Natural Language Processing Project 2.1: English-German Machine Translation

English-German Machine Translation

Dimitrij Schulz

TU Berlin, dmtschulz@gmail.com

Machine Translation (MT) leverages software to automatically translate text between languages, significantly enhanced by deep neural networks. This project aims to develop a machine translation model specifically for translating between English and German. Utilizing the "European Parliament Proceedings Parallel Corpus 1996-2011," which comprises extensive multilingual data from European Parliament sessions, we will focus on English-German and German-English parallel texts to train and evaluate our translation models.

CCS CONCEPTS • Computing methodologies~Artificial intelligence~Natural language processing~Machine translation

1 DATA EXPLORATION

This task focuses on extracting meaningful insights through various statistics and visualizations. Key areas of investigation include analyzing length differences between English and German sentences, counting the total number of sentences in the corpus, and uncovering additional patterns or trends within the data.

After loading the sentences from both languages, I proceeded to extract insights from the data. The insights are presented in Table 1.

Table 1: Statistics of German and English Sentences

Statistic	Value
Number of sentences	1920209
Total words (English)	47882343
Total words (German)	44614285
Unique words (English)	**295397**
Unique words (German)	**639030**
Average word length (English)	4.99088206690303
Average word length (German)	6.23566014786519
Average sentence length (English)	149.440577041353
Average sentence length (German)	168.162314102267
Average sentence length difference (English - German)	-18.7217370609136
Most frequent word (English)	the
Most frequent word (German)	die

The table presents a comparison between English and German text data based on several metrics. Significant differences are highlighted in bold.

The data in the Table 1 reveals that although the number of sentences is the same, English has more total words, but fewer unique words compared to German. German words and sentences tend to be longer on average. Some of the other most frequently used words are depicted in Figure 1 as a word cloud.

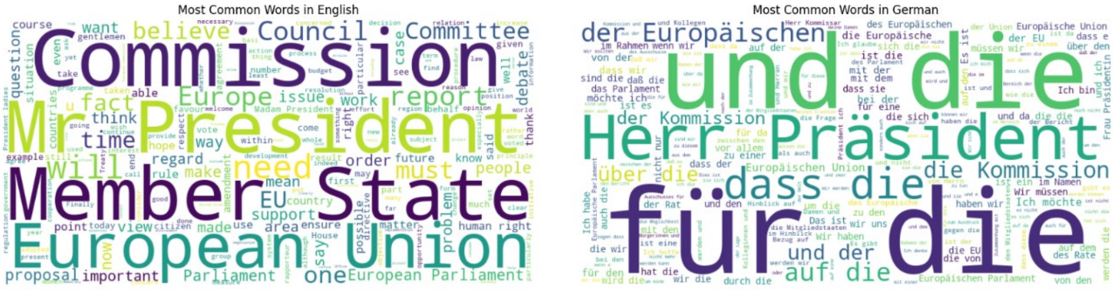


Figure 1: Word Cloud of the Most Common English and German Words.

In Figure 2, we can see the distribution of average word lengths between both languages.

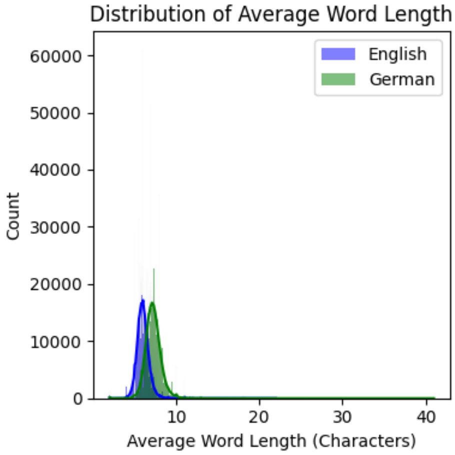


Figure 2: Average Word Lengths in German and English - German Words Tend to be Longer.

Due to a lack of computational resources, I will use 5% of the entire dataset for subsequent tasks.

