LSTM Deep Neural Network for Word Prediction

1st Jicheng Lu
Computer Science and Engineering
Texas A&M University
College Station, TX, 77843
iceljc@tamu.edu

***Abstract*—One of the most challenging tasks in Natural Language Processing is future words prediction, since the computer cannot understand natural language in the same way human does. In this work, we implement sequence-to-sequence recurrent neural networks based on Long Short-Term Memory (LSTM). We aim to accomplish two tasks: language modeling and future words prediction. Due to the nature of each task, we use stacked unidirectional LSTM and bidirectional LSTM neural networks, respectively. Experiments on the two word prediction tasks show that these models can achieve a high prediction performance. On the PTB language modeling task, our model reaches a competitive accuracy of 22.55%. On the BBC news word prediction task, our model can reach an F1 score of 0.141, which is the best so far. We also present the model performance by giving real world texts. The results show that both tasks are accomplished within the AI course project.**

***Index Terms*—NLP, Neural Network, LSTM, Word Prediction**

I. Introduction

Natural Language Processing (NLP) is a sub-field of Artificial Intelligence that is focused on enabling computers to understand and process human languages. While human can easily understand a language, the ambiguity and imprecise characteristics of the natural languages make NLP difficult for computers to implement. For example, consider the following sentence: ”Steph Curry is on fire.” For human, it is easy to know that a basketball player plays very well in a game. However, a computer may understand the sentence Steph Curry that may be a person or place is actually lit on fire. Thus, it is a great challenge to develop systems that read and understand text in the same way human does.

Recurrent neural networks (RNN) have been firmly established as state of the art approaches in sequence modeling and transduction problems, such as language modeling and machine translation [1]–[3]. Numerous efforts have continued to develop the recurrent language models and encoder-decoder architectures [4]–[6]. Among the RNN techniques, Long Short-Term Memory (LSTM) has been widely implemented in the many NLP problems, such as relation classification [7], speech tagging [8], question-answering [9]. The LSTM cell is able to detect long-term dependencies in the data and requires less time to train. LSTMs implement recurrence with four fully-connected layers interacting in a specific way [10]. Fig 1 presents a typical LSTM cell, where h_t and c_t stand for short-term state and long-term state, respectively. As we can see from Fig 1, there are three gates in the cell: forget gate, input gate and output gate. The forget gate is to erase part of the long-term state, input gate is to add part of g_t to the long-term state, and the output gate is to read part of long-term state and output to h_t and y_t. In short, the LSTM cell can learn to recognize an important input, store it in the long-term state, learn to preserve it as long as it is needed, and learn to extract it whenever it is needed.

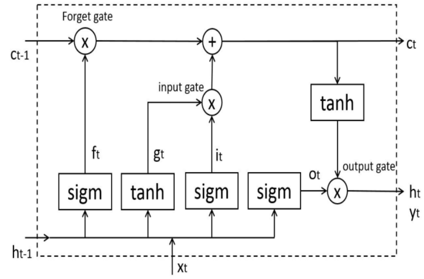


Fig. 1. An LSTM cell.

In this work, we implement and train LSTM-based deep neural networks to accomplish two word prediction tasks: language modeling and future words prediction. Section II states the definition of two tasks. Section III introduce the architecture of our models, including the details of encoder and decoder, word representation, training data, optimizer, loss function and hyper-parameter setting. We wrap up the work by presenting prediction examples from the models in Section IV

II. Task Statement

This section states the two tasks that we aim to accomplish by implementing the neural networks.

A. Predict Next One Word - Language Modeling

The first task is to predict next one word, given a sequence of length m. For example, if we have a sentence: ”I read a”, we aim to predict the next word: ’book’. Mathematically, the model is trained as a probabilistic classifier that learn to predict a probability distribution $P(w_t|w_{t-1}, ..., w_{t-m})$, where w_i represents the i-th word.

B. Predict Future Words

Our second task is to predict the words in the future. Given the first half of a text, we aim to predict the meaningful words in the second half. Here, the meaningful words excludes stop words (e.g. is, are) and words with high frequency in different documents. Therefore, by doing this, we are imitating human thinking process. Given some text that we are reading (e.g. algorithm), we may think about 'complexity', 'programming' and 'memory' that may appear in the text we are about to read.

