SENTENCE AUTO COMPLETION USING NLP

GROUP - 2

Trinadh Reddy Buthukuri
Sai Teja Reddy Pathika
Hitesh Rambhaskar Rachuri

University at Buffalo The State University of New York

Objective

- Sentence auto completion is just predicting what might be written next. Or, we can say autocomplete generates the next most probable word that goes on with whatever was written already.
- N-gram model is trained for this purpose. Basically N-gram is a sequence of N - words, for example “This Project” is a bi-gram because it contains two words. Based on the training data, the model will compute the best probable word.

Data Description

- The data consists of financial news headlines, with nearly 5000 observations.
- Number of unique words from the data are around 12k.

Data Source:

https://www.kaggle.com/datasets/ankurzing/sentiment-analysis-for-financial-news

One observation from the data:

A trilateral agreement on investment in the construction of a technology park in St Petersburg was to have been signed in the course of the forum , Days of the Russian Economy , that opened in Helsinki today .

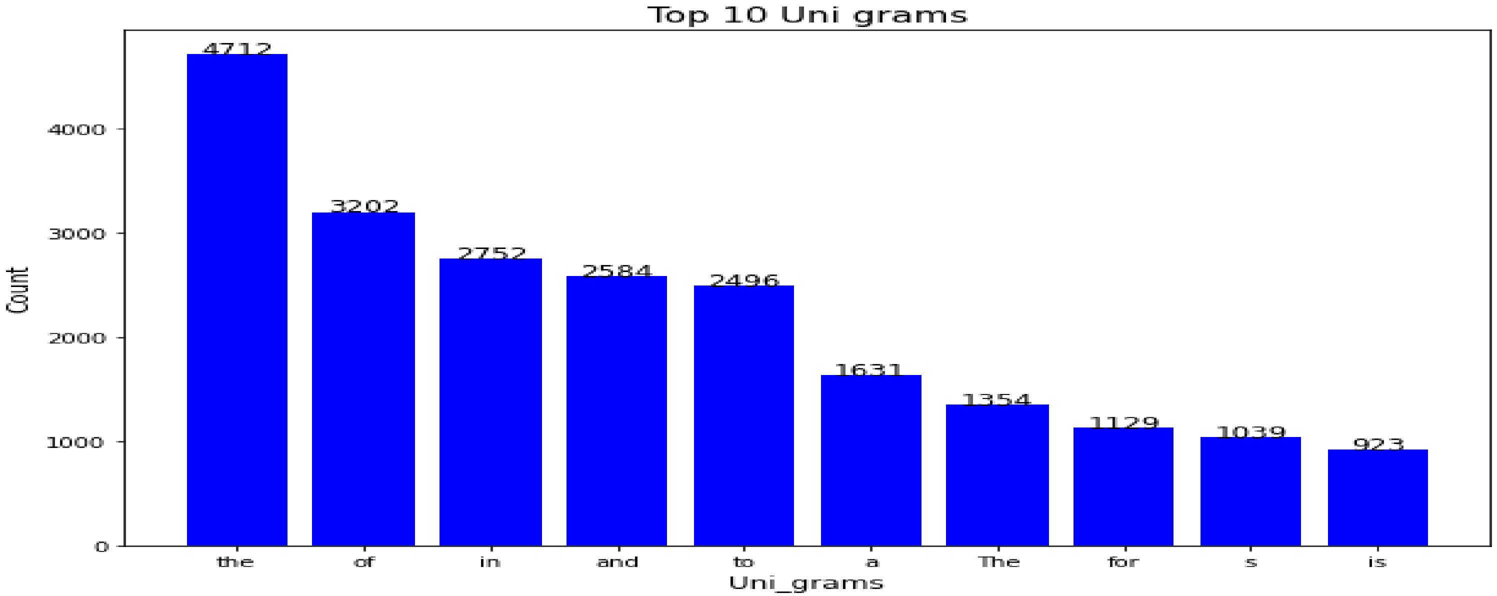
Data Preprocessing

- Removal of Null values.
- Removal of Extra spaces and Punctuation using regular expression.
- Word tokenization using Keras Tokenizer.
- Removal of stop words for N-gram Analysis.

