A deep learning approach for information extraction in real estate advertisements

Trung Trinh
Faculty of Computer Science and Engineering
Bach Khoa University
Ho Chi Minh City, Vietnam
Email: 1414316@hcmut.edu.vn

Dang Ngo
Faculty of Computer Science and Engineering
Bach Khoa University
Ho Chi Minh City, Vietnam
Email: 1410859@hcmut.edu.vn

Hon Pham
Faculty of Computer Science and Engineering
Bach Khoa University
Ho Chi Minh City, Vietnam
Email: 1411394@hcmut.edu.vn

Tho Quan
Faculty of Computer Science and Engineering
Bach Khoa University
Ho Chi Minh City, Vietnam
Email: qttho@hcmut.edu.vn

Gia-Long Hoang-Ngoc
Atomic Vietnam Co., LTD
Ho Chi Minh City, Vietnam
Email: long.hoang@zeniius.com

Hung Hoang
Atomic Vietnam Co., LTD
Ho Chi Minh City, Vietnam
Email: hung.hoang@zeniius.com

***Abstract*—Real estate has become a hot market in the modern society today. Usually, people search for a suitable living place through various channels such as the Internet and realtors. There are many websites online dedicated to trading real estate but most of them require users to specify their desired criteria on a predefined form. However, natural languages are preferred by common people when advertising or searching a real estate property.**

Nevertheless, the real estate search requests and advertisements in practical situations are mostly unstructured and naturally ambiguous, which pose a real challenge for automatic processing. In this paper, we consider this problem as a sequence labelling task and apply a deep learning model to extract features of a property (i.e. address, type of property, etc.). We adopt the current state-of-the-art deep learning architecture, which includes word-level and character-level Convolutional Neural Networks, bidirectional Gated Recurrent Unit, Conditional Random Field and word embedding. The advantage of our approach has been proven when experimented with real data collected from real estate trading sites.

***Keywords*-sequence labelling; deep learning; information extraction; real estate; Recurrent Neural Network; Gated Recurrent Unit; Conditional Random Field; Convolutional Neural Network**

I. INTRODUCTION

Vietnam's real estate market has been thriving in recent years. More and more websites dedicated to trading real estate are established and the number of real estate advertisements on those websites is growing rapidly. People visit those websites to search for a property that meets their personal needs. However, the searching systems on those websites are fairly limited, allowing users to only select from a set of predefined options, which may not include the user's criteria. Therefore, a better searching mechanism will be useful to many people and greatly improve user experience.

The first step to building such system is to develop an algorithm that can automatically process real estate advertisements. This algorithm should be able to analyze adverts for relevant phrases that capture the characteristics of the property such as price, address, architecture and surrounding area, etc. that could be of interest to other people.

We defined this problem as the task of *sequence labelling*, where each element of an input sequence is matched with a label. Well-known examples include speech and handwriting recognition, part-of-speech tagging, named-entity recognition and event detection.

For instance, let us consider the following Vietnamese input.

Bán đất tiện xây phòng trọ, cho thuê, gần nhà máy sữa Vinamilk (Land for sale, convenient for *inn business, house renting, close to the Vinamil factory).*

Consider this input at a sequence of Vietnamese, this sequence will be matched with a sequence of labels as follows.

('Bán (sale)', 'TRANSACTION_TYPE'), ('đất (land)', 'REAL_ESTATE_TYPE'), ('tiện (convenient)', 'O'), ('xây phòng trọ (inn business)', 'POTENTIAL'), ('cho thuê (rent)', 'POTENTIAL'), ('gần (close)', 'O'), ('nhà máy sữa (milk factory)', 'SURROUNDING_PLACE'), ('Vinamilk', 'SURROUNDING_NAME')

where TRANSACTION_TYPE, REAL_ESTATE_TYPE, O, POTENTIAL, SURROUNDING_PLACE, SURROUNDING_NAME are the labels bearing the corresponding semantics.

