French-to-English Machine Translation

A Comparison of Embedding Methods and Model Architectures

Travis Twigg, Dmitry Pankratov, Rahul Gite,
Nikhil Goparapu, Abhiram Kalidindi

Data Science Graduate Program,
University of Maryland, Baltimore County

Fall 2020

"The difference between the right word and the almost right word is really a large matter – it's the difference between lightning and a lightning bug"

– Mark Twain

Outline

- Objectives of the research
- Background
- Related studies
- Methodology
- Results
- Discussion
- References
- Demo
- Questions

Objectives of the research

Objectives:

- To compare the performance of different word embedding methods: learned and pre-trained.
- To analyze the impact of Attention on an encoder-decoder network.
- To develop a machine translation scoring method that evaluates both syntactic and semantic similarities between sentences.

Hypotheses:

- The addition of Attention to a encoder-decoder network will improve translation performance.
- The use of pre-trained word embeddings will decrease training time and improve translation performance.

Background

Machine Translation:

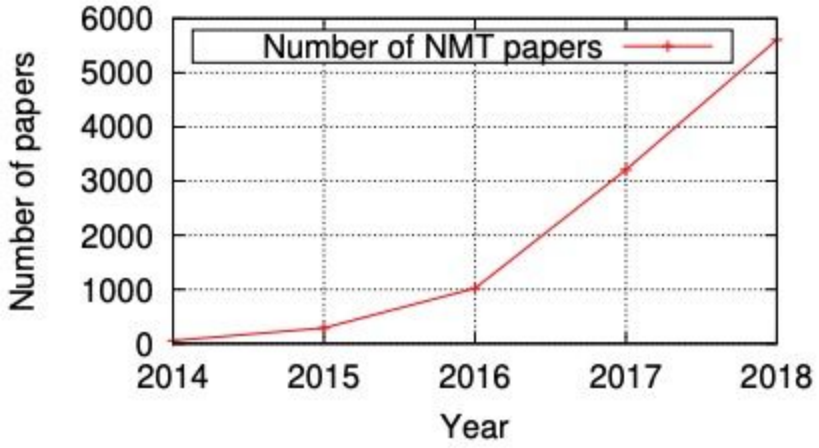
- The automatic translation of text from one natural language into another.
- One of the earliest goals for computers - proposed 2 years after the creation of ENIAC. [1]
- Increasingly important in current era of globalization.

Three main approaches:

- Rule-based (RBMT): relies on linguistic rules and millions of bilingual dictionary pairs.[1]
- Statistical (SMT): relies on statistical models that learn to translate text – data driven and requires pipeline of specialized systems.[1]
- Neural (NMT): relies on single end-to-end neural network to learn a statistical model.[1]

Neural Machine Translation

- One of the most challenging AI tasks given the inherent ambiguity and fluidity of human language.[1]
- Achieves state-of-the-art results compared to SMT. Mostly due to the single system approach and use of vector embeddings. [1]
- Widely adopted after 2014 with the development of Encoder-Decoder networks, and 2015 with the creation of Attention Mechanisms. [2]
- Deployed in production systems by Google, Microsoft, Facebook, Amazon.[2]



Number of papers mentioning “neural machine translation” per year according Google Scholar.[2]

Literature Review

The use of attention mechanisms to improve machine translation has been covered extensively in research literature:

- Bahdanau et al. (2015): Local attention mechanism.
- Luong et al. (2015): Global attention mechanism.
- Vaswani et al. (2017): Transformer network.

The same is true for semantic evaluation of translation results:

- Wong et al. (2010): Use of latent semantic analysis (LSA).
- Agirre et al. (2015): Semantic textual similarity tool.
- Wieting et al. (2019): SimiLe, a continuous metric for semantic similarity.

Literature Review

In contrast, the use of pre-trained word embeddings in machine translation has received much less attention.

Qi et al. (2018) showed that:

- Pre-trained embeddings are most effective where there is very little training data. [3]
- Pre-trained embeddings are more effective if the two languages in the translation pair are more linguistically similar. [3]

After a thorough literature search, our group was not able to find any research contrasting embedding methods for machine translation.

Dataset

135,000 English-French sentence pairs from Tatoeba.org,
a large database of example sentences translated into many languages.

Advantages:

- Free and widely used database.
- Large corpus of English-French pairs.

Disadvantages:

- Some translations are incorrect.
- Some sentences are grammatically inaccurate.
- Multiple translations exist for the same sentence.