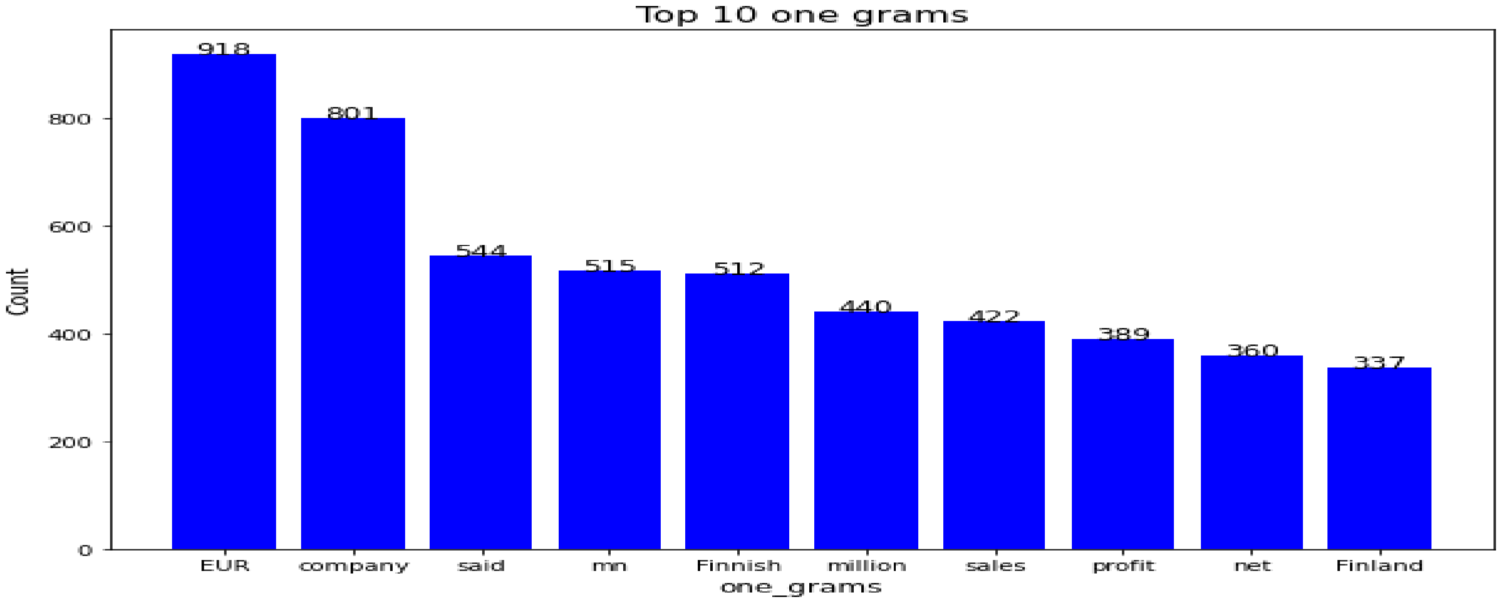
N-gram

- It is a sequence of N-words in a text-file or document.
- n=1 Unigram ; n=2 – Bi gram ; n=3 – Tri gram
- Example: Data Analysis -> Bi gram