2 PRE-PROCESSING

For this task, I applied several pre-processing steps to prepare the data for model training.

1. **Lowercasing**: Converting all text to lowercase ensures uniformity and helps in reducing the complexity of the vocabulary.
2. **Number List Removal**: Removing number lists (e.g., "123, 234, 345") cleans the text from unnecessary elements that do not contribute to text translation quality.
3. **Punctuation Removal**: Removing punctuation simplifies the text.
4. **Specific Character Removal**: Replace specific characters (e.g., "½", "¾", "£", "°", "§") with their word equivalents to maintain semantic meaning while standardizing the text.
5. **Whitespace Normalization**: Removing empty lines and extra whitespaces, especially around numbers.

6. **Report Number Removal**: Stripping report numbers (e.g., "H-123") helps in removing non-informative tokens that could confuse the model.
7. **Number Conversion**: Converting numerical values to words (e.g., "1" to "ein") helps the model learn better translations for numbers, as it treats them as words rather than isolated tokens.

Some steps, like stemming or lemmatization, were omitted because they could strip important linguistic nuances necessary for accurate machine translation. Similarly, complex text normalization processes that might reduce the readability or semantic integrity of the text were avoided.

The processed data was saved into separate files for English and German to facilitate further stages of model development. Data exploration and pre-processing are done one single time in the file word_model_en_de.ipynb. All the subsequent tasks are done with the 2 loaded files which were the results of task 1 and 2.

3 NEURAL MACHINE TRANSLATION

I load the pre-processed dataset and divide it into three parts: training, validation, and test sets. Specifically, 20% of the dataset (which constitutes 5% of the entire original dataset) was allocated as the test set. This resulted in 7006 train sentences, 610 validation and 1904 test sentences. During further preparation for the training, I came up with a vocabulary size of 4437 word for English and 5335 for German vocabulary. I used words which appeared 3 times or more.

For this task, I have chosen a simple RNN-based sequence-to-sequence (Seq2Seq) architecture for the machine translation model. This architecture comprises an encoder and a decoder, both implemented using RNNs and both having an embedding layer which will be initialized with pre-trained embeddings before training procedure.

Encoder: The encoder processes the input sequence and compresses it into a context vector (the hidden state). This context vector encapsulates information about the entire input sequence.

Decoder: The decoder takes the context vector from the encoder and generates the target sequence. At each step, it produces an output (the next token in the sequence) and updates its hidden state.

Seq2Seq Model: The seq2seq model combines the encoder and decoder. During training, it uses teacher forcing, which means it feeds the actual target token to the decoder at each time step with a certain probability (the teacher forcing ratio). This helps the model learn more effectively by providing the correct context during training.

Overall, this RNN-based Seq2Seq architecture was chosen because it is simple yet meets the requirements. Another reason is that it can be easily adapted for the attention mechanism, making it a good starting point to build more complex models on top of the chosen architecture.

I trained few models using nn.RNN class inside the encoder and decoder but the results were too poor so I decided to use nn.LSTM instead of nn.RNN which gave better results.

The neural network models use the following hyperparameters:

1. **Embedding Dimension**: Both the encoder and decoder use an embedding dimension of 300 to align with the dimensions of pre-trained GloVe and Word2Vec embeddings.
2. **Hidden Dimension**: The hidden dimension is set to 256 to reduce training time.
3. **Number of Layers**: 2.
4. **Dropout Rate**: A dropout rate of 0.5 is applied to prevent overfitting by randomly deactivating half of the neurons during training.
5. **Number of Epochs**: All models are trained for 10 epochs.

6. **Gradient Clipping**: Gradient clipping is set to 1.0 to maintain training stability by preventing excessively large gradients.
7. **Teacher forcing Ratio:** 0.5, it specifies the probability with which teacher forcing is applied at each step of the decoding process.

