# Machine Translation
# BDRP

Buse Ozer
Eugen Patrascu

# What is Machine Translation?

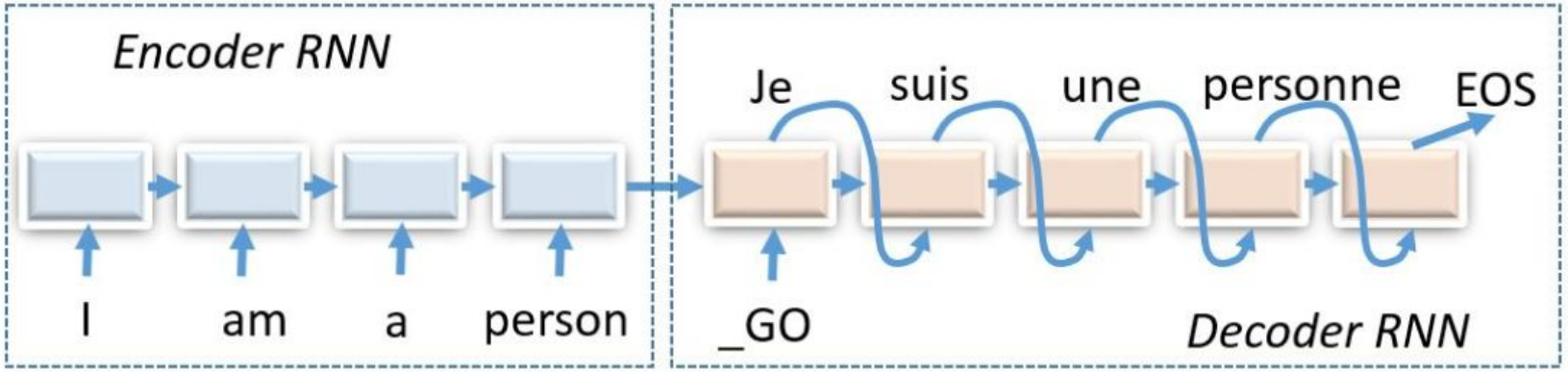
Machine Translation is a sub-field of computational linguistics that aims to automatically translate text from one language to another.

Demand of MT has grown exponentially over past couple of years, considering the enormous exchange of information between different regions with different regional languages

# Challenges in Machine Translation?

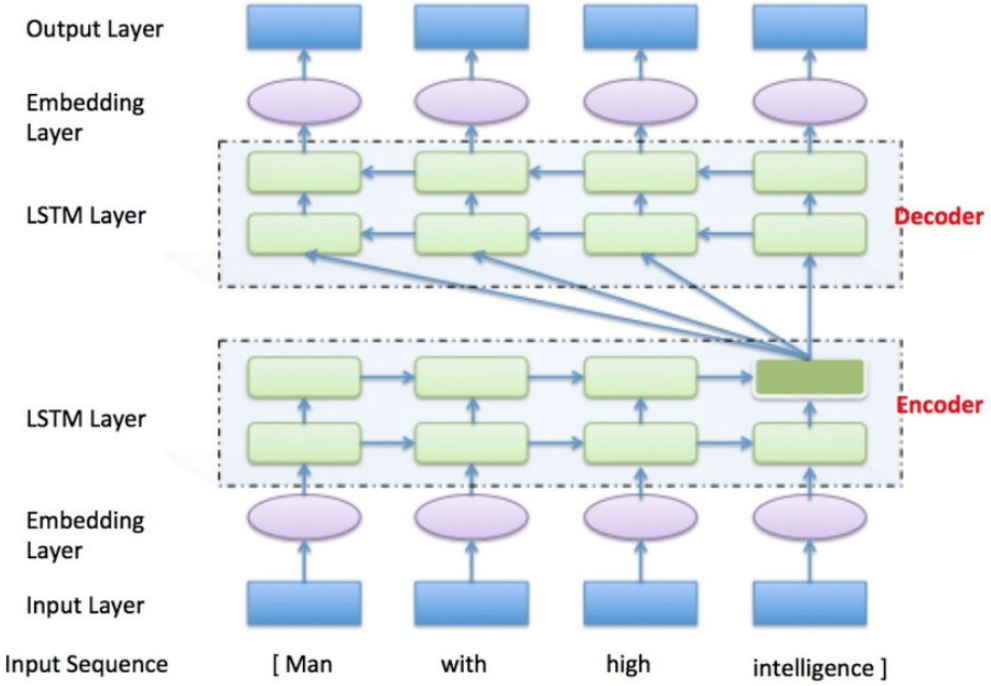
- Not all words in one language has equivalent word in another language
- Two given languages may have completely different structures
- Words can have more than one meaning
- There is no unique way for a word, sentence to  be translated