Uni-gram Analysis

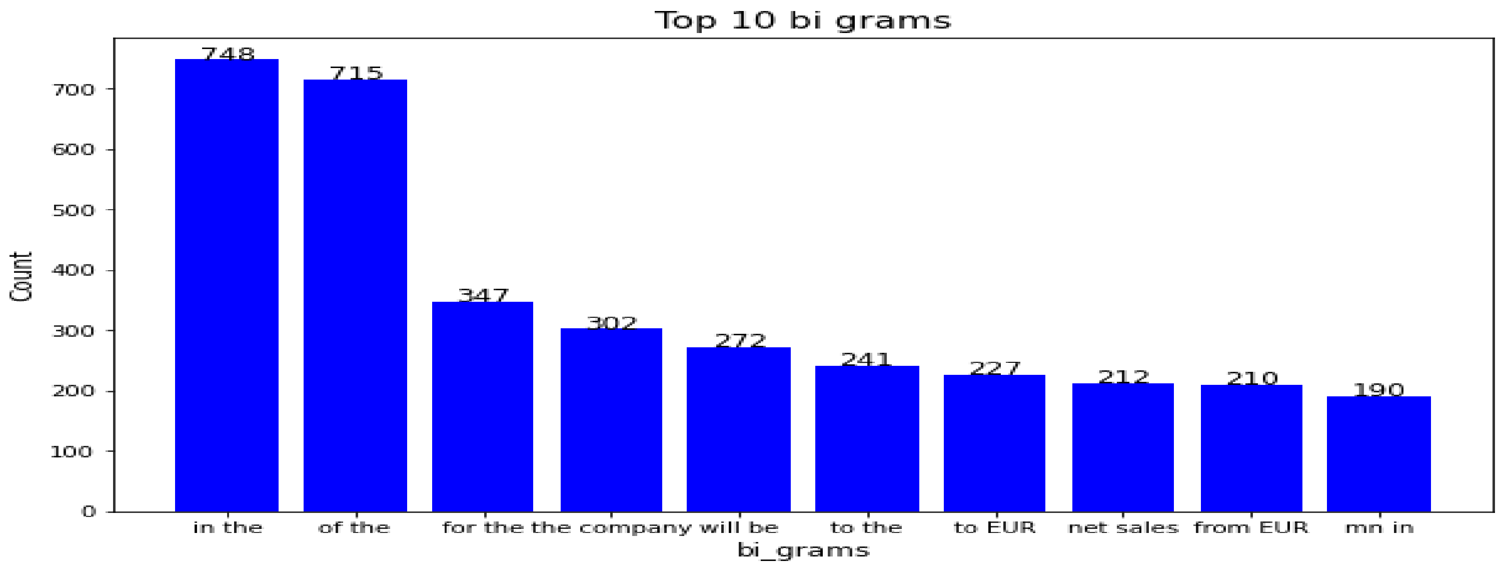


- The Bar chart consists of top-10 uni grams and their count from the data including stop words.
- “the” and “of ” are two most repeated uni grams.

