USING AI TO MAKE 絵 COMMUNICATION ACCESSIBLE

ABSTRACT

This project tested which Attention mechanism that can be implemented into a **Neural Machine Translation (NMT)** model is better suited for the translation of **low-resource datasets** using a pretrained model with a small dataset. Low-resource languages often lack sufficient training data to be able to accurately translate. This poses a problem as people native to the language have fewer accessible resources and the languages have a higher chance of going extinct.

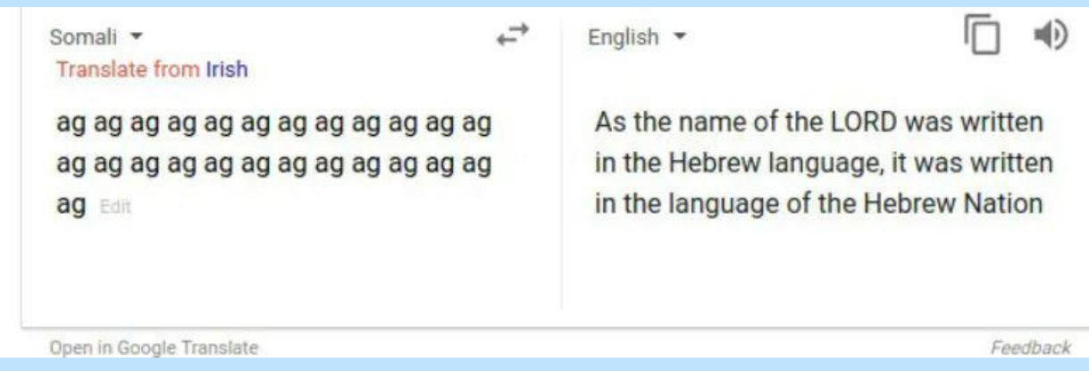
The research utilized **pre-trained models** from an **open source repository**, Hugging Face Hub, that used small English to German datasets to mimic low-resource languages. Three different mechanisms were implemented to three NMT models and fine-tuned before being tested using installed programs from OpenNMT, an open source repository. Scaled-dot product Attention, Multi-head Attention, and Cross Attention were implemented onto a **Transformer architecture**. The hypothesis was that the NMT model will be the most accurate with the implementation of the Transformer with the Multi-head attention mechanism because it will additionally incorporate the Scaled Dot mechanism but not be exaggerated in the number of mechanisms working at once. This hypothesis was based on the rationale that in regards to small datasets, having too many mechanisms working in the model would lead to a contextually inaccurate translation.

The hypothesis did not align with the results of the experiment. According to the Bilingual Evaluation Understudy (BLEU) metric and Perplexity (PLP) metric used, the tests that provided the highest BLEU score used the Cross Attention mechanism. This is relevant to improving translation for low-resource languages as it improves the model's response to input without further training. The model and experiment were made cost-accessible, however, it would be beneficial to consider in the future how the Cross Attention mechanism would perform for true low-resource language pairs.

PROBLEM

- Many indigenous and uncommon languages are considered **low-resource** in their datasets and are difficult to translate
  - E.g. Vietnamese, Swahili, Hindi, Thai, Urdu, Bengali, Hawai'ian
- Artificial Intelligence models need lots of training on lots of data in order to produce accurate translations
- **Neural Machine Translation Models (NMT)** are able to translate idiomatic expressions taking context into account
  - Does this by mimicking the cognitive functions of the human brain, weighing options before coming to a decision
- These models still have issues with low-resource languages
  - Models can unintentionally contain **bias**
- Can be improved with the implementation of **Attention algorithms**
  - This is a mechanism that mimics how humans focus on certain parts of sentences

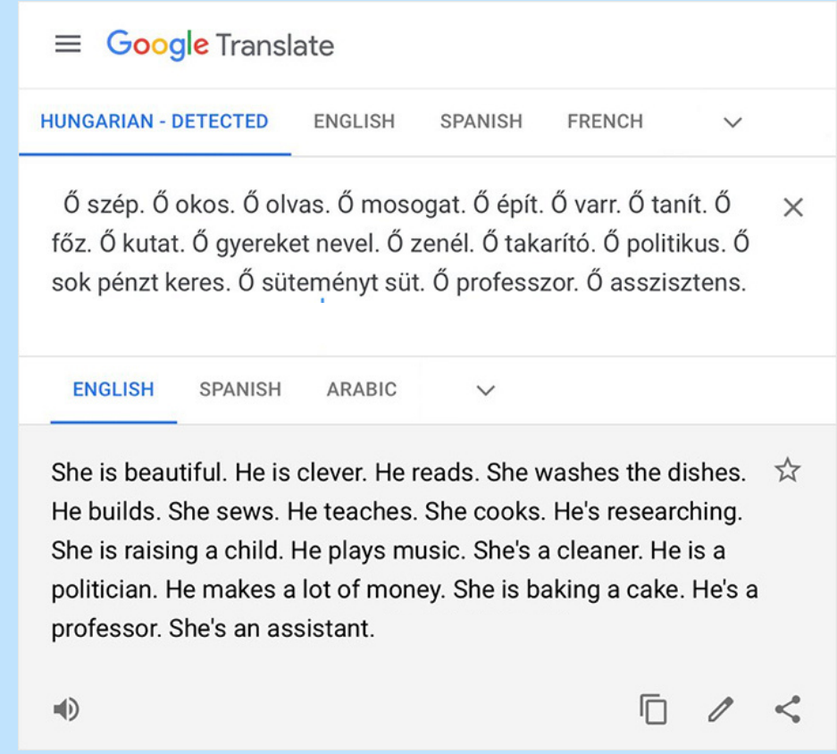
EXAMPLE: "IT'S RAINING CATS AND DOGS" CANNOT BE DIRECTLY INTO SPANISH AS "ESTA LLOVIENDO GATOS Y PERROS"



ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag

As the name of the LORD was written in the Hebrew language, it was written in the language of the Hebrew Nation

GOOGLE TRANSLATE'S NEURAL MACHINE TRANSLATION MODEL INCORRECTLY INTERPRETING A LOW-RESOURCE LANGUAGE IN 2017



Ő szép. Ő okos. Ő olvas. Ő mosogat. Ő épít. Ő varr. Ő tanít. Ő főz. Ő kutat. Ő gyereket nevel. Ő zenél. Ő takarító. Ő politikus. Ő sok pénzt keres. Ő süteményt süt. Ő professzor. Ő asszisztens.

She is beautiful. He is clever. He reads. She washes the dishes. He builds. She sews. He teaches. She cooks. He's researching. She is raising a child. He plays music. She's a cleaner. He is a politician. He makes a lot of money. She is baking a cake. He's a professor. She's an assistant.

GOOGLE TRANSLATE'S NEURAL MACHINE TRANSLATION MODEL ASSIGNING A GENDER FOR THE TRANSLATION FROM HUNGARIAN, A GENDER NEUTRAL LANGUAGE THAT DOES NOT USE GENDERED PRONOUNS

GOALS

- Improve the performance of the Neural Machine Translation (NMT) without further training
- Find an attention mechanism that improves a model specifically for low-resource languages
- Help make NMT accessible to different sociocultural identities, particularly those with low-resource datasets

HYPOTHESIS

The performance of the Neural Machine Translation model will be the most accurate with the implementation of the Transformer with the Multi-head attention mechanism because it will additionally incorporate the Scaled-dot product attention mechanism, but not be exaggerated in the number of mechanisms working at once.

This is based on the rationale above as using too many mechanisms at once may lead to overfitting while using only one mechanism may lead to issues in context. Therefore, Multi-head attention will be the most accurate.

METHODOLODY

TYPES OF ATTENTION MECHANISMS

- Attention mechanisms are algorithms that can be integrated into a Neural Machine Translation model in order to change the way the models interprets input texts
- Meant to mimic how humans focus on certain words of phrases more than others
- Words such as "the" and "to" would **weigh** less, or have less importance, in the model's interpretation
- **Analogy: if the Neural Machine Translation model was a car, the Attention mechanisms are the engine which determine the efficiency and speed of the overall car**

SCALED-DOT PRODUCT

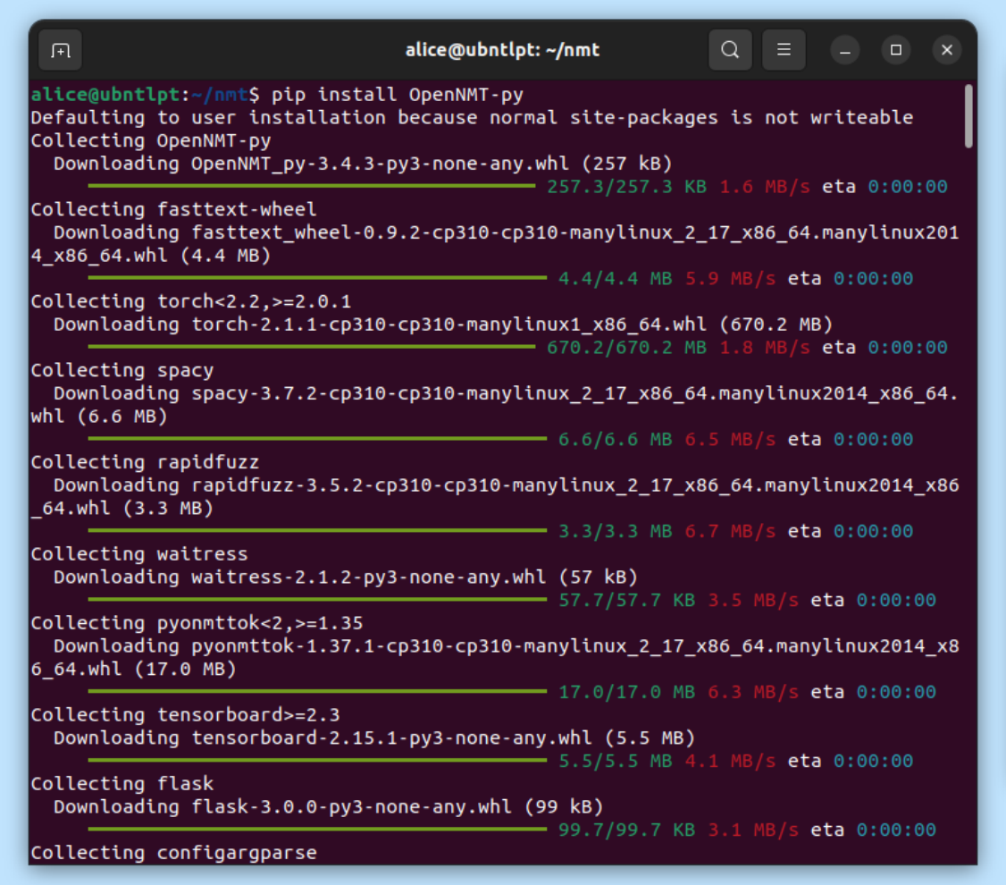
- **Analogy: Similar to working with a group of people, each person representing a word or token, that pay attention to each other to ensure coordination in their work**
- Decides which parts of phrase are more important through attention scores
- Compares two sets of lists to find how similar they are, then weighs the words to place a numerical value of their importance for the model to understand
- Calculates the dot product between query and key vectors