He's a powerful sorcerer.	*C'est un puissant sorcier.*
Do you like music?	*Aimes-tu la musique ?*
I'm a big coffee drinker.	*Je suis une grosse buveuse de café.*

Preprocessing Methods

Filtering dataset:
Max length < 10, Starts with = [‘I’m’, ‘He’s’, ‘She’s’, ‘You’re’, ‘We’re’, ‘They’re’]

Removal of french characters with accents by converting Unicode text to plain ASCII text

Lowercasing

Replacement of punctuation with whitespace (with the exception of periods, commas, exclamation marks, and question marks)

Removal of extra whitespace

Tokenization

135,842 pairs → 10,599 pairs

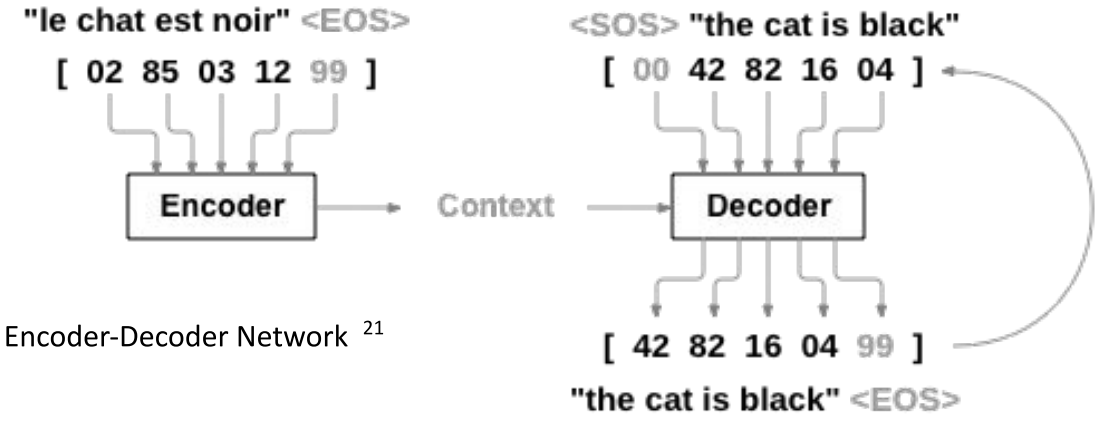
Example:

I love it !	J'adore ça !
I love it !	J'adore ca !
i love it !	j'adore ca !
i love it !	j'adore ca !
i love it !	j adore ca !
[‘i love it !’,	‘j adore ca !’]

Model Architecture

Sequence to sequence (Seq2seq):

- Family of encoder-decoder architectures used in many language processing tasks.
- Developed by Google in 2014. [4]
- Maps variable-length sequences from one domain (French) to variable-length sequences in another domain (English).



Encoder-Decoder Network [21]

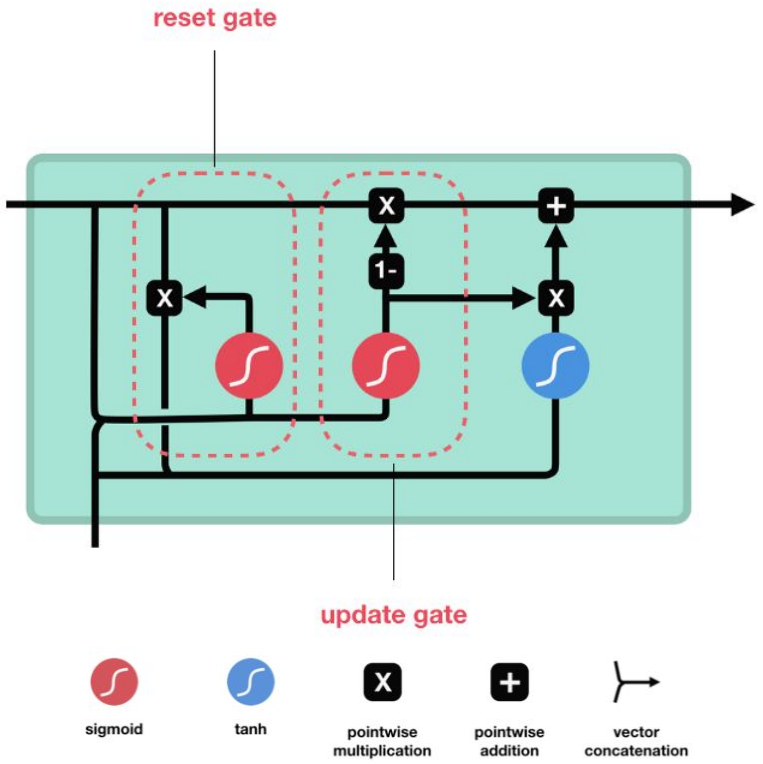
Model Architecture

Gated Recurrent Units (GRU):

