilities are retrieved with the help of huggingface models and pytorch.

A.1.1. Transcription probability

To retrieve the final transcription probability the huggingface model of Whisper returns a tuple (or dictionary) that contains the probability of each vocabulary entry at each decoding step. Those values are then processed so that the sum of the highest probabilities is taken and then normalised by the number of decoding steps that were done. This gives the transcription probability mean. For reference the pure probability sum was also collected and returned during the experiments. There is a small difference in the resulting QE scores between the two versions but that is more noticeable in the dropout version of the transcription probability.

A.1.2. Translation probability

To get the probability of the top token at each decoding step the functionality from huggingface is used, which is able to return the log-probabilities and scores, so the processed log-probability, of all vocabulary entries. Then to get the final probability of the sequence the top token of each decoding step and its probability is picked and then summed up and divided by the number of decoding steps that are in the sequence.

A.1.3. Softmax entropy

To get the vocabulary values at each decoding step from the seamless model, basic code has been written that iterates over all vocabulary tokens, which are returned by the model generation by the huggingface model, and calculates the entropy of each decoding step.

To get them from DeltaLM the fairseq source code was adapted to do the same as the huggingface model, namely return the probability of all tokens, as the fairseq toolkit is unable to do so natively so far. But due to how the fairseq toolkit returns information, the sum of the softmax entropy over the vocabulary at each decoding step was actually returned.

In the Formicheva [15] paper the softmax entropy measure was proposed as

$$-\frac{1}{T}\sum_{t=1}^{T}\sum_{v=1}^{V} p(y_t^v) log p(y_t^v)$$

with V being the Vocabulary size and T being the length of the translation. In this definition this doesn't work with the data that the models produce as there are quite a lot of 0 values for the vocabulary after using the softmax, which means that the result of the sum like that would not be defined, as the log of 0 is undefined but approaches $-\infty$. To mitigate this the 0 values are masked for the log and thus these values are ignored in the sum.

A.1.4. Standard deviation

The standard deviation is calculated over the top token probability at each decoding step. Those probabilities are the same ones that are retrieved in the translation probability part. On those results the numpy implementation of the standard deviation [31] was then used to retrieve the score.

On DeltaLM the standard deviation was calculated with the help of the numpy implementation on the probabilities that have been output by fairseq. The numpy implementation of the standard deviation takes an array like data structure and if it is not given, calculates the mean of the values in the array and then with that the deviation of the values from the mean. The standard deviation used in numpy is defined as

$$\sqrt{\frac{\sum_i |a_i - \overline{a}|}{N}}$$

where $\overline{a}$ is the mean and a_i is the i-th element in the array.

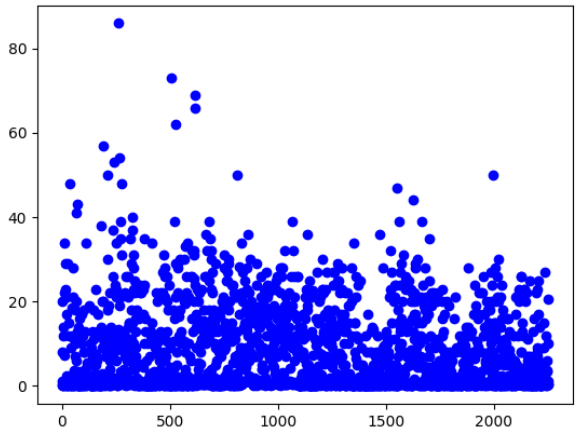
A.2. Other interesting pearson correlation scores

The pearson correlation of the entropies in dropout shows a significant prediction value with a correlation of the mean of -0.23241 and 0.29450, and a correlation of -0.17032 and - 0.12806 of the variance of the entropies on seamless and DeltaLM respectively. This is higher than just the regular translation probability and variance but it comes with a significant cost to compute as calculating the softmax entropy is very rarely implemented in frameworks.

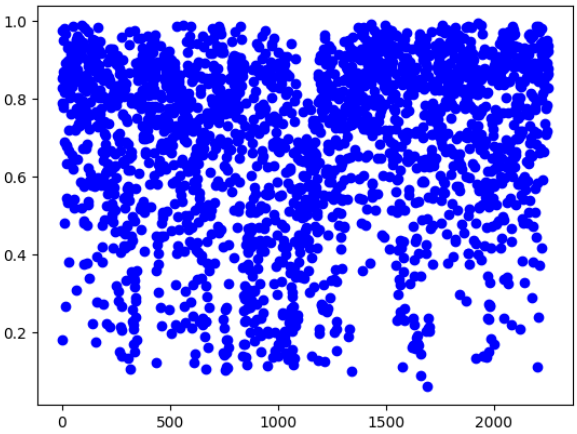
A.3. References graphs

Since there are a lot of plots that contain plots over the reference scores it might be interesting to see how those references fall over the data set.

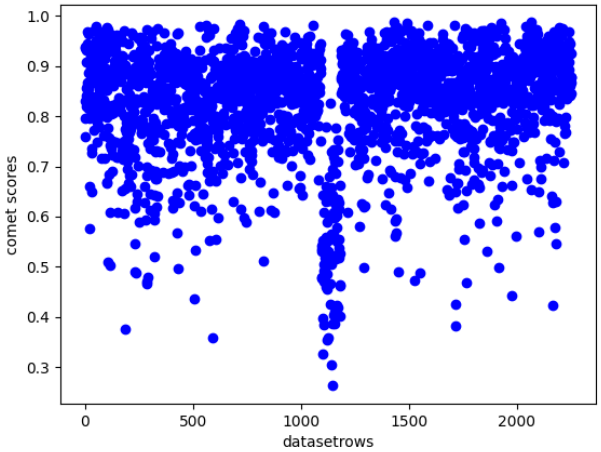
Figure A.1 contains the WER scores from Whisper, the comet scores from seamless as well as the comet scores for DeltaLM, all of these are shown over the lines in the dataset.



(a) all of the reference scores of the dataset; the x axis is the lines in the dataset, the y axis is the WER scores



(b) reference comet scores from seamless



(c) reference comet scores from DeltaLM

Figure A.1.: Reference scores over dataset lines