3.1 English to German Model Evaluation with BLEU and ROUGE Metrics

For the English-to-German translation, the evaluation results using GloVe or Word2Vec embeddings were as follows:

3.1.1GloVe

After evaluating the test data, the BLEU score for the predictions was:

1. **BLEU Score**: 0.002
2. **Precisions**: [0.063, 0.007, 0.001, 0.0002]
3. **Brevity Penalty**: 0.723
4. **Length Ratio**: 0.755
5. **Translation Length**: 33,175
6. **Reference Length**: 43,953

The BLEU score of 0.002 indicates very little overlap between predicted and reference translations. The brevity penalty of 0.723 shows the translation was shorter than the reference, contributing to the low BLEU score.

For ROUGE scores:

1. **ROUGE-1**: 0.059
2. **ROUGE-2**: 0.011
3. **ROUGE-L**: 0.057

ROUGE-1 suggests some unigram overlap, while ROUGE-2 indicates limited bigram overlap. ROUGE-L shows modest overlap in longest common subsequences.

3.1.2Word2Vec

After evaluating the test data, the BLEU score for the predictions was:

1. **BLEU Score**: 0.003
2. **Precisions**: [0.090, 0.010, 0.001, 0.0003]
3. **Brevity Penalty**: 0.715
4. **Length Ratio**: 0.748
5. **Translation Length**: 32,897
6. **Reference Length**: 43,953

The BLEU score of 0.003 indicates a minimal overlap between the predicted and reference translations. The brevity penalty of 0.715 reflects that the translation was shorter than the reference, contributing to the low BLEU score.

For ROUGE scores:

1. **ROUGE-1**: 0.078
2. **ROUGE-2**: 0.012

3. **ROUGE-L**: 0.073

ROUGE-1 shows some unigram overlap, while ROUGE-2 indicates limited bigram overlap. ROUGE-L suggests a modest amount of overlap in longest common subsequences.

3.2 German to English Model Evaluation with BLEU and ROUGE Metrics

For the German-to-English translation, the evaluation results using GloVe or Word2Vec embeddings were as follows:

3.2.1GloVe

After evaluating the test data, the BLEU score for the predictions was:

1. **BLEU Score**: 0.006
2. **Precisions**: [0.189, 0.021, 0.002, 0.0008]
3. **Brevity Penalty**: 0.681
4. **Length Ratio**: 0.723
5. **Translation Length**: 34,273
6. **Reference Length**: 47,418

The BLEU score of 0.006 indicates very limited overlap between the predicted and reference translations. The brevity penalty of 0.681 shows that the translation was shorter than the reference, affecting the BLEU score.

For ROUGE scores:

1. **ROUGE-1**: 0.152
2. **ROUGE-2**: 0.017
3. **ROUGE-L**: 0.144

ROUGE-1 indicates some unigram overlap, while ROUGE-2 reflects limited bigram overlap. ROUGE-L shows a modest amount of overlap in longest common subsequences.

3.2.2Word2Vec

After evaluating the test data, the BLEU score for the predictions was:

1. **BLEU Score**: 0.013
2. **Precisions**: [0.218, 0.034, 0.006, 0.003]
3. **Brevity Penalty**: 0.700
4. **Length Ratio**: 0.737
5. **Translation Length**: 34,935
6. **Reference Length**: 47,418

The BLEU score of 0.013 indicates minimal overlap between the predicted and reference translations. The brevity penalty of 0.700 shows the translation was shorter than the reference, impacting the BLEU score.

For ROUGE scores:

1. **ROUGE-1**: 0.176
2. **ROUGE-2**: 0.027
3. **ROUGE-L**: 0.160

ROUGE-1 indicates a moderate amount of unigram overlap, while ROUGE-2 shows limited bigram overlap. ROUGE-L reflects a reasonable amount of overlap in longest common subsequences.