# Related Work



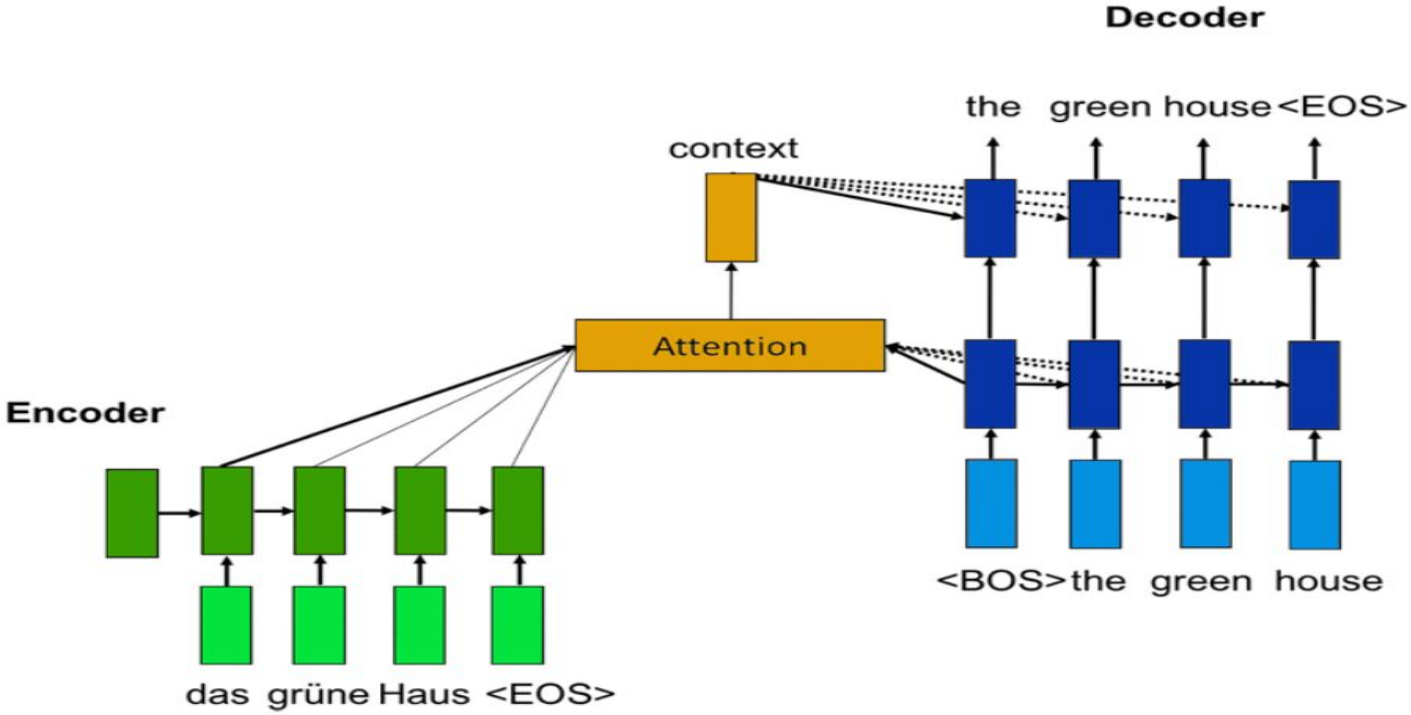

Learning phrase representations using RNN encoder-decoder for statistical machine translation, Cho, 2014

# Related Work




Sequence to sequence learning with neural networks, Sutskever 2014

# Related Work




Neural machine translation by jointly learning to align and translate, Bahdanau

# Related Work

- Google’s neural machine translation system: Bridging the gap between human and machine translation, Wu 2016: used in Google Translate
- Convolutional sequence to sequence learning, Gehring 2017: by Facebook, outperformed GMT
- Attention is all you need, Vaswani 2017: currently best results