III. Methodology

In this section, we introduce the neural networks and their details. We first go over the model architectures for the two tasks and then introduce the word representation for the model. After that, we describe the training regime for our models. We introduce the training data at first. Then we look into some details for the optimizer and loss functions we used during training. Finally, we give the setting for hyper-parameters of the models.

A. Model Architecture

The models for the two tasks follow the typical encoder-decoder structure, which has been widely used in many sequence-to-sequence models. Here, the encoder maps an input sequence of symbol representations $(x_1, ..., x_n)$ to a sequence of continuous representations z=$(z_1, ..., z_n)$. Then, given z the decoder produces an output sequence $(y_1, ..., y_m)$, which can be converted to what human understands, such as the English word. Note that there exists difference in the models for each task. For language modeling task, we use a stacked unidirectional LSTM neural network, while we implement a bidirectional LSTM neural network for the future words prediction task. There is also some difference in the decoders.

1) Unidirectional LSTM Neural Network: The structure of the unidirectional LSTM neural network in shown in Fig 2.

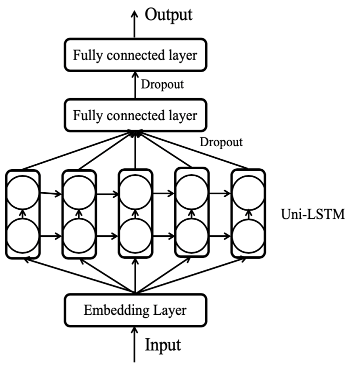


Fig. 2. The architecture of Unidirectional LSTM Model.

- Encoder: The encoder is composed of an embedding layer and a stacked LSTM layer. The embedding layer can be seen as a lookup table that stores embeddings. Its details will be introduced later. The unidirectional LSTM layer is stacked on the embedding layer. The unidirectional LSTM preserves information of the past, since the only inputs it has seen are from the past. This is imitating the human reading behavior: we read from left to right and want to know what the next word is, which is the objective of language modeling. Note that we apply dropouts for the output of each LSTM sublayer.
- Decoder: The decoder is composed of two fully-connected layers. The first one receives the output of LSTM layer while the second layer is used to output the predicted words. We believe that the combination of two linear layers here is deep enough for the current task. We also apply dropout between the two layers.

2) Bidirectional LSTM Neural Network: The structure of the bidirectional LSTM neural network in shown in Fig 3.

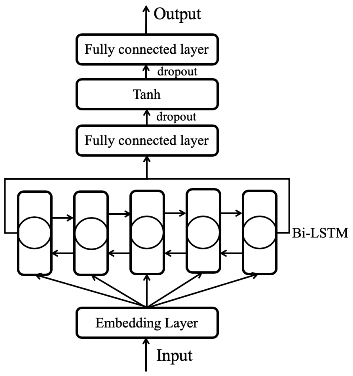


Fig. 3. The architecture of Bidirectional LSTM Model.

- Encoder: The encoder is composed of an embedding layer and a bi-LSTM layer. We first embed the source words so that our model can exploit the fact that certain words (e.g. cat and dog) are semantically similar, and can be processed in a similar way. Then we add a bi-LSTM cell with single sublayer. The bidirectional LSTM run the inputs in two ways, one from past to future and the other from future to past. By running both forwards and backwards, we can preserve information from past and future and let the model understand the text better. We do this by concatenating both forward and backward hidden states. The reason why we use single sublayer for LSTM here is that more sublayers do not improve the prediction accuracy, and we can also have fewer model parameters and speed up the training process.
- Decoder: The decoder is composed of two fully-connected layers and one Tanh layer. We also apply dropouts between each two layers. The output of decoder is used to show the predicted words.

B. Word Representation

Word representation is what we use to numerically represent words such that they can be processed by computers. Word embedding is one of the most popular representation of vocabulary. It is capable of capturing context of a word in a document, semantic and syntactic similarity, relation with other words, etc. Basically, word embeddings are vector representations of a word. It is a good word embedding if the words with similar meanings (e.g. king and queen) can be clustered in the high dimensional space.