3.3 Character based English to German Model Evaluation with BLEU and ROUGE Metrics

Due to performance considerations, I will only focus on evaluating the character-level translation model for English-to-German translation using GloVe embeddings. The BLEU score for this model is 0.0000, indicating no overlap between the predicted translations and the reference translations. The ROUGE scores are as follows:

1. **ROUGE-1 Score**: 0.0016
2. **ROUGE-2 Score**: 0.0
3. **ROUGE-L Score**: 0.0016

These scores indicate minimal overlap at the unigram level and no overlap at the bigram level, reflecting a very low similarity between the generated translations and the reference texts.

During training attempts I noticed that shorter sentences have better translations.

4 NEURAL MACHINE TRANSLATION WITH ATTENTION

4.1 English to German Model Evaluation with BLEU and ROUGE Metrics with Attention

For the word-level English-to-German translation model using GloVe and Word2Vec embeddings with the attention mechanism, the evaluation results were:

4.1.1GloVe

After evaluating the test data, the BLEU score for the predictions was:

1. **BLEU Score**: 0.002
2. **Precisions**: [0.098, 0.0065, 0.0007, 0.0001]
3. **Brevity Penalty**: 0.748
4. **Length Ratio**: 0.775
5. **Translation Length**: 34,071
6. **Reference Length**: 43,953

The BLEU score of 0.002 indicates very limited overlap between the predicted and reference translations. The brevity penalty of 0.748 reflects that the translation was significantly shorter than the reference, contributing to the low BLEU score.

For ROUGE scores:

1. **ROUGE-1**: 0.088
2. **ROUGE-2**: 0.009
3. **ROUGE-L**: 0.083

ROUGE-1 suggests some unigram overlap, while ROUGE-2 shows minimal bigram overlap. ROUGE-L indicates a modest amount of overlap in longest common subsequences.

4.1.2Word2Vec

After evaluating the test data, the BLEU score for the predictions was:

1. **BLEU Score**: 0.007
2. **Precisions**: [0.142, 0.019, 0.004, 0.0011]
3. **Brevity Penalty**: 0.723
4. **Length Ratio**: 0.755
5. **Translation Length**: 33,193
6. **Reference Length**: 43,953

The BLEU score of 0.007 indicates limited overlap between the predicted and reference translations. The brevity penalty of 0.723 shows that the translation was shorter than the reference, which contributed to the low BLEU score.

For ROUGE scores:

1. **ROUGE-1**: 0.119
2. **ROUGE-2**: 0.021
3. **ROUGE-L**: 0.109

ROUGE-1 shows a moderate level of unigram overlap, while ROUGE-2 indicates limited bigram overlap. ROUGE-L reflects a reasonable amount of overlap in longest common subsequences.

4.2 German to English Model Evaluation with BLEU and ROUGE Metrics with Attention

For the word-level German-to-English translation model using GloVe and Word2Vec embeddings with the attention mechanism, the evaluation results were:

4.2.1GloVe

After evaluating the test data, the BLEU score for the predictions was:

1. **BLEU Score**: 0.005
2. **Precisions**: [0.189, 0.021, 0.0013, 0.0006]
3. **Brevity Penalty**: 0.698
4. **Length Ratio**: 0.735
5. **Translation Length**: 34,869
6. **Reference Length**: 47,418

The BLEU score of 0.005 indicates very limited overlap between the predicted and reference translations. The brevity penalty of 0.698 suggests that the translation was shorter than the reference, impacting the BLEU score.

For ROUGE scores:

1. **ROUGE-1**: 0.153
2. **ROUGE-2**: 0.016
3. **ROUGE-L**: 0.143

ROUGE-1 indicates moderate unigram overlap, while ROUGE-2 shows limited bigram overlap. ROUGE-L reflects a reasonable amount of overlap in longest common subsequences.

4.2.2Word2Vec

After evaluating the test data, the BLEU score for the predictions was:

1. **BLEU Score**: 0.012
2. **Precisions**: [0.221, 0.035, 0.0055, 0.0022]
3. **Brevity Penalty**: 0.674
4. **Length Ratio**: 0.717
5. **Translation Length**: 33,985
6. **Reference Length**: 47,418

The BLEU score of 0.012 indicates limited overlap between the predicted and reference translations. The brevity penalty of 0.674 shows that the translation was shorter than the reference, affecting the BLEU score.