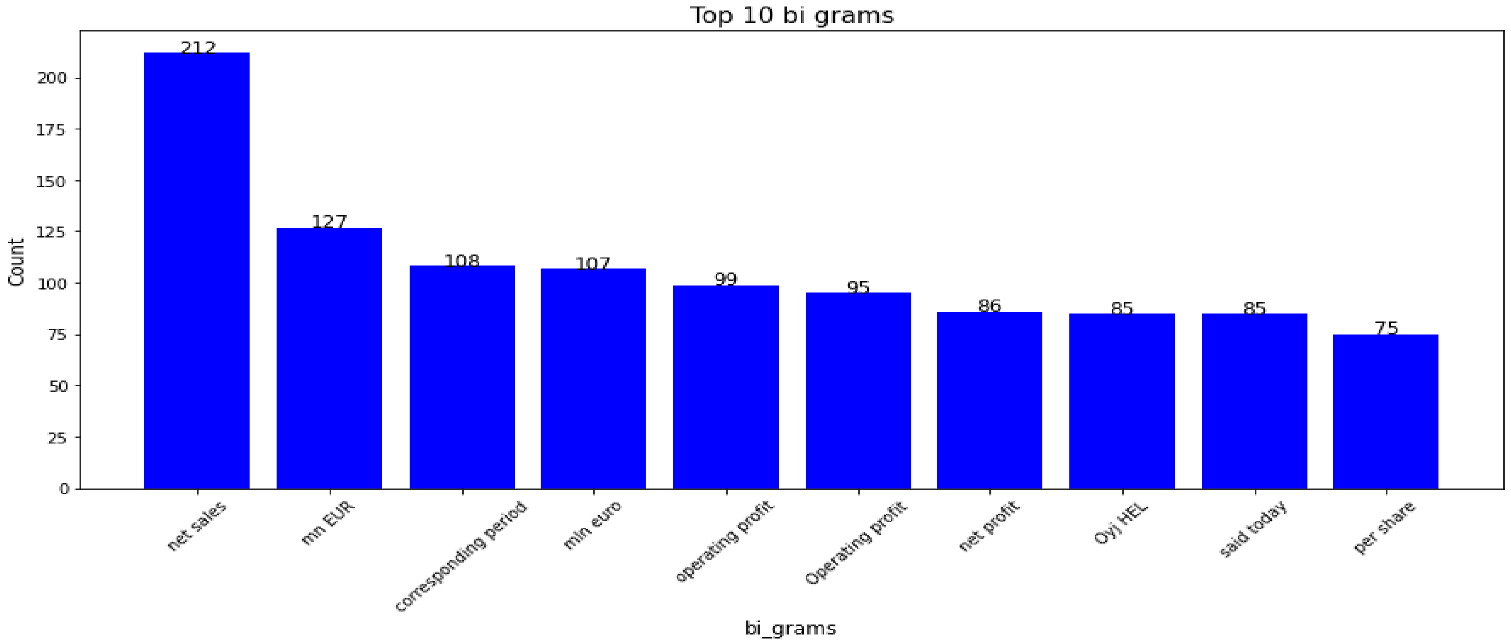


- The Bar chart consists of top-10 uni grams and their count from the data excluding stop words.
- “EUR” and “company” are two most repeated uni grams.

Bi-gram Analysis



- The Bar chart consists of top-10 bi grams and their count from the data including stop words.
- “In the” and “of the” are two most repeated bi grams.



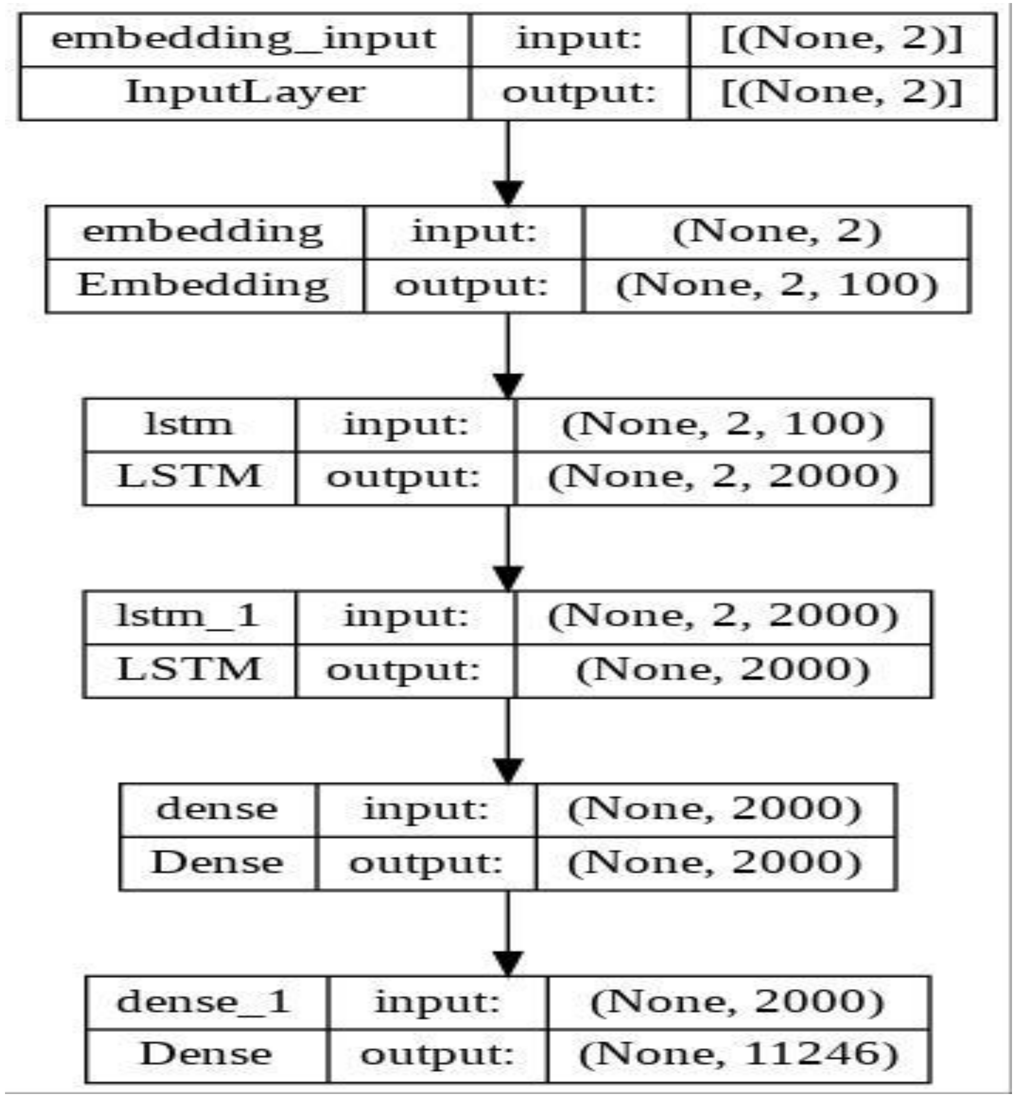
- The Bar chart consists of top-10 bi grams and their count from the data excluding stop words.
- “net sales” and “mn EUR” are two most repeated bi grams.