- Solves the long term memory problem of RNNs.
- The update gate decides what information to throw away and what new information to add.
- The reset gate is used to decide how much past information to forget. [5]

Teacher forcing:

- Strategy for training a RNN that uses the model output from the prior time step as input in the current time step. order to overcome an issue of slow convergence, instability and poor results while training. [6]



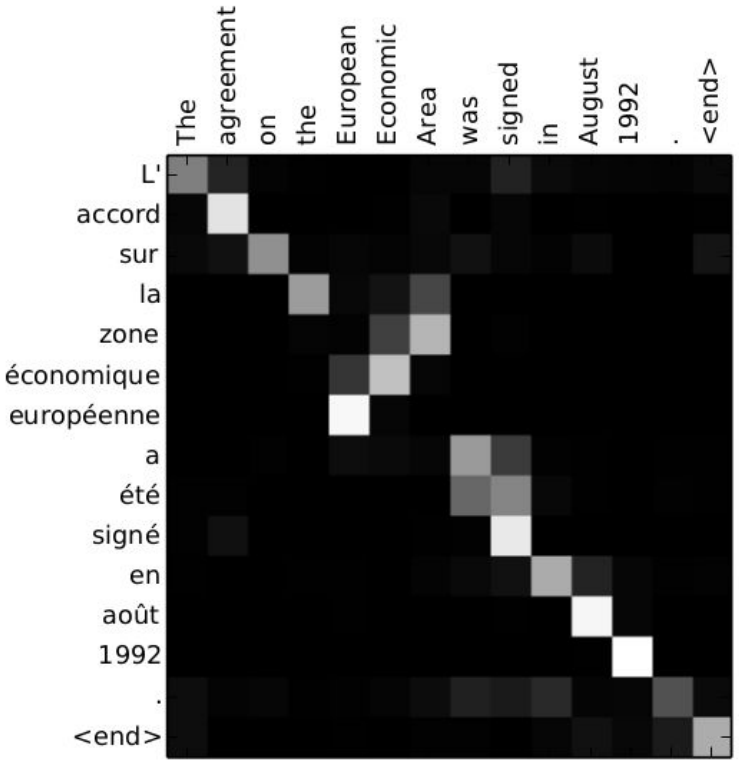
GRU Cell [7]

Model Architecture

Attention Mechanisms

- Proposed by Bahdanau et al. in 2015.
- Helps the decoder learn to focus over a specific range of the input sequence, instead of the whole sequence.
- Helps solve the alignment problem, as words can be arranged differently depending on the language. [5]

Figure on the right displays the attention weights and illustrates which part of the input sequence the model is focusing on at each time step.

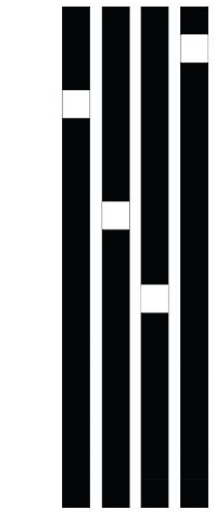


Attention Weights [8]

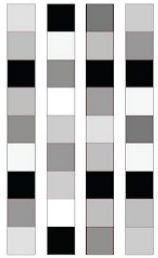
Embedding Methods

Word embeddings

- Learned representation of text where words that have a similar meaning have a similar representation.
- One of the key breakthroughs responsible for the impressive performance of deep learning methods on challenging natural language processing problems. [9]
- In our study, we utilized two word embedding techniques: learned embeddings and pre-trained embeddings. The pre-trained embeddings include GloVe, Word2Vec, frWac, and FastText.