For ROUGE scores:

1. **ROUGE-1**: 0.179
2. **ROUGE-2**: 0.028
3. **ROUGE-L**: 0.163

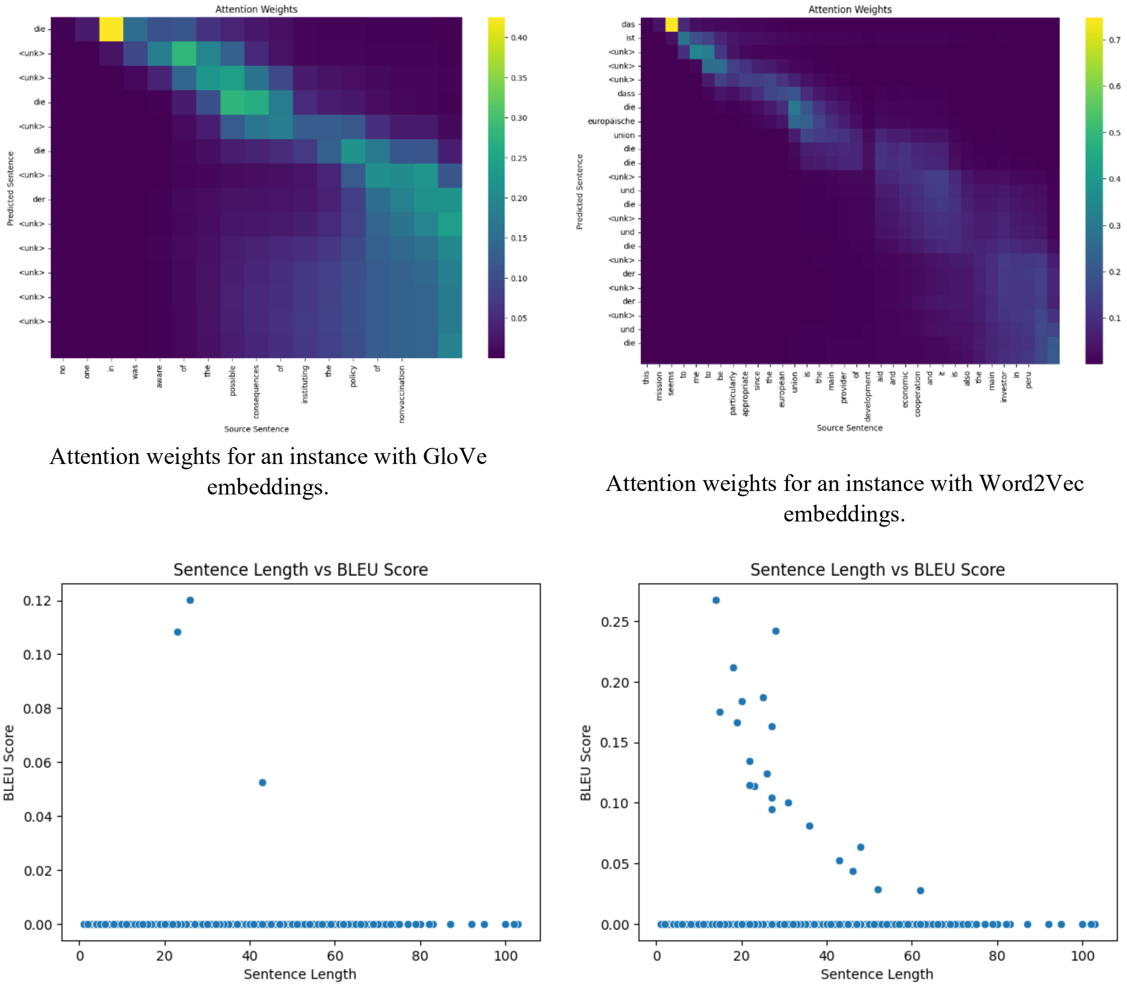
ROUGE-1 indicates a moderate level of unigram overlap, while ROUGE-2 reflects limited bigram overlap. ROUGE-L shows a reasonable amount of overlap in longest common subsequences