In our model, we use pre-trained word embedding: GloVe (Global Vectors for Word Representation) [11]. It consists of 40,000 words, each of which is represented as a 300-dimensional vector. Words are mapped into a meaningful space where distance between words is related to semantic similarity. For example, 'woman' and 'queen' are close to each other, since they both stand for same gender. Moreover, the distance between 'woman' and 'man' is similar and nearly parallel to the one between 'king' and 'queen', since both represent from female to male. More examples are shown in Fig 4.

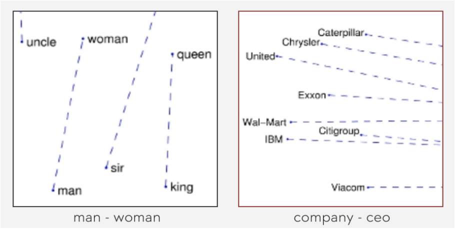


Fig. 4. Examples of word representation in GloVe.

Therefore, because of robust nature of the GloVe representation, we use the word vectors as weights in the embedding layer and they are fixed during training. For the words that are not in the GloVe dictionary, we randomly sample the word vectors with normal distribution of mean 0 and standard deviation 0.1.

C. Training

Next, we describes the training process for our models, including training data, optimizer, loss function and model hyper-parameters.

1) Training Data: We train our models on different datasets respectively. For the language modeling task, we use the Penn Tree Bank (PTB) dataset. It is a popular benchmark for measuring the quality of the language model, while being small and relatively fast to train. Besides the real words, it also contains two special symbols, $< eos >$ and $< unk >$, which represent "end-of-sequence" and non-english words, respectively.

For the future word prediction, we train the model on BBC news dataset. It consists of 2,225 documents corresponding to stories in five topics, namely, business, entertainment, politics, sports and technology. The dataset contains plenty of texts while being limited in five topics. This is beneficial for the training, because the meaningful words are not too diverse so that the model can learn with certain accuracy. Note that we keep all types of words in the source while eliminating the stop words and high-doc-frequency words in the target.

2) Optimizer: We use Adam optimizer in both two models with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. Note that we keep the learning rate the same during training.

3) Loss Function: We use different loss functions for the two tasks. For the language modeling task, we use the cross entropy loss since this is a classification problem with C classes.

$$loss[x, class] = -log(\frac{exp(x[class])}{\sum_j exp(x[j])}) \tag{1}$$

For the future words prediction task, we use the multi-label soft margin loss, which optimizes a multi-label one-versus-all loss based on max-entropy between input x and target y of size (N, C).

$$\begin{aligned} loss[x, y] = -\frac{1}{C} * \sum_i y[i] * log((1 + exp(-x[i]))^{-1}) \\ + (1 - y[i]) * log(\frac{exp(-x[i])}{1 + exp(-x[i])}) \end{aligned} \tag{2}$$

where $i = 0, ..., size(x) - 1$ and $y[i]$ is either 0 or 1.

4) Hyper-parameter Setting: For the language modeling task, we give 1,500 hidden nodes in the LSTM cell, which is good enough for the model, and assign the batch size and sequence length to be 20 and 35, respectively. Other hyper-parameters' setting is given in Table I.

TABLE I
HYPERPARAMETERS IN LANGUAGE MODEL.

Name	Values
embedding dimension	300
hidden nodes	1500
sequence length	35
number of LSTM layers	2
batch size	20
dropout probability	0.65
learning rate	1×10^{-4}

For the future words prediction task, we give 1,024 hidden nodes in the LSTM cell, and assign the batch size to be 4. The number of LSTM sublayers is 1 as discussed in model architecture. Other hyper-parameters' setting is given in Table II.

TABLE II
HYPERPARAMETERS IN WORD PREDICTION MODEL.

Name	Value
embedding dimension	300
hidden nodes	1024
number of LSTM layers	1
dropout probability	0.5
learning rate	3×10^{-5}
batch size	4

IV. Results

The section shows the experiments we have implemented as well as their results and analysis.