One-hot word vectors:
- Sparse
- High-dimensional
- Hard-coded



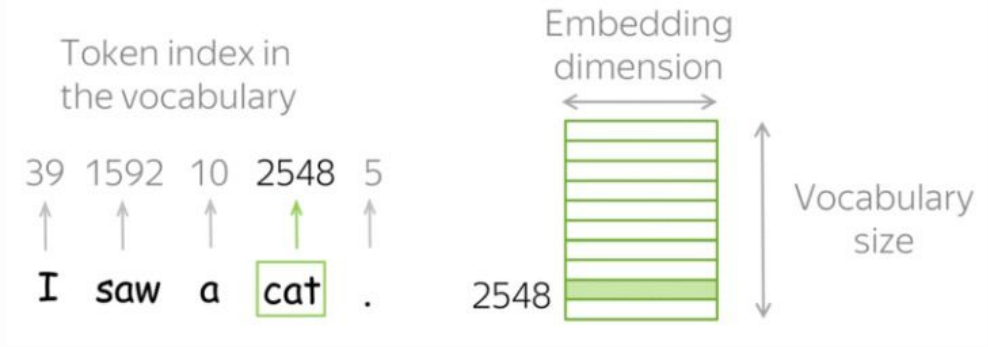
Word embeddings:
- Dense
- Lower-dimensional
- Learned from data

Vector Embeddings [10]

Learned Embeddings

Learned Embeddings:

- Embeddings that are trained from randomized values for a specific dataset and task.
- Acts as a trainable look-up table for word vectors.
- When learned embeddings are trained on a particular task, they tend to perform better compared to a more generalized pre-trained embedding, but this is highly dependent on dataset size and training time. [16]

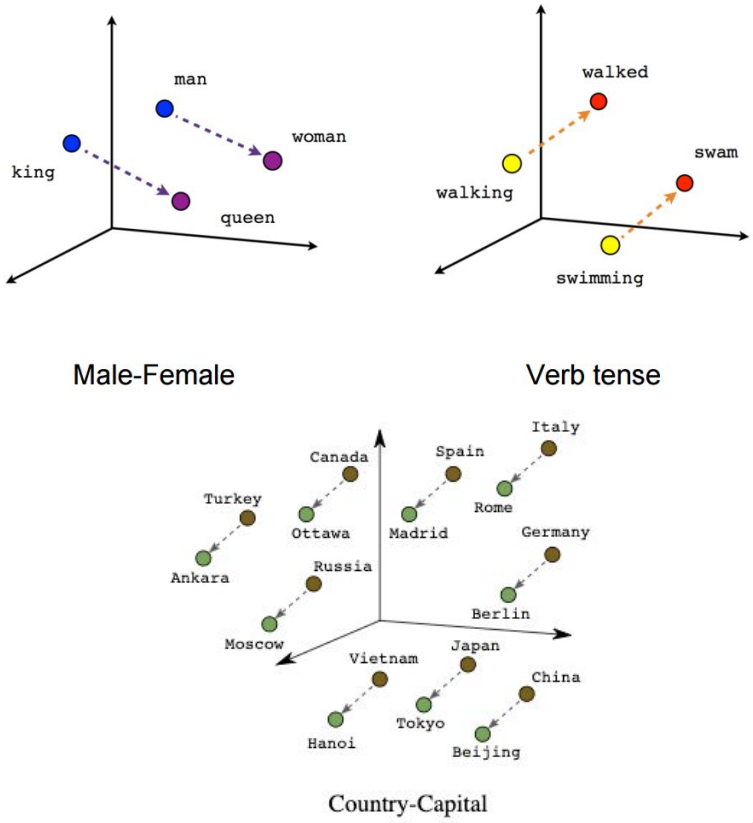


Embedding Look-up Table [11]