Analyzing the properties of the top predicted sentences for a model with attention and Word2Vec embeddings for German-to-English translations reveals that shorter sentences are translated better. Another observed property is that these sentences contain words that appear very frequently, some of them can be seen in the word cloud. For example: Top 1 Sentence:

Sentence: mr president ladies and gentlemen the <unk> of the <unk> of the <unk>

BLEU Score: 0.36624839813098803

Figure 3 depicts attention weights for an instance with GloVe and Word2Vec and BLEU scores for the English to German translation model. We can see that shorter sentences tend to have a higher precision. Figure 4 depicts attention weights for an instance with GloVe and Word2Vec for the German to English translation model. We can see that shorter sentences tend to have a higher precision.



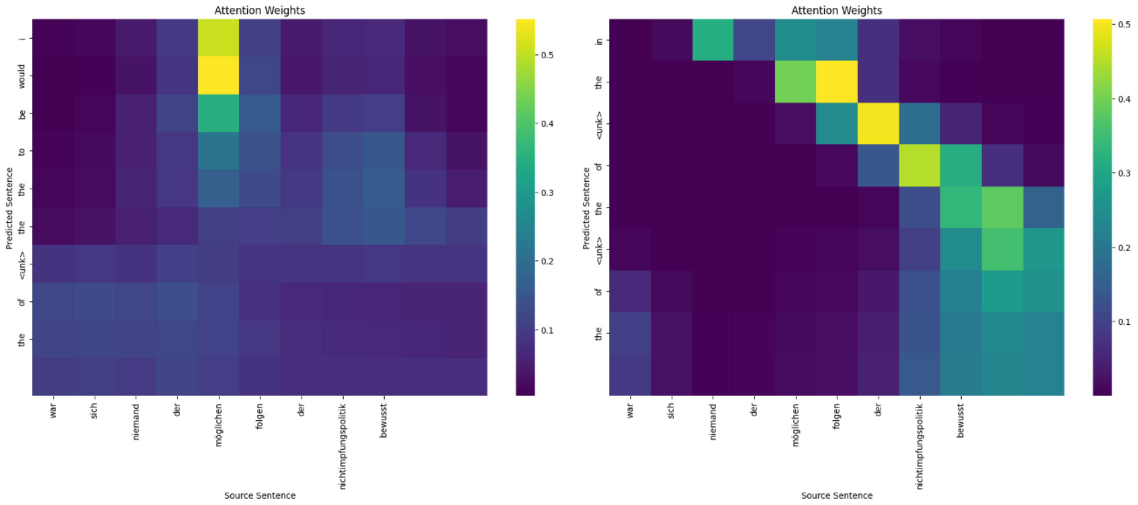
Attention weights for an instance with GloVe embeddings.

Attention weights for an instance with Word2Vec embeddings.

BLEU scores for test data using GloVe embeddings.

BLEU scores for test data using Word2Vec embeddings.

Figure 3: Attention weights for an instance with GloVe and Word2Vec embeddings and BLEU scores for the whole test data. English to German translation.



Attention weights for an instance with GloVe embeddings.

Attention weights for an instance with Word2Vec embeddings.

Figure 4: Attention weights for an instance with GloVe and Word2Vec embeddings. German to English sample translation.

When comparing the results with and without the attention mechanism, I did not observe a significant improvement for either type of embedding. Although there was an expected improvement, training with nn.RNN showed a significant improvement (despite the overall performance not being outstanding), and the improvement with the attention mechanism was still evident in that version. I decided to use nn.LSTM in the encoder and decoder because the models showed promising performance without the attention mechanism compared to nn.RNN. When comparing GloVe and Word2Vec embeddings, the latter performed almost every time 2 times better.