MULTI-HEAD

- **Analogy: In the same scenario of working with a group of people, they all give their personal input about the same task from different perspectives**
- Breaks up input text and sends it to multiple heads of the model, resulting in an output consisting of the concatenated inputs

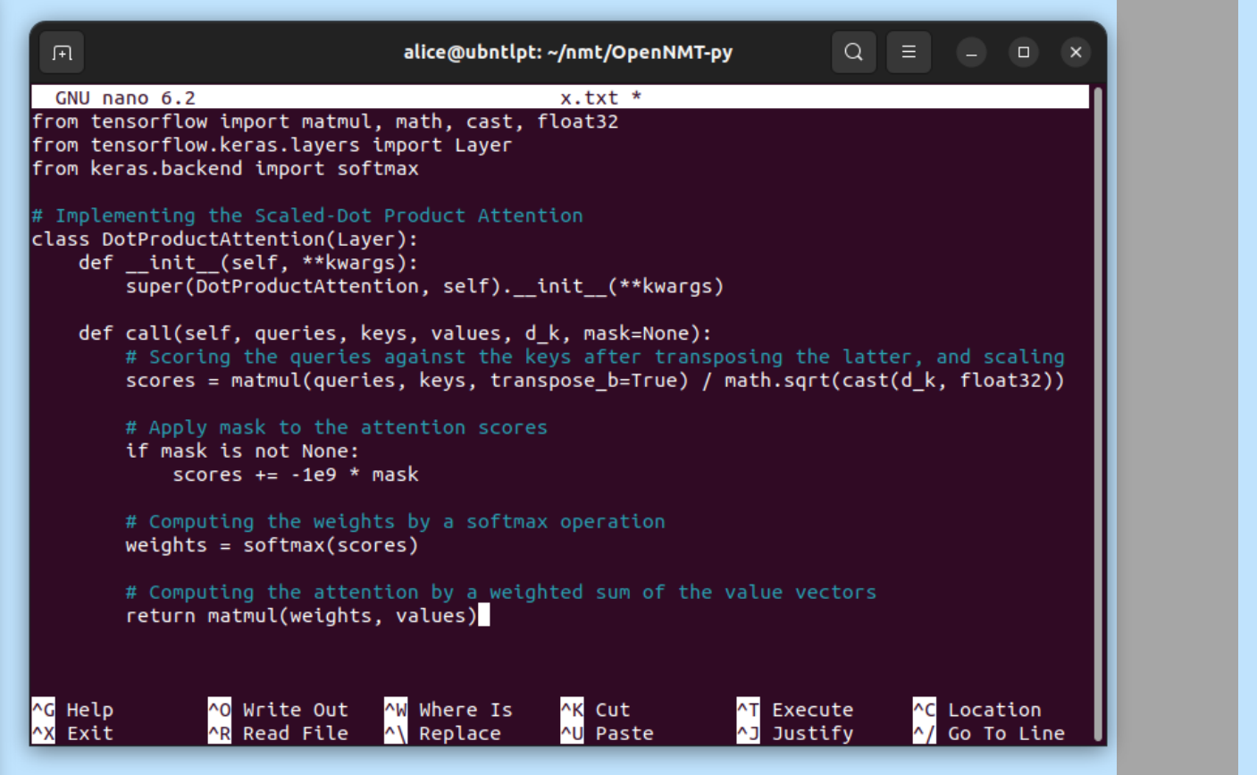
CROSS

- **Analogy: In the same scenario of working with a group of people, it is as if each member specializes in different areas and breaks the task up to focus on the specific details they specialize in**
- Utilizes a separate embedding sequences in order to incorporate relevant input information for the output sequence



```
alice@ubntlpt:~/nmt$ pip install OpenNMT-py
Defaulting to user installation because normal site-packages is not writeable
Collecting OpenNMT-py
  Downloading OpenNMT_py-3.4.3-py3-none-any.whl (257 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 257.3/257.3 KB 1.6 MB/s eta 0:00:00
Collecting fasttext-wheel
  Downloading fasttext_wheel-0.9.2-cp310-cp310-manylinux_2_17_x86_64.manylinux201
4_x86_64.whl (4.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.4/4.4 MB 5.9 MB/s eta 0:00:00
Collecting torch<2.2,>=2.0.1
  Downloading torch-2.1.1-cp310-cp310-manylinux1_x86_64.whl (670.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 670.2/670.2 MB 1.8 MB/s eta 0:00:00
Collecting spacy
  Downloading spacy-3.7.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.
whl (6.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.6/6.6 MB 6.5 MB/s eta 0:00:00
Collecting rapidfuzz
  Downloading rapidfuzz-3.5.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86
_64.whl (3.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.3/3.3 MB 6.7 MB/s eta 0:00:00