Pre-trained Embeddings

Word2Vec

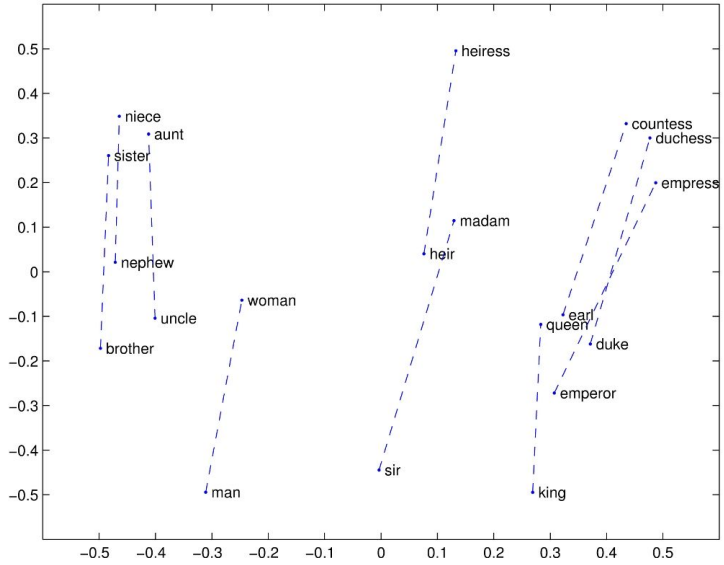
- Word2Vec is a statistical method for efficiently learning word embeddings from a text corpus.
- Developed by Mikolov, et al. at Google in 2013. [9]
- This method introduced two different learning models to learn embeddings: the Continuous Bag-of-Words model (CBOW) and the Continuous Skip-Gram model. [9]
- In our project we used 300-dimensional vectors representing 3 million English words trained on the Google News Dataset.



Pre-trained Embeddings

Global Vectors for Word Representation (GloVe)

- GloVe is an unsupervised learning algorithm for obtaining vector representations for words.
- Developed by Pennington, et al. at Stanford in 2014. [9]
- GloVe combines the power of matrix factorization techniques (such as LSA) at using global text statistics with Word2Vec's ability to use local statistics to create vector space model representations. [9]
- GloVe embeddings used in this project are 100-dimensional vectors representing 400,000 english uncased words and were trained on Wikipedia 2014 and Gigaword 5.



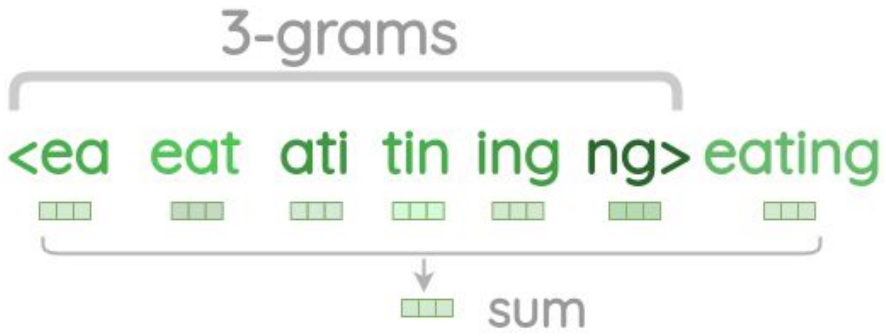
Linear substructures of words in vector space.

Glove Vectors [13]

Pre-trained Embeddings

Fastext

- FastText is an extension to Word2Vec developed by Facebook in 2016. [9]
- Instead of feeding individual words into the model FastText breaks words into several n-grams (sub-words). [9]
- The FastText embeddings used in our project are 300-dimensional vectors representing 1 millions words trained on Wikipedia 2017, UMBC domain corpus, and statmt.org news dataset.



Tri-gram subword generation. [14]

Pre-trained Embeddings

frWac2Vec

- French words are embedded using frWac2Vec, a pre-trained modified word2vec model developed by Jean-Philippe Fauconnier at Apple. [17]
- frWac2Vec is a 200 dimensional word embedding model created using the continuous skip-gram model architecture. [17]
- frWac2Vec consists of 1.6 billion word corpus constructed from the Web limiting the crawl to the .fr domain. [17]

Evaluation

Metrics

Four evaluation metrics were utilized to test the performance and robustness of our models: BLEU score, GLEU score, a custom scoring method, and comparison against Google Translate.

BLEU:

Calculates a modified precision of n-grams (unigram to four-gram) for translation and reference sentences. [18]

GLEU

Calculates the precision and recall for n-grams (unigram to four-gram) and returns the minimum. Compared to BLEU, it approaches human judgments more closely, but can overly penalize for altered word order. [18]

Input:	je suis content de te voir .
Target:	i am happy to see you .
Prediction:	i am glad to see you .
BLEU Score:	0.83
GLEU Score:	0.50