A. Predicting Next One Word - Language Modeling

For the language modeling task, we first describe the way we obtain inputs of the network and the targets. Since this is a sequence-to-sequence model, so we need to ensure that both input and output should be a sentence. In each training step, we extract the input from the batched data. The input data matrix has size with "$batch\_size \times seq\_len$", where "$seq\_len$" indicates the length of sequence. Recall that the goal of this task is to predict the next one word given the sequence. Thus, we extract the target from the batched data in a similar way. The difference lies in that we shift the matrix to the right by one word. For example, the entire sentence is 'I read a book today', so the input should be 'I read a book' and the target is 'read a book today'. Then the model should be able to predict the last word 'today' regardless of what other words it predicts. Fig 5 briefly illustrates this procedure. Moreover, there are 10,000 different words in the PTB dataset. More specifically, there are 929,000 words in the training set, 73,000 words in the validation set, and 82,000 words in the test set.

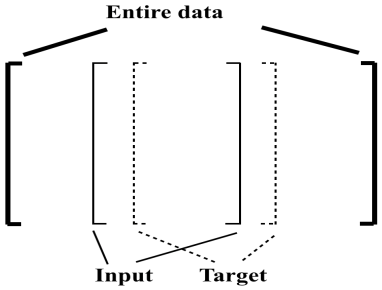


Fig. 5. Illustration of obtaining input and target data.

Next, we present the training and testing results. The evolution of loss and accuracy with epochs are presented in Fig 6 and Fig 7, respectively. The test loss is 4.86 after 100 epochs. The test accuracy reaches 22.55% after 100 epochs. Note that the model becomes overfitting after 30 epochs.

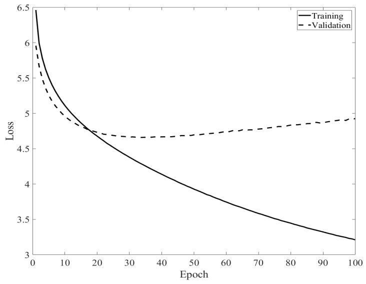


Fig. 6. Training and validation loss.

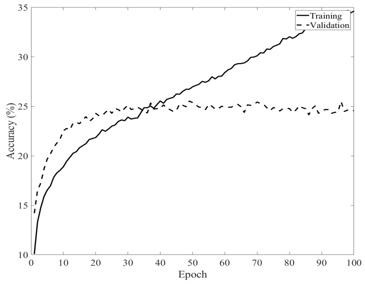
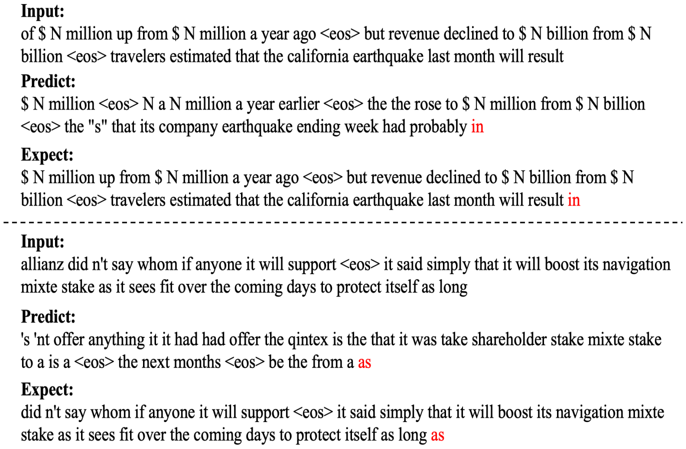


Fig. 7. Training and validation accuracy.

Fig 8 lists some examples about the language model predictions, where the words in red means correct prediction.