Here what we aim to recognize is all the criteria of a property that people usually consider when purchasing a real estate. This is a challenging problem because: (1) these criteria are very diverse; (2) they are expressed in many different ways in natural language; and (3) spelling mistakes are frequently made in these advertisements. Previous works on linguistic sequence labelling employed statistical model such as *Conditional Random Field* [1], *Hidden Markov Model* [2] and *Maximum Entropy Model* [3] with carefully designed feature engineering that are domain-and-language-specific. Such features are costly to develop and thus these result cannot be easily adapted to new problems. Recent development in the field has yielded state-of-the-art result with neural network models, such as [4], [5], [6]. All of those works combined Bidirectional Long Short-Term Memory (LSTM) with Conditional Random Field (CRF), with some additions such as character-level word embedding generated either by a Convolutional Neural Network (CNN) or a Bidirectional LSTM. These models typically operated on word embedding trained through the use of unsupervised methods such as GloVe [7] or Word2Vec [8], therefore requires only minor or no feature engineering at all.

In this paper, we present the way that we improve upon the work in [6], adding components to make our model more suitable to the characteristics of the Vietnamese language. Similar to that paper, we first apply a simple 1D CNN-Max Pooling layer to generate the character-level representation of a word. Then we combine the pre-trained word embedding with its character-level representation and feed them into another CNN, this time on word level, to capture the features of n-grams. These features are the input to a stack of Bidirectional Gated Recurrent Unit (GRU) layer. The output layer is a Conditional Random Field to decode the sequence of labels with the highest score given the input sequence. We evaluate our model on our dataset of real estate advertisements and demonstrate that the added components to the model certainly improved our model performance on the dataset.

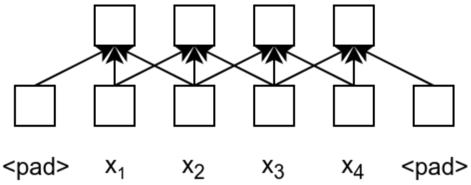


Figure 1: A 1D convolution layer

II. Background

A. Convolution and pooling

Convolution [9] is an operation that is widely used in many deep learning models for various research fields such as computer vision and natural language processing, the latter usually involves a 1-D convolution whose structure is depicted in Figure 1.

Let d be the size of each word vector and s be the length of the sequence. Let $C \in \mathbb{R}^{s \times d}$ be the sequence matrix. A convolution operation applies a kernel $H \in \mathbb{R}^{k \times d}$ to a window of k words to capture the features of *k-grams* in the sequence. Specifically, a feature of an *k-grams* from a window of words $C[i : i + k; \bullet]$ is generated by:

$$c_i = \sigma(\sum(C[i : i + k; \bullet] \odot H) + b)$$

where $b \in \mathbb{R}$ is the bias term, σ is a non-linear activation function such as $tanh$ or $ReLU$ and $\odot$ is the element-wise matrix multiplication. To retain the length of a sequence after convolution is applied, we zero-padded the sequence evenly on both side with a padding size of $(k - 1)/2$. Under such padding scheme, a kernel is convolved with each possible *k-grams* in the sequence to produce a feature map $c = [c_1, c_2, \ldots, c_s]$ with $c \in \mathbb{R}^s$.

Pooling is an operation used to aggregate a group of generated features. A popular function for pooling is to find the maximum value among them. In term of sequence modelling, global pooling is often used, which returns the most distinctive features of

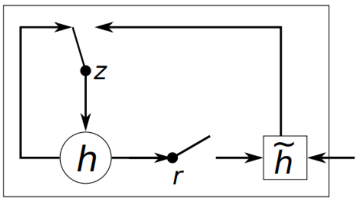


Figure 2: An illustration of a GRU cell [10]

a sequence when coupled with the max pooling function.