Collecting waitress
  Downloading waitress-2.1.2-py3-none-any.whl (57 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 57.7/57.7 KB 3.5 MB/s eta 0:00:00
Collecting pyonmttok<2,>=1.35
  Downloading pyonmttok-1.37.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x8
6_64.whl (17.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 17.0/17.0 MB 6.3 MB/s eta 0:00:00
Collecting tensorboard>=2.3
  Downloading tensorboard-2.15.1-py3-none-any.whl (5.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.5/5.5 MB 4.1 MB/s eta 0:00:00
Collecting flask
  Downloading flask-3.0.0-py3-none-any.whl (99 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 99.7/99.7 KB 3.1 MB/s eta 0:00:00
Collecting configargparse
```

INSTALLATION OF MODEL THROUGH OPEN-NMT

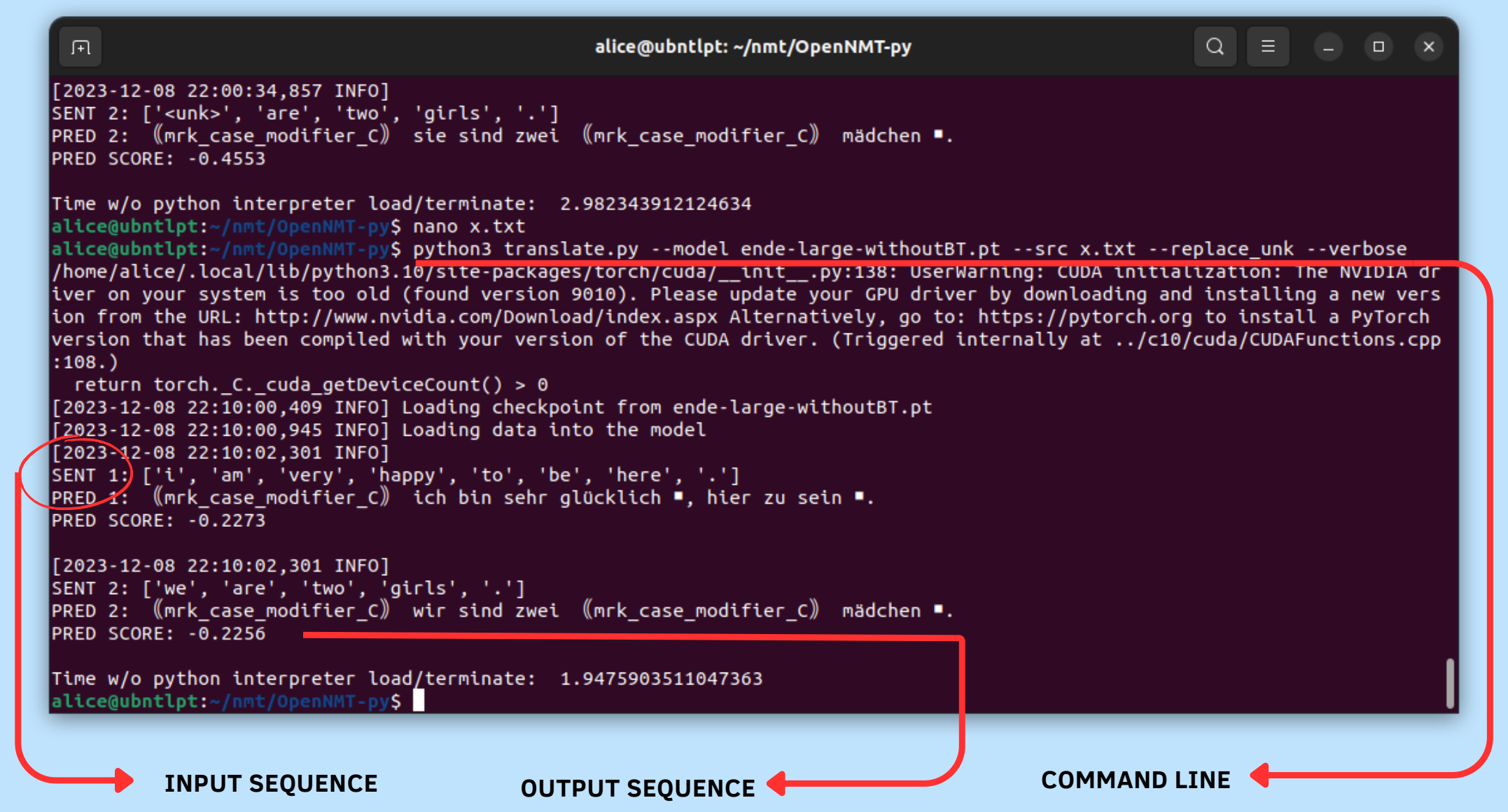


```
from tensorflow import matmul, math, cast, float32
from tensorflow.keras.layers import Layer
from keras.backend import softmax

# Implementing the Scaled-Dot Product Attention
class DotProductAttention(Layer):
    def __init__(self, **kwargs):
        super(DotProductAttention, self).__init__(**kwargs)

    def call(self, queries, keys, values, d_k, mask=None):
        # Scoring the queries against the keys after transposing the latter, and scaling
        scores = matmul(queries, keys, transpose_b=True) / math.sqrt(cast(d_k, float32))

        # Apply mask to the attention scores
        if mask is not None:
            scores += -1e9 * mask

        # Computing the weights by a softmax operation
        weights = softmax(scores)

        # Computing the attention by a weighted sum of the value vectors
        return matmul(weights, values)
```