Input:
of $ N million up from $ N million a year ago <eos> but revenue declined to $ N billion from $ N billion <eos> travelers estimated that the california earthquake last month will result

Predict:
$ N million <eos> N a N million a year earlier <eos> the the rose to $ N million from $ N billion <eos> the "s" that its company earthquake ending week had probably in

Expect:
$ N million up from $ N million a year ago <eos> but revenue declined to $ N billion from $ N billion <eos> travelers estimated that the california earthquake last month will result in

Input:
allianz did n't say whom if anyone it will support <eos> it said simply that it will boost its navigation mixte stake as it sees fit over the coming days to protect itself as long

Predict:
's 'nt offer anything it it had had offer the qintex is the that it was take shareholder stake mixte stake to a is a <eos> the next months <eos> be the from a as

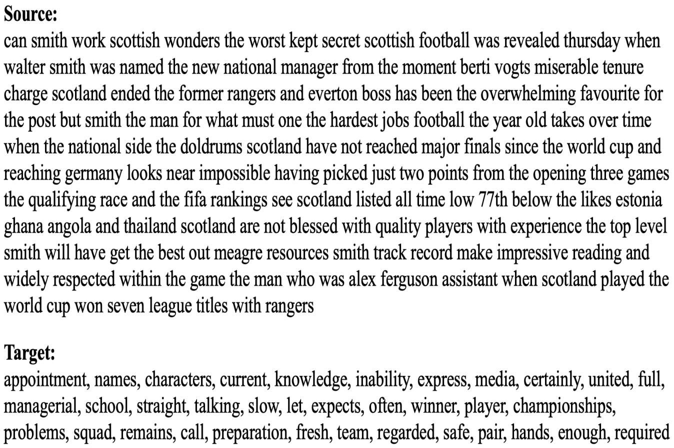
Expect:
did n't say whom if anyone it will support <eos> it said simply that it will boost its navigation mixte stake as it sees fit over the coming days to protect itself as long as

Fig. 8. Examples of language model prediction.

B. Future Words Prediction

For the future words prediction, we first describes the batching method we apply on the training data. In order to explain the data batching more concretely, we present a training example including source sentences and target words, shown in Fig 9. As we can see, the training source is a group of sequences while the training target is a group of separate words. Thus, in each batch of data, there are multiple texts in the source and their corresponding words in the target, where the source sequences are sorted by length. For those shorter sequences, we add paddings after them and make them equal length in each batch. Moreover, we assign 1558 texts for training, 334 texts for validation, and 333 texts for testing.

Since this is a multi-label classification problem, we use precision, recall and F1 score to evaluate the results. The evolution of the loss as well as these three scores with epoch are presented in Fig 10, Fig 11, Fig 12 and Fig 13, respectively. As we can see, the training loss is converging with the training epoch going further while the validation loss is going up after a few epochs. The precision, recall and F1 score are increasing with the epochs, although there is



Source:
can smith work scottish wonders the worst kept secret scottish football was revealed thursday when walter smith was named the new national manager from the moment berti vogts miserable tenure charge scotland ended the former rangers and everton boss has been the overwhelming favourite for the post but smith the man for what must one the hardest jobs football the year old takes over time when the national side the doldrums scotland have not reached major finals since the world cup and reaching germany looks near impossible having picked just two points from the opening three games the qualifying race and the fifa rankings see scotland listed all time low 77th below the likes estonia ghana angola and thailand scotland are not blessed with quality players with experience the top level smith will have get the best out meagre resources smith track record make impressive reading and widely respected within the game the man who was alex ferguson assistant when scotland played the world cup won seven league titles with rangers

Target:
appointment, names, characters, current, knowledge, inability, express, media, certainly, united, full, managerial, school, straight, talking, slow, let, expects, often, winner, player, championships, problems, squad, remains, call, preparation, fresh, team, regarded, safe, pair, hands, enough, required

Fig. 9. A training example of future words prediction.

a fluctuation in recall at the beginning of the training. The test results are presented in Table III. Note that the F1 score reaches 0.14, which is an acceptable result.

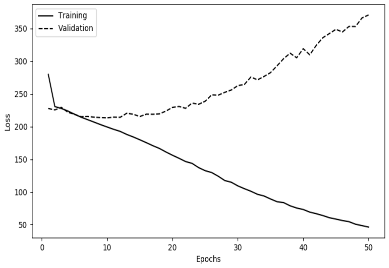


Fig. 10. Training and validation loss.