Training Data

FEATURES AND TARGET VARIABLE

- In a sentence every third consecutive word is the target variable of the previous two words.
- I am very happy.
  - For “I am” – “very” is the output.
  - For “I am very” – happy is the output.
- X – Feature ; Y – Target Variable
- After the word tokenization, for every 2 words appended to Feature (X) variable, the next word(output) is appended to the target(Y) variable.

Model Architecture



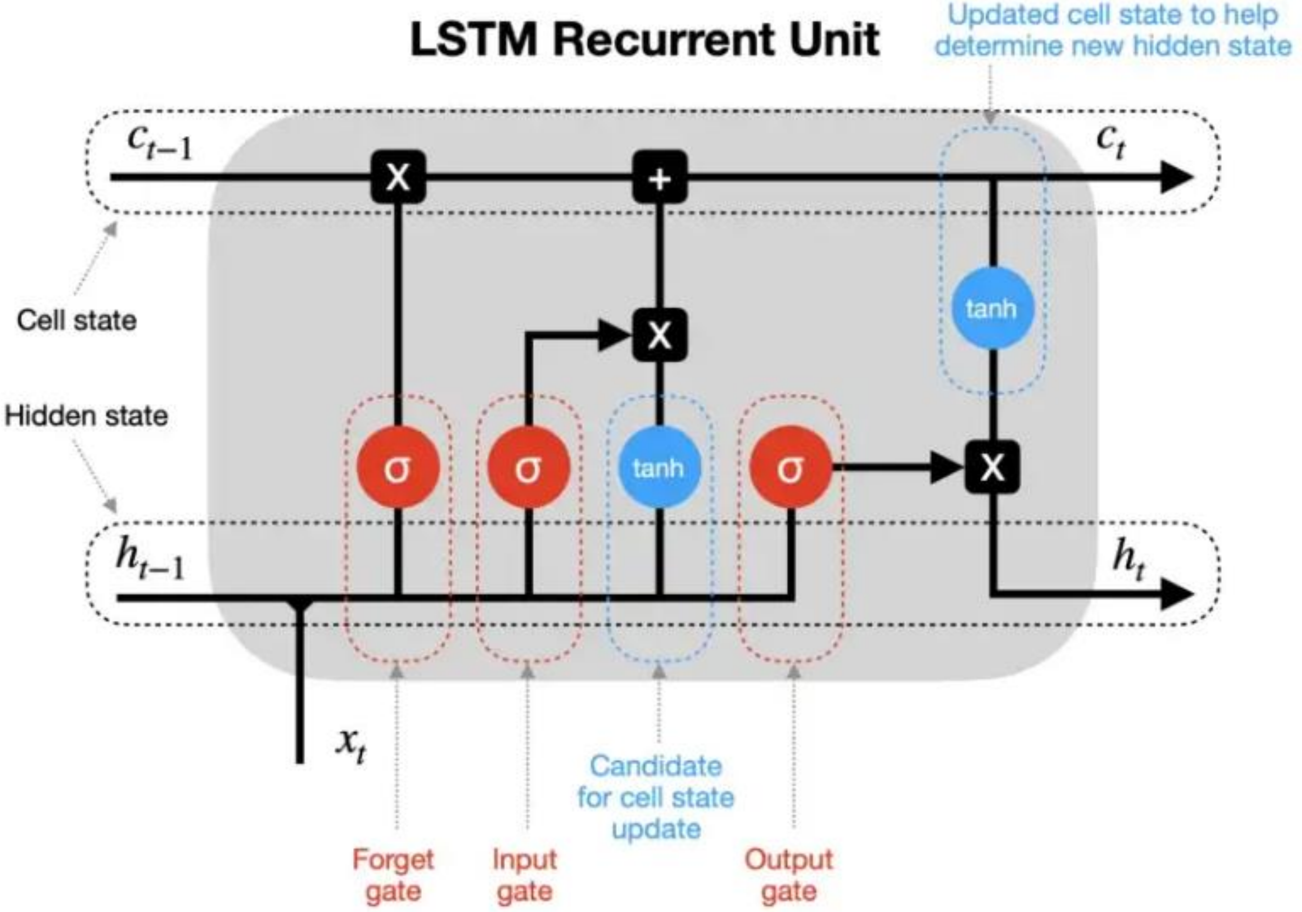
embedding_input InputLayer	input: output:	[(None, 2)] [(None, 2)]
embedding Embedding	input: output:	(None, 2) (None, 2, 100)
lstm LSTM	input: output:	(None, 2, 100) (None, 2, 2000)
lstm_1 LSTM	input: output:	(None, 2, 2000) (None, 2000)
dense Dense	input: output:	(None, 2000) (None, 2000)
dense_1 Dense	input: output:	(None, 2000) (None, 11246)