IMPLEMENTATION OF ATTENTION MECHANISM ALGORITHM TO MODEL

DESIGN

1. Used a Linux based operating system, Ubuntu, so that Python 3.8 and Pytorch could be installed.
2. Downloaded **four different models** from Hugging Face Hub, an open source machine learning library: a non-Transformer model without attention mechanisms, a Transformer model with Scaled-dot product attention, a Transformer model with Multi-head attention, and a Transformer model that uses Cross-head attention. The use of four different models prevented the results from being skewed due to the model learning from the previous tests. Used models trained on the same dataset.
3. Fine-tuned the models using the "train.py" command script that imported components from the "models" directory from **OpenNMT**
4. Obtained three different text inputs from three different sources of literature.
5. Divided into difficulty categories based on the Lexile metric the library that carries the texts used
6. Tokenized the inputs in a shell script
7. **Ran the program** using the command line "python3 translate.py –[model name] –src [script name] –replace_unk –verbose"
8. Ran bilingual evaluation understudy (BLEU) algorithm and Perplexity score (PLP) using "evaluate.py" command script
9. Recorded test input/output as well as PLP score and BLEU score
10. Repeated steps 6-9 using each text input
11. Changed command line to utilize next model "python3 translate.py –[next model name] –src [next script name] –replace_unk –verbose"
12. Repeated steps 7-10 using new model and test inputs
13. Graphed and compared data
14. Looked for upward trends in both PLP and BLEU to indicate improvement



```
[2023-12-08 22:00:34,857 INFO]
SENT 2: ['<unk>', 'are', 'two', 'girls', '.']
PRED 2:  ⟪mrk_case_modifier_C⟫  sie sind zwei  ⟪mrk_case_modifier_C⟫  mädchen ■.
PRED SCORE: -0.4553

Time w/o python interpreter load/terminate:  2.982343912124634
alice@ubntlpt:~/nmt/OpenNMT-py$ nano x.txt
alice@ubntlpt:~/nmt/OpenNMT-py$ python3 translate.py --model ende-large-withoutBT.pt --src x.txt --replace_unk --verbose
/home/alice/.local/lib/python3.10/site-packages/torch/cuda/__init__.py:138: UserWarning: CUDA initialization: The NVIDIA dr
iver on your system is too old (found version 9010). Please update your GPU driver by downloading and installing a new vers
ion from the URL: http://www.nvidia.com/Download/index.aspx Alternatively, go to: https://pytorch.org to install a PyTorch
version that has been compiled with your version of the CUDA driver. (Triggered internally at ../c10/cuda/CUDAFunctions.cpp
:108.)
  return torch._C._cuda_getDeviceCount() > 0
[2023-12-08 22:10:00,409 INFO] Loading checkpoint from ende-large-withoutBT.pt
[2023-12-08 22:10:00,945 INFO] Loading data into the model
[2023-12-08 22:10:02,301 INFO]
SENT 1: ['i', 'am', 'very', 'happy', 'to', 'be', 'here', '.']
PRED 1:  ⟪mrk_case_modifier_C⟫  ich bin sehr glücklich ■, hier zu sein ■.
PRED SCORE: -0.2273

[2023-12-08 22:10:02,301 INFO]
SENT 2: ['we', 'are', 'two', 'girls', '.']
PRED 2:  ⟪mrk_case_modifier_C⟫  wir sind zwei  ⟪mrk_case_modifier_C⟫  mädchen ■.
PRED SCORE: -0.2256

Time w/o python interpreter load/terminate:  1.9475903511047363
alice@ubntlpt:~/nmt/OpenNMT-py$
```