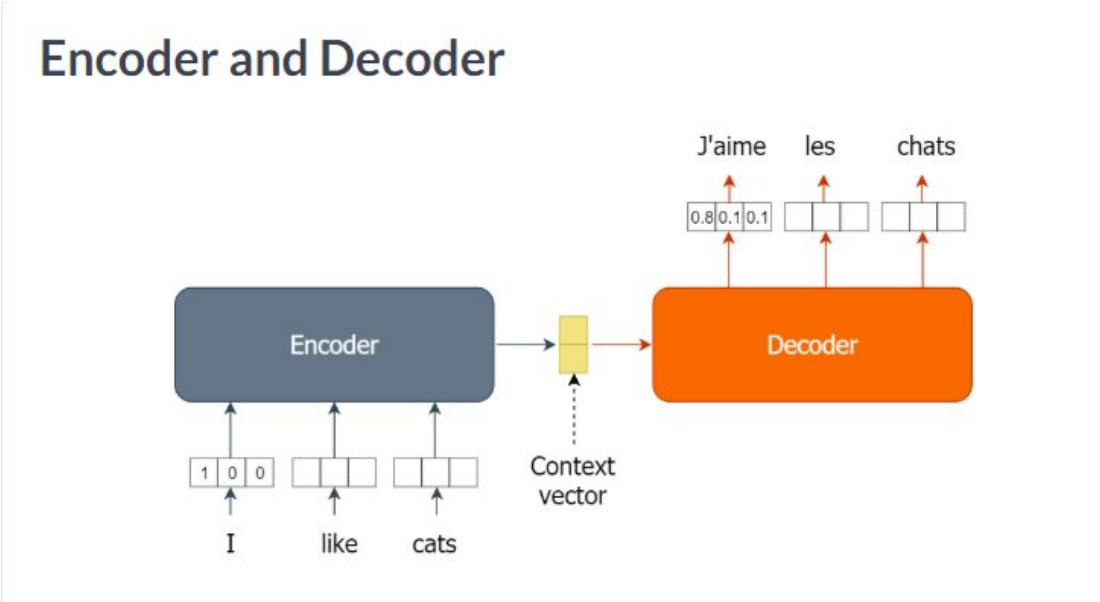
# Problem Definition

Analysing the impact of different configurations/strategies on the results of machine translation accuracy, considering the constraint of low computational resources.

Strategies:

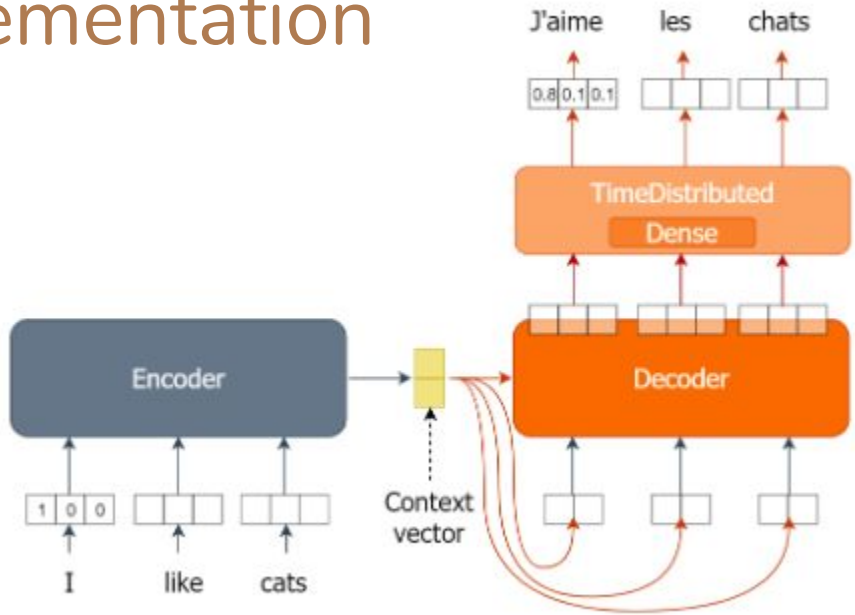
- Using more data for training and testing
- Unidirectional vs Bidirectional layer in the encoder
- Shallow vs Deep 2-layer network
- Using different numbers of nodes within a LSTM unit
- Word embeddings

# Base Model for Machine Translation




- Word embeddings
- 1-layer Encoder with LSTM units
- 1-layer Decoder with LSTM units
- Adam optimizer
- Categorical cross-entropy loss function

# Keras Implementation




```
def create_model(vocab_size_src, vocab_size_trg, seq_size_src, seq_size_trg, n_nodes):
    nn = Sequential()
    nn.add(Embedding(vocab_size_src, n_nodes, input_length=seq_size_src, mask_zero=True))
    nn.add(LSTM(n_nodes))
    nn.add(RepeatVector(seq_size_trg))
    nn.add(LSTM(n_nodes, return_sequences=True))
    nn.add(TimeDistributed(Dense(vocab_size_trg, activation='softmax')))
    return nn
```

# Environments/Libraries



NLTK

A Tool Kit for Natural Language Processing


# Data

Tatoeba project - Tab delimited Bilingual Sentence Pairs: https://tatoeba.org/eng

- Afrikaans - English afr-eng.zip (749)
- Albanian - English sqi-eng.zip (408)
- Algerian Arabic - English arq-eng.zip (155)
- Arabic - English ara-eng.zip (11175)
- Azerbaijani - English aze-eng.zip (2072)
- Basque - English eus-eng.zip (666)
- Belarusian - English bel-eng.zip (3698)

| | | |
|---|---|---|
| 4 | Run! | Courez ! |
| 5 | Who? | Qui ? |
| 6 | Wow! | Ça alors ! |
| 7 | Fire! | Au feu ! |
| 8 | Help! | À l'aide ! |