Evaluation

Custom Scoring Method

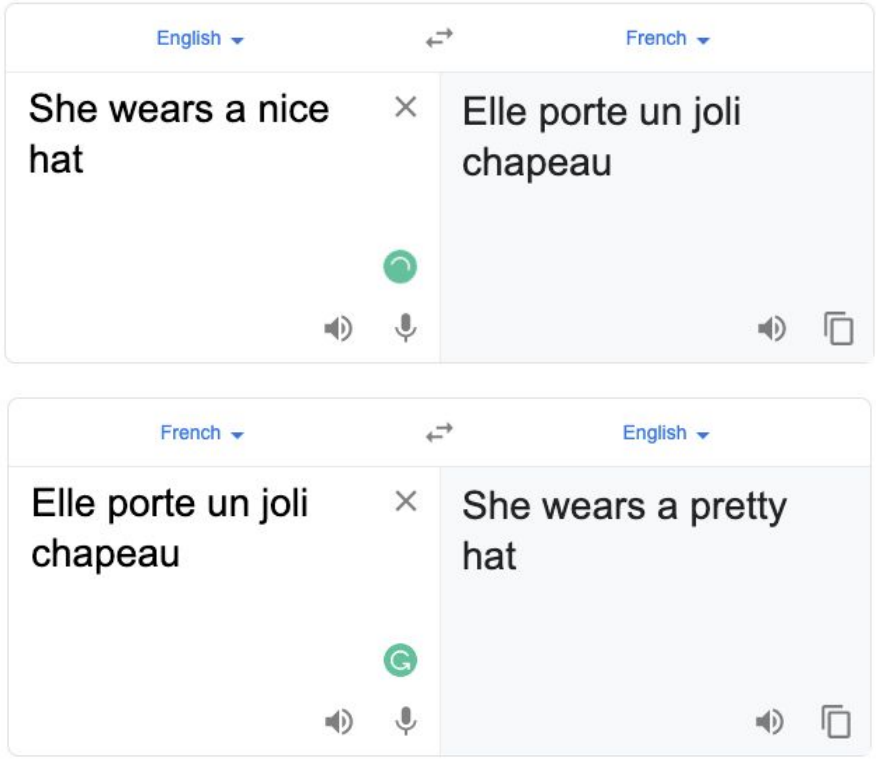
Custom Score = Weighted Average(BLEU+GLEU) + Weighted Semantic Bonus - Double Word Penalty

Input:	vous etes tous fous .
Target:	you are all mad .
Prediction:	you are all crazy .
BLEU Score:	0.750
GLEU Score:	0.600
Avg Score:	0.713
Semantic similarities:	['mad', 'crazy', 0.7385839]
Semantic similarity bonus:	+ 0.22157517671585084
Double word penalty:	- 0.0
Custom Score:	0.934

Evaluation

Google Translate

- The Google Translation API provides a simple interface for dynamically translating an arbitrary string into any supported language using state-of-the-art neural machine translation.
- At the moment of writing this paper for translation Google uses Transformer encoder and an RNN decoder architecture, implemented in TensorFlow. [19]
- The use of Google Translate as an evaluation tool gave us the ability to compare our model to a state-of-the-art architecture.



Google Translate Web Interface

Results

Evaluation Scores

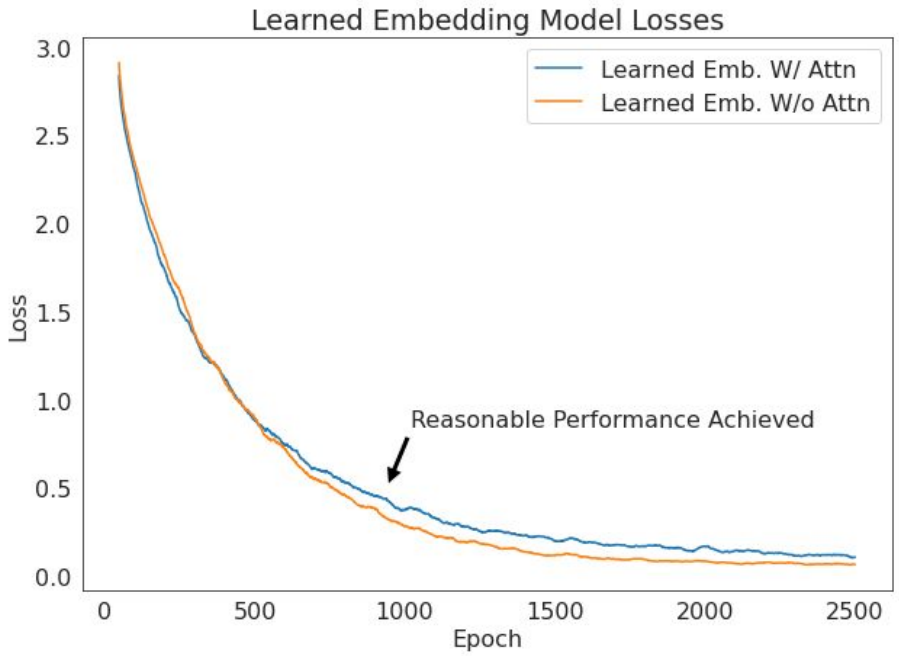
Embedding Method	*Avg BLEU*	*Avg GLEU*	*Avg Custom Score*	*Rank*
With Attention:				
Learned Embeddings	0.980	0.967	0.984	2
Without Attention:				
Learned Embeddings	0.984	0.973	0.988	1
Glove + frWac2Vec	0.764	0.676	0.815	4
FastText + frWac2Vec	0.759	0.669	0.810	5
Word2Vec + frWac2Vec	0.710	0.604	0.764	6
Open Source Model				
Google Translate API	0.851	0.751	0.905	3