INPUT SEQUENCE

OUTPUT SEQUENCE

COMMAND LINE

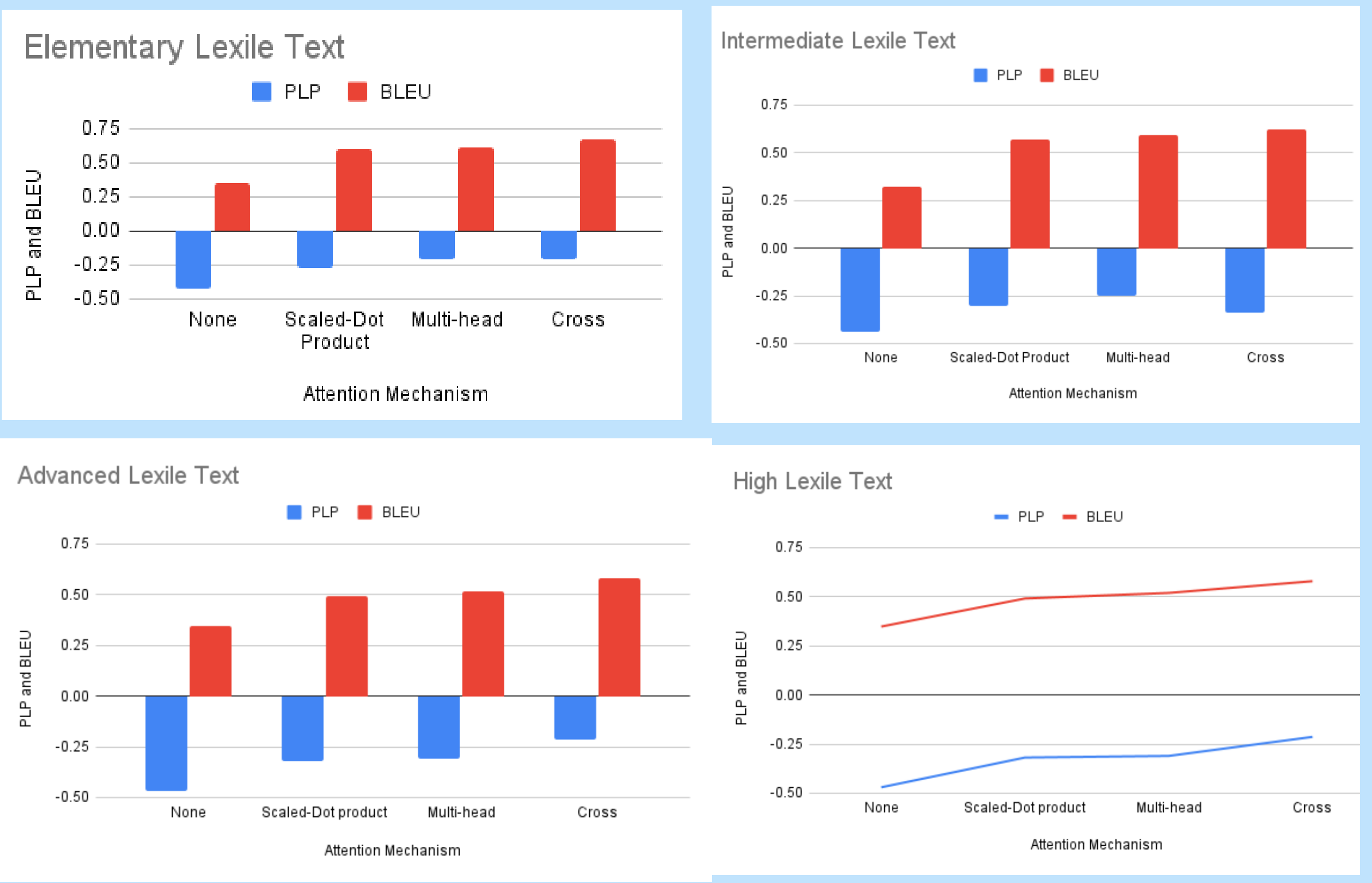
EVALUATION METRICS

- Two algorithms were used to evaluate the quality of the translation
- **BLEU (Bilingual Evaluation Understudy)**
  - Most commonly used evaluation metric in machine translation tasks
  - Compares input text with output text
    - Divides the number of predicted words with the number of output words
  - Score between 0 and 1
    - 1 being a perfect translation
- **PLP (Perplexity)**
  - Measures the uncertainty the model has in the translation
  - How "surprised" the model is with the given input
  - Scores are negative
    - Nearer to 0 indicating the model is better at predicting the sequence of words

RESULTS

- Cross-head attention mechanisms had highest positive trend in both BLEU and PLP scores
- Issues in formatting the **command** with **incorrect paths**, which was resolved by moving the **directory** where the model was
- Without implementing any attention mechanisms, model was **case sensitive** and marked capital "i" as an unknown word
  - Led to higher perplexity score of -0.4553
- When text was properly **tokenized** the score went up to -0.2256
- **Run time** for Scaled-Dot Product mechanism was longer than for Multi-head mechanism