- Train and test on French to English sentence corpus
- Total data available: 170K+ sentences
- Initially working with 10K sentences
- For the model we require the size of the vocabulary and maximum length of a sentence for either language

# Data Preprocessing

Text pre-processing:

- Lowercase all the words
- Normalising language-specific characters (e.g. é to e)
- Remove punctuation
- Remove non-alphabetic characters

Input and output pre-processing:

- Input: word embeddings
- Output: one hot vectors

# Model Evaluation

Bilingual Evaluation Understudy, BLEU

- An algorithm for evaluating the quality of text which has been machine-translated from one language to another.
- The metric is that "the closer a machine translation is to a professional human translation, the better it is".
- A metric which is highly correlated with human judgments of quality

# Experiments

## Experiment 1: Impact of data set size

| | 1 LSTM layer of encoder, 1 LSTM layer of decoder | |
|---|---|---|
| Metric | 10000 phrases | 20000 phrases |
| BLEU-1 | 0.451 | 0.576 |
| BLEU-2 | 0.326 | 0.462 |
| BLEU-3 | 0.265 | 0.402 |
| BLEU-4 | 0.135 | 0.238 |

## Experiment 2: Impact of bidirectional layer

| Metric | Base model (10K) | Bidirectional encoder (10K) |
|---|---|---|
| BLEU-1 | 0.54732 | 0.56076 |
| BLEU-2 | 0.44148 | 0.44965 |
| BLEU-3 | 0.35848 | 0.38341 |
| BLEU-4 | 0.15688 | 0.18903 |

# Experiments

## Experiment 3: Shallow vs Deep network

| Metric | Base model (10K) | Base model (20K) | Deep network (10 K) | Deep network (20K) | Deep network (30K) |
|---|---|---|---|---|---|
| BLEU-1 | 0.54732 | 0.5766 | 0.41228 | 0.53371 | 0.55356 |
| BLEU-2 | 0.44148 | 0.46271 | 0.27795 | 0.4151 | 0.43459 |
| BLEU-3 | 0.35848 | 0.40283 | 0.19157 | 0.34934 | 0.37877 |
| BLEU-4 | 0.15688 | 0.2384 | 0.0629 | 0.18943 | 0.23749 |

## Experiment 4: Different number of units within LSTM

| | 1 LSTM layer of encoder, 1 LSTM layer of decoder | | |
|---|---|---|---|
| | 10000 phrases; 9000 for training, 1000 for testing | | |
| Metric | 128 units | 256 units | 512 units |
| BLEU-1 | 0.369 | 0.451 | 0.563 |
| BLEU-2 | 0.228 | 0.326 | 0.463 |
| BLEU-3 | 0.145 | 0.265 | 0.394 |
| BLEU-4 | 0.041 | 0.135 | 0.201 |

# Experiments

## Experiment 5: Impact of word embeddings

| | 1 LSTM layer of encoder, 1 LSTM layer of decoder | |
|---|---|---|
| Metric | Word2Vec | pre-trained Word2Vec |
| BLEU-1 | 0.55 | 0.561 |
| BLEU-2 | 0.428 | 0.438 |
| BLEU-3 | 0.364 | 0.371 |
| BLEU-4 | 0.205 | 0.216 |

## Experiment 6: Different language pair DE - EN

| Metric | Base model (10K) for FR-EN | Base model (10K) for DE-EN |
|---|---|---|
| BLEU-1 | 0.54732 | 0.53537 |
| BLEU-2 | 0.44148 | 0.40294 |
| BLEU-3 | 0.35848 | 0.31645 |
| BLEU-4 | 0.15688 | 0.14515 |

# Conclusions

- Crucial to have a good amount of high-quality training data
- Important to have a sufficiently complex architecture
  - Bidirectional LSTM layer
  - Having deep network depending on the amount of data
  - Having sufficient number of units within a layer

- Differences on the translation quality depending on the different pair of languages

# Future Work

- Deeper networks such as 4 layers
  - requires higher computing power and millions of sentences
- Training with more data
- Attention mechanism
- Pre-trained word embeddings on huge corpuses

# References


Slide 4, https://rickyhan.com/jekyll/update/2017/09/14/autoencoders.html
Slide 5, https://www.researchgate.net/figure/LSTM-Encoder-Decoder-Model_fig1_308072447
Slide 6, https://aws.amazon.com/blogs/machine-learning/train-neural-machine-translation-models-with-sockeye/
Slides 9,10, https://www.datacamp.com/home
Slide 11, https://abstract-technology.fr/media/technology/python-logo.png/image_view_fullscreen
Slide 12, http://www.manythings.org/anki/

# Overleaf

https://www.overleaf.com/project/5dc3f746ca27c80001d30f2a