B. Gated Recurrent Unit

Gated Recurrent Unit (GRU) was introduced in [10]. It is a variant of Recurrent Neural Networks (RNNs) capable of learning long-term dependencies while avoiding the problem of vanishing or exploding gradients faced by vanilla RNNs. GRU is proved to be comparable to LSTM [11] while maintaining a smaller set of parameters. Figure 2 presents a graphical illustration of GRU.

The parameters are updated using the following equations:

$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z) \tag{1}$$

$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r) \tag{2}$$

$$\widehat{h}_t = \sigma_h(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \tag{3}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widehat{h}_t \tag{4}$$

where σ_g and σ_h are activation functions and $\sigma_g(x) \in [0, 1]$, which usually are a sigmoid function and hyperbolic tangent respectively, $\odot$ denotes the element-wise matrix multiplication, r_t is the reset gate, z_t is the update gate, x_t is the input vector and h_t is the output vector.

C. Conditional Random Fields

Let $x = [x_1, x_2, \ldots, x_T]$ be an input sequence and $y = [y_1, y_2, \cdots, y_T]$ be the sequence of corresponding tags. Let D be the set of possible tags, that is $y_i \in D \ \forall i \in [1, T]$ and Y be the set of all possible label sequences. A linear-chain conditional random fields (CRF) model [1] is defined as

$$s(y) = \sum_{t=1}^{T} A_{x_t,y_t} + \sum_{t=2}^{T} V_{y_{t-1},y_t} \tag{5}$$

$$P(y|x) = \frac{e^{s(y)}}{\sum_{\tilde{y} \in Y} e^{s(\tilde{y})}} \tag{6}$$

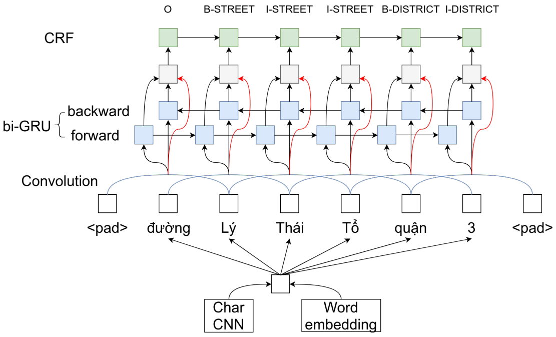


Figure 3: The model architecture where *char CNN* is the CNN network for generating character-level representation; red lines indicate skip connection; blue lines indicate convolution; blue boxes indicate GRU and green boxes indicate CRF.

where A_{x_t,y_t} denotes the score of the t-th word having tag y_t, V_{y_{t-1},y_t} denotes the transition score between the previous tag y_{t-1} and the current tag y_t and $s(y)$ denotes the score of the label sequence y. In this model, prediction can be decoded using the $O(D^2T)$ Viterbi algorithm to find a sequence of tags with maximum score given the input sequence.

CRF models have a property of incorporating interactions between consecutive tags into making predictions. This introduced two major advantages. The first one is the guarantee that certain constraints of a tagging scheme (e.g: IOB tagging) are always upheld. The second one is the added constraint means that the model requires less training data. However, it has worse computational complexity compare to independent prediction, that is making a prediction at each position based only on the features of that position.

III. Model Architecture

Figure 3 gives an overview of our model which includes the following components.

A. Bidirectional GRU with CRF, convolution and skip connection

Follow the current state-of-the-art model for sequence tagging [4], our model's main component is a bidirectional GRU (bi-GRU) network for feature extraction, followed by a linear-chain CRF as the output layer to determine the tag at each position while considering tags of previous positions. We chose to use GRU in place of LSTM because it has been showed that GRU is comparable to LSTM and even better in certain tasks [12] while allowing faster training time and lower sample complexity due to having a smaller number of parameters. To account for the fact that syllables in the Vietnamese language are written separably, hence each element in a sequence is usually a syllable, we use a convolution layer after the word embedding to capture soft n-gram of syllables, using the output of this layer as input to the bi-GRU component. Using this method, our model is able to correctly capture long and polysyllabic words (e.g "trường đại học") more frequently. To alleviate the problems of vanishing gradients when training over long sequences and to allow our model to have more layers, we added skip connection at each bi-GRU layers. Specifically, each input and output of a bi-GRU layer is concatenated to form the final output of that layer. Given an input sequence $x = [x_1, x_2, \ldots, x_T]$, each bi-GRU layer can be described using following equations:

$$h_t^f = \overrightarrow{GRU}(x_1, x_2, \ldots, x_t) \tag{7}$$

$$h_t^b = \overleftarrow{GRU}(x_t, x_{t+1}, \ldots, x_T) \tag{8}$$

$$o_t = [x_t, h_t^f, h_t^b] \tag{9}$$

For training, the loss function maximizes the probability of getting the correct label sequence from the corresponding input sequence:

$$J = -log(P(y|x)) = -s(y) + \sum_{\tilde{y} \in Y} e^{s(\tilde{y})} \tag{10}$$

where $s(y)$ is defined by (5)

B. Character-level representation

Normally a model for natural language processing can only include a fixed size vocabulary and replace any unseen words with an out-of-vocabulary (OOV) token. This strategy could reduce performance since the model neglects useful information unknown words could provide. One solution to this problem is to incorporate morphological information of each word into the model. By adding this information, important features such as word case is considered when making predictions. We decided to generate such representation of each word by using a simple convolutional neural network depicted in Figure 4.

This representation is then combined with the corresponding word embedding by concatenating them together to create the final vector that contains all the information about the word. By using this method, we indirectly embedded information such as word case and groups of characters into the model.

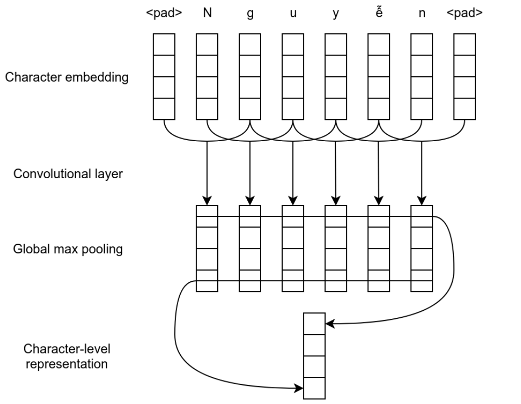


Figure 4: CNN for character-level embedding [6]

IV. Experiment

A. Dataset

Our data comprise 9231 documents collected from *mogi.vn*, a website for trading real estate in Vietnam. Based on the information usually contained in a real estate advertisement, we came up with 19 different labels representing different properties of a real estate. Table I shows information about each label.

For data preprocessing, we replaced all the digits with the character '0'. Any word with less than five occurrences in the dataset is replaced with the generic OOV token. We did not convert the text to a single case because each word case could give useful information (e.g word start with an uppercase letter usually indicate a street name). We first used fastText [13] to train the word embeddings in an unsupervised manner, then using this word embedding as inputs to the model. We decided to use word embeddings with 100 dimensions. We chose to use IOB tagging scheme in our data.

B. Hyperparameter

Table II includes all the hyperparameters of our model.

C. Training method

To optimize our model, we use the Adam Optimizer [14] with a learning rate of 0.001 as recommended. We applied dropout [15] of 0.5 after each layer to avoid overfitting problem. We chose a batch size of 100 and shuffle the dataset on each epoch to avoid our model falling into a local minimum. We divided our dataset into 3 part: 80% for training, 10% for validation and 10% for testing. We stopped