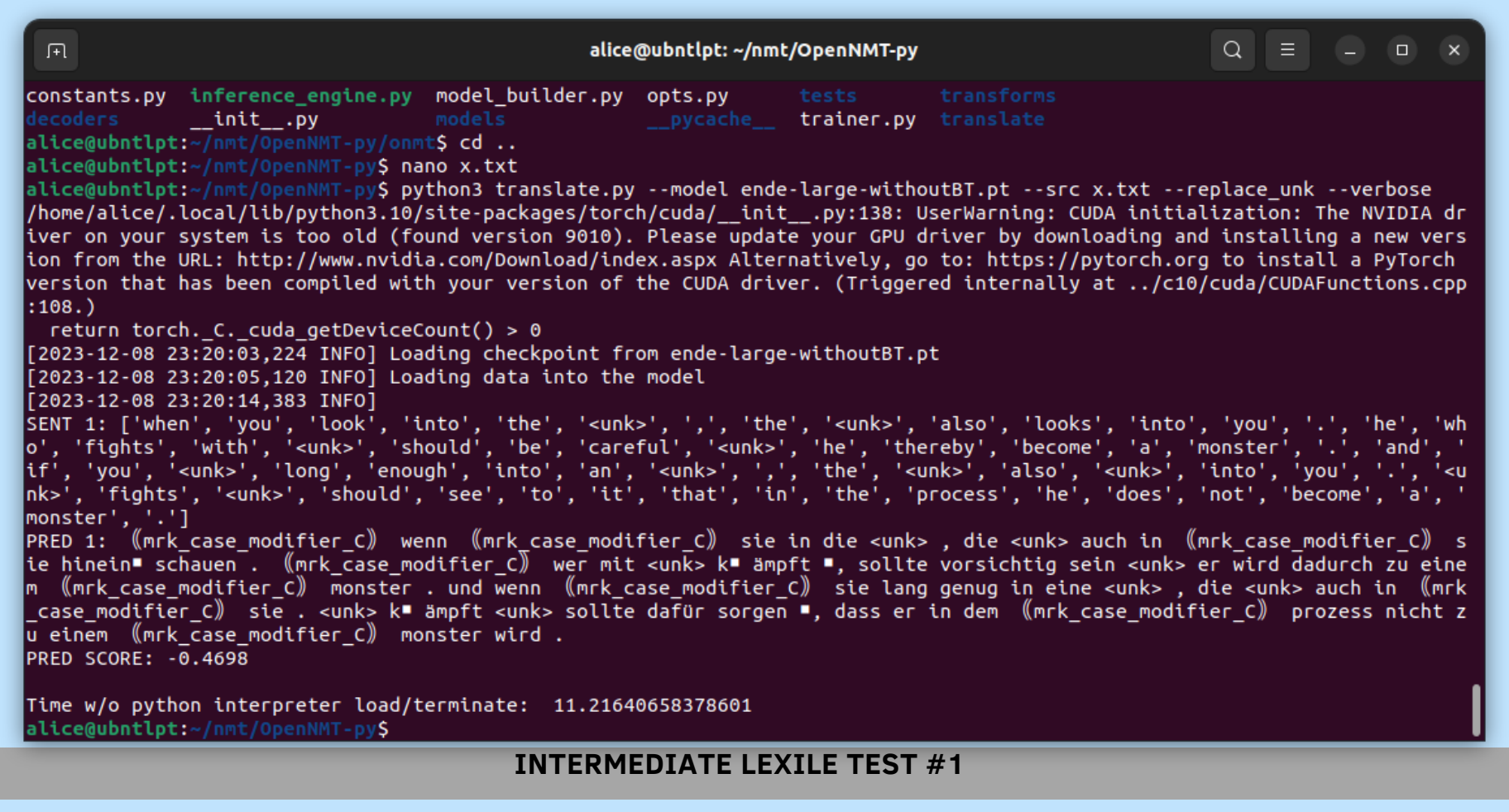
Test #	Advanced Technique	Target Language	Reading Level	Perplexity Score (0)	BLEU Score (1)
1	None	German	Intermediate	-0.4360	0.3225
2	None	German	Advanced	-0.4698	0.3478
3	None	German	Elementary	-0.4309	0.4210
4	Scaled Dot	German	Advanced	-0.3184	0.4905
5	Scaled Dot	German	Elementary	-0.2681	0.6021
6	Scaled Dot	German	Intermediate	-0.3023	0.5667
7	Multi-head	German	Intermediate	–0.2504	0.5938
8	Multi-head	German	Advanced	-0.3102	0.5189
9	Multi-head	German	Elementary	-0.2191	0.6120
10	Cross-head	German	Intermediate	-0.3412	0.6232
11	Cross-head	German	Elementary	-0.2087	0.6678
12	Cross-head	German	Advanced	-0.2134	0.5786



SHOWS A POSITIVE RELATIONSHIP BETWEEN THE USE OF ATTENTION MECHANISMS AND THE ACCURACY OF TRANSLATION ACCORDING TO THE PLP AND BLEU METRICS