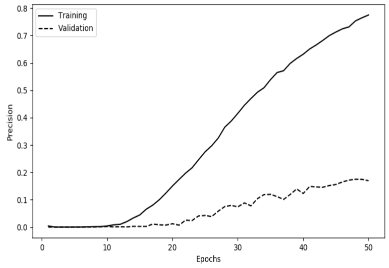


Fig. 11. Training and validation precision.

TABLE III
TEST RESULTS.

Score	Value
Precision	0.546
Recall	0.081
F1	0.141

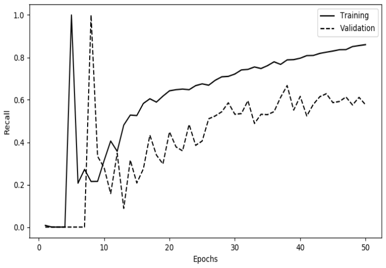


Fig. 12. Training and validation recall.

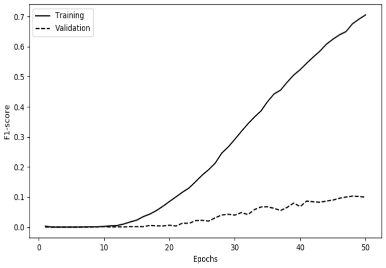
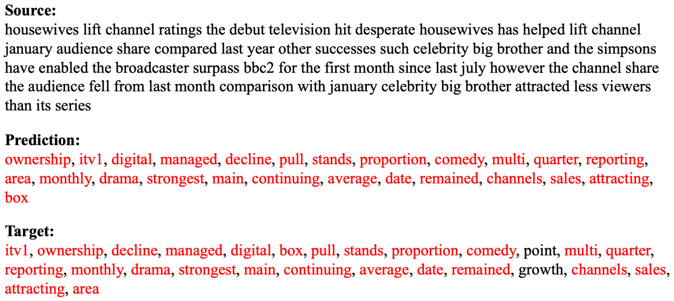


Fig. 13. Training and validation F1 score.

Finally, we present the model performance through several prediction examples, shown in Fig 14, Fig 15 and Fig 16. Note that the words are the correct predictions. We can see that from the given examples the model can predict most of the target words correctly. However, the model may behave differently for other texts, either predicting nothing or predict words but not in the target.



Source:
housewives lift channel ratings the debut television hit desperate housewives has helped lift channel january audience share compared last year other successes such celebrity big brother and the simpsons have enabled the broadcaster surpass bbc2 for the first month since last july however the channel share the audience fell from last month comparison with january celebrity big brother attracted less viewers than its series

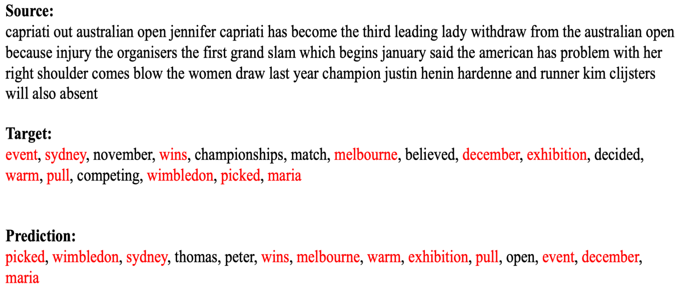
Prediction:
ownership, itv1, digital, managed, decline, pull, stands, proportion, comedy, multi, quarter, reporting, area, monthly, drama, strongest, main, continuing, average, date, remained, channels, sales, attracting, box

Target:
itv1, ownership, decline, managed, digital, box, pull, stands, proportion, comedy, point, multi, quarter, reporting, monthly, drama, strongest, main, continuing, average, date, remained, growth, channels, sales, attracting, area

Fig. 14. Example I of future words prediction.