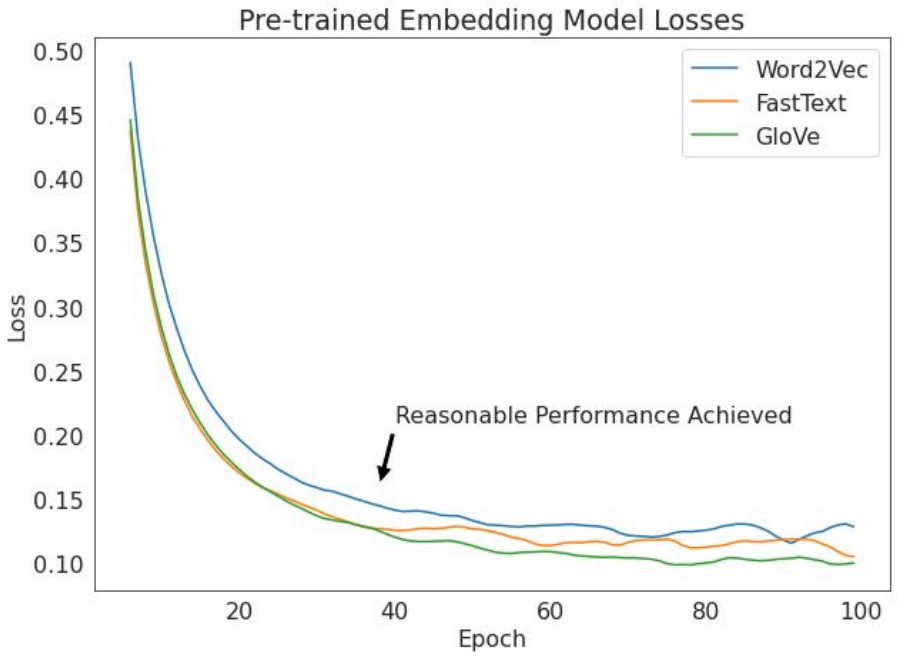
Sample Output

Input:	vous etes deux fois plus forts que moi .
Target:	you are twice as strong as i am .
GloVe:	you are twice as strong as i am .
Word2Vec:	you are twice as strong as me .
FastText:	you are twice as strong as me .
Learned Emb.:	you are twice as strong as i .
Input:	je cherche du travail .
Target:	i am looking for a job .
GloVe:	i am looking for a job .
Word2Vec:	i am looking for a job .
FastText:	i am looking for a job .
Learned Emb.:	i am looking for work .
Input:	nous en avons termine .
Target:	we are all done .
GloVe:	we are through .
Word2Vec:	we are done .
FastText:	we are finished already .
Learned Emb:	we are finished .

Results

Loss Curves





- Pre-trained embedding models required much less time to train.
- Requiring less than 40 epochs to achieve reasonable performance.

Discussion

Expected findings:

- GloVe and FastText embeddings outperformed Word2Vec embeddings for translation in the study.

Unexpected findings:

- The addition of an attention mechanism did not significantly improve translation performance but did lower the efficiency of the model (time per epoch).
- The models which utilized learned embeddings outperformed the models which utilized pre-trained word embeddings.
- The learned embedding models outperformed the Google Translate API, which is considered a state-of-the-art platform.