LSTM

- Long Short-Term Memory networks are a special kind of Recurrent Neural Networks. They are introduced to deal with the long- term dependency problems.
- "The sun rises in the East" . “I grew up in Germany… I speak fluent *German*."

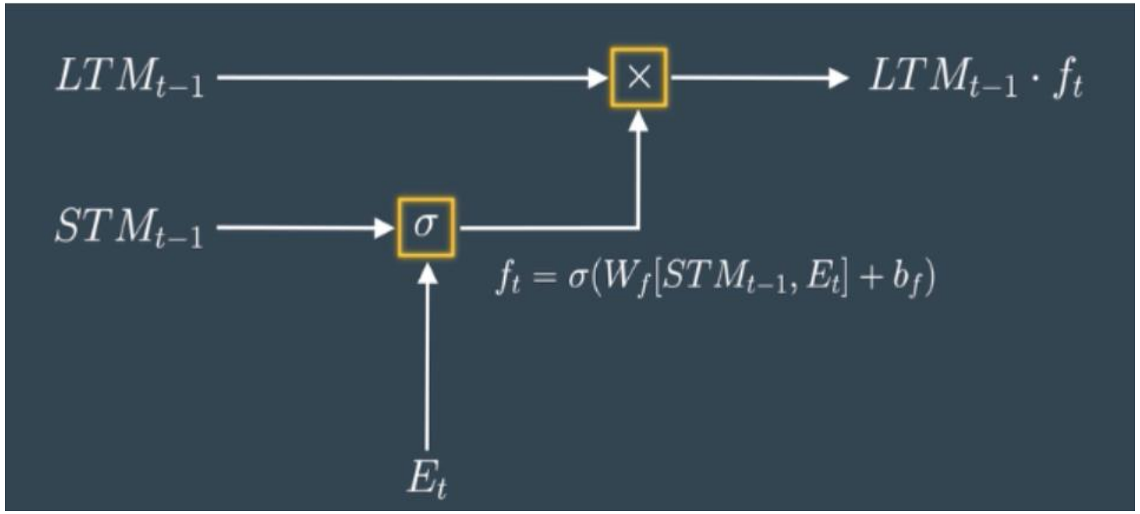
LSTM Architecture:

- LSTM deals with both short-term memory and long-term memory and it uses the concepts of gate.
- There are 4 gates in LSTM, they are: Forget Gate ,Learn Gate , Remember Gate and Use Gate.