Table I: The labels used in the dataset

Label	Description	Supports
City	The city where the real estate is located.	1542
District	The district where the real estate is located.	5826
Street	The street where the real estate is located	7351
Ward	The ward where the real estate is located	3512
Area	The area of the real estate	13917
Floor	Number of floors	10956
Room	Number of rooms	10185
Legal	Legal information related to the real estate (e.g certificate of home ownership)	5263
Orientation	The direction which the real estate is facing	1424
Position	Whether the real estate is on a street or a lane.	6908
Potential	Potential usage of the real estate (e.g: to open a shop, a company office)	17706
Price	Price of the real estate	10320
Project	the project the real estate belongs to (e.g Sunrise city)	1876
Type	The type of the real estate (e.g land, apartment, villa, house)	20058
Surrounding places	Places near the real estate (e.g school, hospital, restaurant)	31155
Surrounding characteristics	The characteristics of surrounding area (e.g secured, quiet, populous)	11624
Surrounding names	Names of the places near the real estate (e.g Bách Khoa, Bình Dân)	21563
Transaction type	Type of the transaction (e.g buy, for rent, to rent, for sale)	8526
Normal	Any others information that does not belong to the labels above	194892

Table II: Hyperparameters of our model

Word embedding size	100
Character embedding size	30
Character CNN filter size	3
Character CNN number of filters	50
Word CNN filter size	3
Word CNN number of filters	128
GRU hidden size	100
Number of BiGRU layers	2

Table III: Experimental results

	F1	Precision	Recall
BiGRU-CharCNN-CRF	85.6	85.3	85.8
BiGRU-CharCNN-CRF (+ skip)	85.8	85.7	85.8
BiGRU-CharCNN-CRF (+ WordCNN)	86.1	85.4	86.9
BiGRU-CharCNN-CRF (+ WordCNN + skip)	86.3	86.1	86.6

training our model using early stopping, ending the process if the performance of the model on the validation had not improved for 10 consecutive epochs. The final model is the one with the highest performance on the validation set.

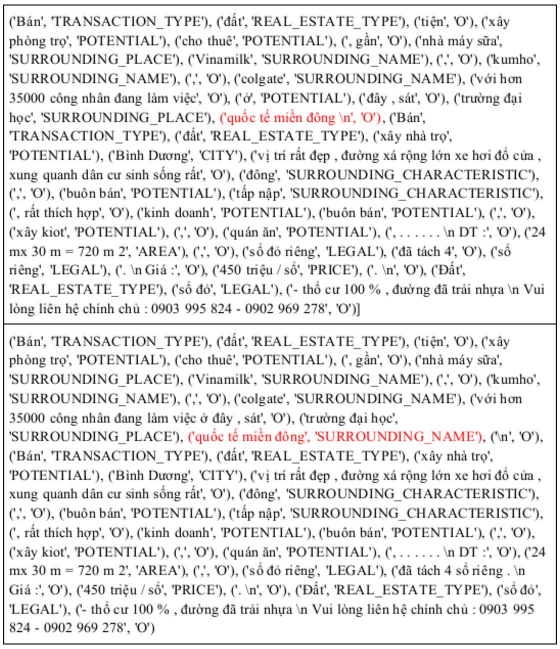
D. Result

Table III consists of results of three models: the original model introduced in [6] (BiGRU-CharCNN-CRF), the model with skip connection (+ skip), the model with word level CNN (+ Word-CNN), and the model with both components.

From the result, we can see that adding skip connection increases the precision of the model, probably due to the fact that shortcut connections allow faster and easier convergence. The WordCNN increases recall significantly because it helps the model to capture long phrases with more than 4 words. Figure 5 compares the result between model before and after adding the WordCNN layer. Finally, adding both the WordCNN and skip connection yields the best result.