Discussion

Lessons learned:

- Pre-trained word embeddings, as useful as they are in many circumstances, aren't always the best method for natural language processing tasks.
- Decisions concerning choice of dataset and data processing steps can have a large impact on results

Challenges:

- A lack of knowledge in the French language made preprocessing decisions difficult.
- Developing an evaluation method that captures the syntactic and semantic similarities between translated sentences and references proved to be difficult and requires future research.

Future work:

- Exploring the impact of fine-tuning pre-trained embeddings on the training dataset
- Incorporating POS tagging into the learning and evaluation process.
- Exploring other architectures such as transformers.

Demo

Machine Translation Demo

```
1 evaluateRandomlySimplifiedOutput(encoder1, attn_decoder1)
```

```
EVALUATION OF MACHINE TRANSLATION MODELS
        Evaluating  10  examples...
```

```
1
```

+ Code

Merci.

References

[1] Brownlee, J. (2019, August 7). A Gentle Introduction to Neural Machine Translation. Machine Learning Mastery. https://machinelearningmastery.com/introduction-neural-machine-translation/

[2] Stahlberg, F. (2019). Neural Machine Translation: A Review and Survey. Arxiv.Org. https://arxiv.org/pdf/1912.02047.pdf

[3] Qi, Y. (2017). When and why are pre-trained word embeddings useful for neural. Aclweb.https://www.aclweb.org/anthology/N18-2084.pdf (2017).

[4] Sutskever, I. (2014, September 10). Sequence to Sequence Learning with Neural Networks. ArXiv.Org. https://arxiv.org/abs/1409.3215

[5] Antonio, M. (2019, October 24). Attention Mechanism in Seq2Seq and BiDAF — an Illustrated Guide. Medium. https://towardsdatascience.com/the-definitive-guide-to-bidaf-part-3-attention-92352bbdcb07

[6] Brownlee, J. (2019, December 18). A Gentle Introduction to Calculating the BLEU Score for Text in Python. Machine Learning Mastery. https://machinelearningmastery.com/calculate-bleu-score-for-text-python/

[7] Phi, M. (2020b, June 28). Illustrated Guide to LSTM's and GRU's: A step by step explanation. Medium. https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

[8] Z. (2017, July 14). Extracting and visualizing the decoder attention weights. OpenNMT Forum. https://forum.opennmt.net/t/extracting-and-visualizing-the-decoder-attention-weights/636

[9] Brownlee, J. (2019b, August 7). What Are Word Embeddings for Text? Machine Learning Mastery. https://machinelearningmastery.com/what-are-word-embeddings/

[10] Embeddings (2019). Deep learning for text. Manning. https://freecontent.manning.com/deep-learning-for-text/

References

[11] Word Embeddings. (2020).
Github. https://lena-voita.github.io/nlp_course/word_embeddings.html

[12] Khandelwal, R. (2019, December 28). Word Embeddings for NLP - Towards Data Science.
Medium. https://towardsdatascience.com/word-embeddings-for-nlp-5b72991e01d4

[13] Pennington, J. (2014). GloVe: Global Vectors for Word Representation.
Stanford. https://nlp.stanford.edu/projects/glove/

[14] Chaudhary, A. (2020, June 21). A Visual Guide to FastText Word Embeddings.
Amit Chaudhary. https://amitness.com/2020/06/fasttext-embeddings/

[16] M. Farahmand. (2019). Pre-trained word embeddings or embedding layer? - a dilemma.
Medium.https://towardsdatascience.com/pre-trained-word-embeddings-or-embedding-layer-a-dilemma-8406959fd76c.

[17] Fauconnier, J. (2017). Jean-Philippe Fauconnier.
Github. https://fauconnier.github.io/

[18] Wu, Y., Schuster M. (2016). Google's Neural Machine Translation System: Bridging the Gap between Human and Machine
Translation. ArXiv, https://arxiv.org/pdf/1609.08144.pdf

[19] Caswell, I. (2020, June 8). Recent Advances in Google Translate.
Google AI Blog. https://ai.googleblog.com/2020/06/recent-advances-in-google-translate.html

[20] Translators, F. F. (2020, March 17). Thirty Five of the Very Best Quotes on Translation.
Food For Translators. https://www.foodfortranslators.com/2014/10/26/best-quotations-on-translation/

[21] NLP From Scratch: Translation with a Sequence to Sequence Network and Attention — PyTorch Tutorials 1.7.0 documentation. (2017).
Pytorch. https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html