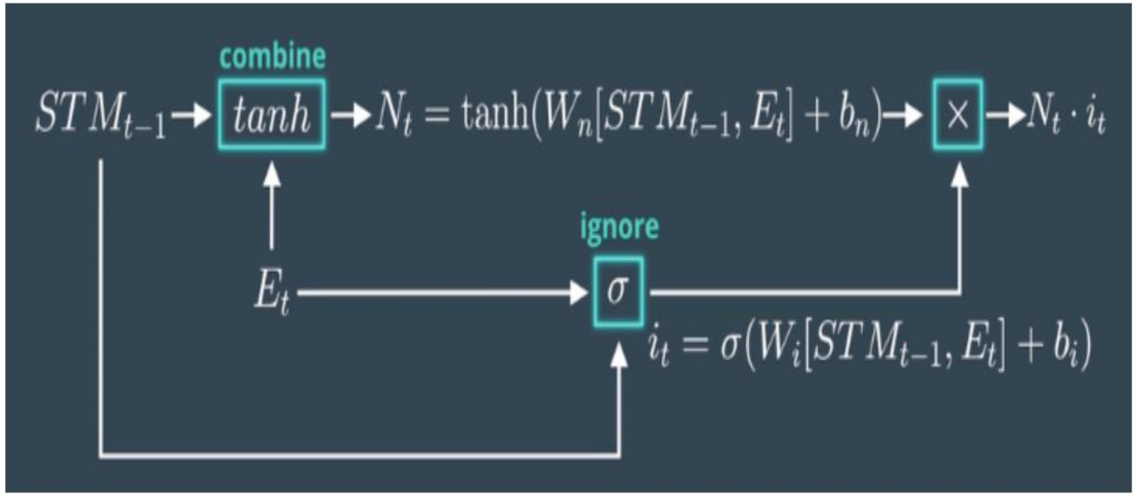
Forget Gate:

- The inputs for the Forget gate are previous LTM and output from the forget layer.
- The previous STM and current event is passed through a sigmoid function (Forget Layer) which gives an output between 0 and 1 where 1 means remember everything and 0 means forget everything.



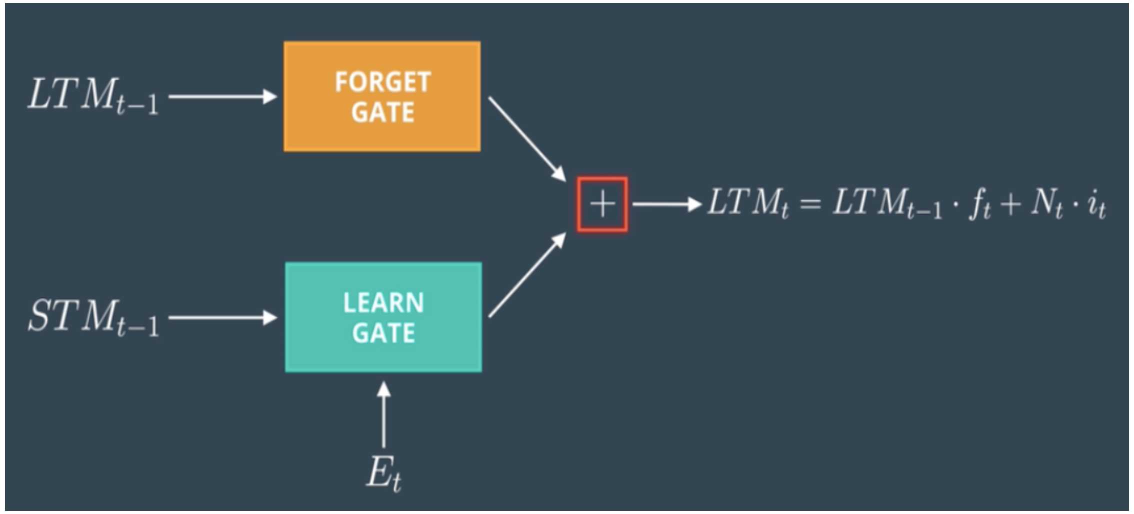
Learn Gate:

- Takes input from previous STM and current event and stores only information required for prediction.
- The previous STM and current event is passed into a tanh function to include non-linearity and combined with the output from the ignorance layer to get the output of learn gate.


combine
STM_{t-1}
$tanh$
$N_t = \tanh(W_n[STM_{t-1}, E_t] + b_n)$
$\times$
$N_t \cdot i_t$
ignore
E_t
σ
$i_t = \sigma(W_i[STM_{t-1}, E_t] + b_i)$

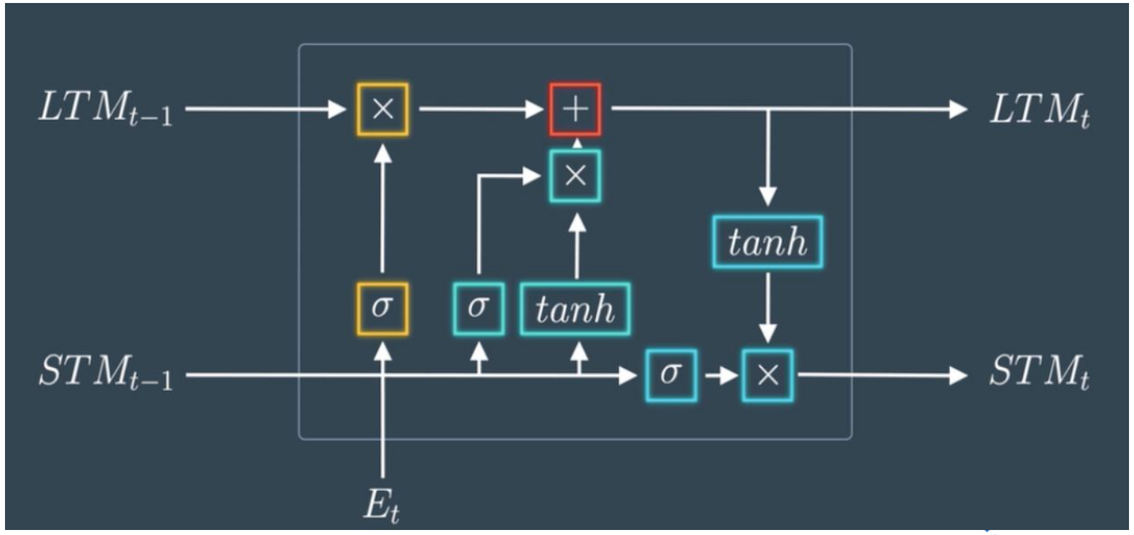
Remember Gate:

- The inputs for the remember gate are the outputs from learn gate and forget gate.
- These two inputs are combined to produce the LTM for the next cell.



Use Gate:

- The inputs for the use gate are the output of remember gate passed through a tanh function and output from the sigmoid function which uses previous STM and Current Event.
- These two inputs are multiplied and the output is passed as a STM for the next cell.



Training and Loss

- Loss function : “categorical_crossentropy”
- Optimiser: “Adam”
- Model is trained for 10 epochs with batch size as 256.
- Loss on the tenth epoch is 3.4168.

Results

Input: I am

Output: Pleased

Input: The international electronic

Output: Measurement

Input: Operating profit rose to

Output: Eur

Input: It’s board of

Output: Directors

Input: Sales for both department

Output: Store

Conclusion

- Sentence Auto completion model is built by using LSTM neural networks trained on the bi-gram data.
- Model performance can be improved further by tuning hyperparameters such as number of layers of LSTM, adding or removing Dense/LSTM layers, number of epochs, loss function, batch size, and covering more scenarios in training data, etc.
- The project can be further expanded by predicting the word from alphabets using dynamic programming and probabilistic models.

Code Link: https://github.com/sai-tej31/AutoCompletion

Sources

- https://www.analyticsvidhya.com/blog/2021/01/understanding-architecture-of-lstm/
- https://towardsdatascience.com/lstm-recurrent-neural-networks-how-to-teach-a-network-to-remember-the-past-55e54c2ff22e

THANK YOU!