ANALYSIS

- Tests without any advanced technique performed poorly with the perplexity and BLEU score
- Cross-head attention mechanism performed the best in comparison to Scaled-dot product and Multi-head attention
- Overfitting did not occur with the Cross-head attention mechanism
- All tests struggled with the advanced text
- Perplexity score improved with each test no matter the text level
- Shows how the model learned from each test



```
constants.py  inference_engine.py  model_builder.py  opts.py      tests        transforms
decoders      __init__.py          models            __pycache__  trainer.py   translate
alice@ubntlpt:~/nmt/OpenNMT-py/onmt$ cd ..
alice@ubntlpt:~/nmt/OpenNMT-py$ nano x.txt
alice@ubntlpt:~/nmt/OpenNMT-py$ python3 translate.py --model ende-large-withoutBT.pt --src x.txt --replace_unk --verbose
/home/alice/.local/lib/python3.10/site-packages/torch/cuda/__init__.py:138: UserWarning: CUDA initialization: The NVIDIA dr
iver on your system is too old (found version 9010). Please update your GPU driver by downloading and installing a new vers
ion from the URL: http://www.nvidia.com/Download/index.aspx Alternatively, go to: https://pytorch.org to install a PyTorch
version that has been compiled with your version of the CUDA driver. (Triggered internally at ../c10/cuda/CUDAFunctions.cpp
:108.)
  return torch._C._cuda_getDeviceCount() > 0
[2023-12-08 23:20:03,224 INFO] Loading checkpoint from ende-large-withoutBT.pt
[2023-12-08 23:20:05,120 INFO] Loading data into the model
[2023-12-08 23:20:14,383 INFO]
SENT 1: ['when', 'you', 'look', 'into', 'the', '<unk>', ',', 'the', '<unk>', 'also', 'looks', 'into', 'you', '.', 'he', 'wh
o', 'fights', 'with', '<unk>', 'should', 'be', 'careful', '<unk>', 'he', 'thereby', 'become', 'a', 'monster', '.', 'and', '
if', 'you', '<unk>', 'long', 'enough', 'into', 'an', '<unk>', ',', 'the', '<unk>', 'also', '<unk>', 'into', 'you', '.', '<u
nk>', 'fights', '<unk>', 'should', 'see', 'to', 'it', 'that', 'in', 'the', 'process', 'he', 'does', 'not', 'become', 'a', '
monster', '.']
PRED 1:  ⟪mrk_case_modifier_C⟫  wenn  ⟪mrk_case_modifier_C⟫  sie in die <unk> , die <unk> auch in  ⟪mrk_case_modifier_C⟫  s
ie hinein■ schauen .  ⟪mrk_case_modifier_C⟫  wer mit <unk> k■ ämpft ■, sollte vorsichtig sein <unk> er wird dadurch zu eine
m  ⟪mrk_case_modifier_C⟫  monster . und wenn  ⟪mrk_case_modifier_C⟫  sie lang genug in eine <unk> , die <unk> auch in  ⟪mrk
_case_modifier_C⟫  sie . <unk> k■ ämpft <unk> sollte dafür sorgen ■, dass er in dem  ⟪mrk_case_modifier_C⟫  prozess nicht z
u einem  ⟪mrk_case_modifier_C⟫  monster wird .
PRED SCORE: -0.4698

Time w/o python interpreter load/terminate:  11.21640658378601
alice@ubntlpt:~/nmt/OpenNMT-py$
```

INTERMEDIATE LEXILE TEST #1

PURPOSE

- Improve Neural Machine Translation on low-resource datasets through attention mechanisms
- Improve accessibility of translation using Artificial Intelligence
- Match the cognitive interpretations the human brain has towards linguistics
- Better translation for high-resource and low-resource languages alike
- Allow for translation that is accessible, accurate, and context-sensitive
- Chip away at language barriers and facilitate multilingual communication to anyone with access to technology
- By improving Neural Machine Translation models through testing of advanced techniques and further training, models can become more more equipped to handle input variations and become faster in training and interference

FUTURE RESEARCH

- Slowly work to revive dead or indigenous languages
- Train a model from scratch using a low-resource dataset
- Use additional language pairs apart from English-German to make the research more extensive
- Additional hours of training between tests to highlight the full benefits of the techniques used
- Efforts to reduce bias in the model
  - Could be made by researching the data the model was trained on, keeping in mind cultural biases
- Data would be more accurate with input texts longer than 50 words
  - In order to test the model for longer dependencies

CONCLUSION

- **Contrary to the hypothesis, Cross attention mechanism improved the translation on the small dataset the best**
  - As shown by the BLEU and PLP scores
- Implementation of attention mechanisms improved the translation, to some extent
- Potential to help in creating machine translation models for languages with low-resource language datasets
- Results are constrained by the specific, high-resource English-German dataset that the model was originally trained on and could not accurately predict results for any model
- Results could be better assessed with additional evaluation metrics such as a **human evaluator**

REFERENCES

- *YouTube: Home, https://www.canva.com/design/DAF9rJANs6g/HXb1o5BVEtwaFY5qenOhCA/edit. Accessed 22 November 2023.*
- *Christian, Jon. "Why Is Google Translate Spitting Out Sinister Religious Prophecies?" VICE, 20 July 2018, https://www.vice.com/en/article/j5npeg/why-is-google-translate-spitting-out-sinister-religious-prophecies. Accessed 22 November 2023.*
- *Grinevičius, Jonas, and Daniel Marsh. "People Tested How Google Translates From Gender Neutral Languages And Shared The "Sexist" Results." Bored Panda, 22 March 2021, https://www.boredpanda.com/google-translate-sexist/. Accessed 22 November 2023.*
- *Singh, Akash. "Neural Machine Translation : Superior Seq2seq Models With OpenNMT | by Akash Singh | Saarthi.ai." Medium, 5 June 2019, https://medium.com/saarthi-ai/neural-machine-translation-using-opennmt-b5e366e92c6. Accessed 22 November 2023.*
- *Zakowski, Igor. "Raining Cats Dogs Stock Illustrations – 52 Raining Cats Dogs Stock Illustrations, Vectors & Clipart." Dreamstime.com, https://www.dreamstime.com/illustration/raining-cats-dogs.html. Accessed 22 November 2023.*