V. Conclusion

In this work, we accomplish two NLP tasks: language modeling and future words prediction. The language modeling task is to predict the next one word given previous words while the future words prediction task is to predict the meaningful words in the near future given part of the text. Based on the characteristics of each task, we use two different LSTM-based neural networks. For the language modeling, a stacked unidirectional LSTM model is implemented since it can preserve the information from the past. For the future

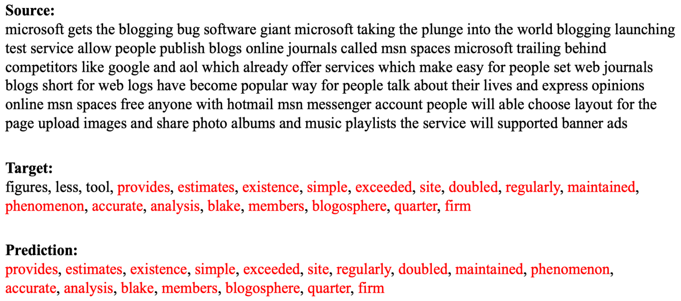


Source:
capriati out australian open jennifer capriati has become the third leading lady withdraw from the australian open because injury the organisers the first grand slam which begins january said the american has problem with her right shoulder comes blow the women draw last year champion justin henin hardenne and runner kim clijsters will also absent

Target:
event, sydney, november, wins, championships, match, melbourne, believed, december, exhibition, decided, warm, pull, competing, wimbledon, picked, maria

Prediction:
picked, wimbledon, sydney, thomas, peter, wins, melbourne, warm, exhibition, pull, open, event, december, maria

Fig. 15. Example II of future words prediction.



Source:
microsoft gets the blogging bug software giant microsoft taking the plunge into the world blogging launching test service allow people publish blogs online journals called msn spaces microsoft trailing behind competitors like google and aol which already offer services which make easy for people set web journals blogs short for web logs have become popular way for people talk about their lives and express opinions online msn spaces free anyone with hotmail msn messenger account people will able choose layout for the page upload images and share photo albums and music playlists the service will supported banner ads

Target:
figures, less, tool, provides, estimates, existence, simple, exceeded, site, doubled, regularly, maintained, phenomenon, accurate, analysis, blake, members, blogosphere, quarter, firm

Prediction:
provides, estimates, existence, simple, exceeded, site, regularly, doubled, maintained, phenomenon, accurate, analysis, blake, members, blogosphere, quarter, firm

Fig. 16. Example III of future words prediction.

words prediction, we use a bidirectional LSTM model because of its strong remembering capability from past and future. We then train the two models on PTB and BBC news dataset, respectively. The results show that the language model can reach an impressive accuracy of 22.55%. Besides, the future words prediction model can achieve an F1 score of 0.141, which is the best performance so far.

In the future, we plan to add self-attention mechanism to the model and perform training on more data.

References

[1] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473. 2014 Sep 1.

[2] Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078. 2014 Jun 3.

[3] Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. InAdvances in neural information processing systems 2014 (pp. 3104-3112).

[4] Jozefowicz R, Vinyals O, Schuster M, Shazeer N, Wu Y. Exploring the limits of language modeling. arXiv preprint arXiv:1602.02410. 2016 Feb 7.

[5] Luong MT, Pham H, Manning CD. Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025. 2015 Aug 17.

[6] Wu Y, Schuster M, Chen Z, Le QV, Norouzi M, Macherey W, Krikun M, Cao Y, Gao Q, Macherey K, Klingner J. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144. 2016 Sep 26.

[7] Zhou P, Shi W, Tian J, Qi Z, Li B, Hao H, Xu B. Attention-based bidirectional long short-term memory networks for relation classification. InProceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) 2016 (Vol. 2, pp. 207-212).

[8] Plank B, Søgaard A, Goldberg Y. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. arXiv preprint arXiv:1604.05529. 2016 Apr 19.

[9] Wang D, Nyberg E. A long short-term memory model for answer sentence selection in question answering. InProceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers) 2015 (Vol. 2, pp. 707-712).

[10] Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997 Nov 15;9(8):1735-80.

[11] Pennington J, Socher R, Manning C. Glove: Global vectors for word representation. InProceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) 2014 (pp. 1532-1543).