V. Conclusion

This paper proposed an adaptation of a current state-of-the-art model for sequence labelling to analyze real estate advertisement in Vietnamese. We introduce two improvements which are adding



```
('Bán', 'TRANSACTION_TYPE'), ('đất', 'REAL_ESTATE_TYPE'), ('tiện', 'O'), ('xây
phòng trọ', 'POTENTIAL'), ('cho thuê', 'POTENTIAL'), (', gần', 'O'), ('nhà máy sữa',
'SURROUNDING_PLACE'), ('Vinamilk', 'SURROUNDING_NAME'), (',', 'O'), ('kumho',
'SURROUNDING_NAME'), (',', 'O'), ('colgate', 'SURROUNDING_NAME'), ('với hơn
35000 công nhân đang làm việc', 'O'), ('ở', 'POTENTIAL'), ('đây , sát', 'O'), ('trường đại
học', 'SURROUNDING_PLACE'), ('quốc tế miền đông \n', 'O'), ('Bán',
'TRANSACTION_TYPE'), ('đất', 'REAL_ESTATE_TYPE'), ('xây nhà trọ',
'POTENTIAL'), ('Bình Dương', 'CITY'), ('vị trí rất đẹp , đường xá rộng lớn xe hơi đổ cửa ,
xung quanh dân cư sinh sống rất', 'O'), ('đông', 'SURROUNDING_CHARACTERISTIC'),
(',', 'O'), ('buôn bán', 'POTENTIAL'), ('tấp nập', 'SURROUNDING_CHARACTERISTIC'),
(', rất thích hợp', 'O'), ('kinh doanh', 'POTENTIAL'), ('buôn bán', 'POTENTIAL'), (',', 'O'),
('xây kiot', 'POTENTIAL'), (',', 'O'), ('quán ăn', 'POTENTIAL'), (', . . . . . . \n DT :', 'O'), ('24
mx 30 m = 720 m 2', 'AREA'), (',', 'O'), ('sổ đỏ riêng', 'LEGAL'), ('đã tách 4', 'O'), ('sổ
riêng', 'LEGAL'), ('. \n Giá :', 'O'), ('450 triệu / sổ', 'PRICE'), ('. \n', 'O'), ('Đất',
'REAL_ESTATE_TYPE'), ('sổ đỏ', 'LEGAL'), ('- thổ cư 100 % , đường đã trải nhựa \n Vui
lòng liên hệ chính chủ : 0903 995 824 - 0902 969 278', 'O')]
```

```
('Bán', 'TRANSACTION_TYPE'), ('đất', 'REAL_ESTATE_TYPE'), ('tiện', 'O'), ('xây
phòng trọ', 'POTENTIAL'), ('cho thuê', 'POTENTIAL'), (', gần', 'O'), ('nhà máy sữa',
'SURROUNDING_PLACE'), ('Vinamilk', 'SURROUNDING_NAME'), (',', 'O'), ('kumho',
'SURROUNDING_NAME'), (',', 'O'), ('colgate', 'SURROUNDING_NAME'), ('với hơn
35000 công nhân đang làm việc ở đây , sát', 'O'), ('trường đại học',
'SURROUNDING_PLACE'), ('quốc tế miền đông', 'SURROUNDING_NAME'), ('\n', 'O'),
('Bán', 'TRANSACTION_TYPE'), ('đất', 'REAL_ESTATE_TYPE'), ('xây nhà trọ',
'POTENTIAL'), ('Bình Dương', 'CITY'), ('vị trí rất đẹp , đường xá rộng lớn xe hơi đổ cửa ,
xung quanh dân cư sinh sống rất', 'O'), ('đông', 'SURROUNDING_CHARACTERISTIC'),
(',', 'O'), ('buôn bán', 'POTENTIAL'), ('tấp nập', 'SURROUNDING_CHARACTERISTIC'),
(', rất thích hợp', 'O'), ('kinh doanh', 'POTENTIAL'), ('buôn bán', 'POTENTIAL'), (',', 'O'),
('xây kiot', 'POTENTIAL'), (',', 'O'), ('quán ăn', 'POTENTIAL'), (', . . . . . . \n DT :', 'O'), ('24
mx 30 m = 720 m 2', 'AREA'), (',', 'O'), ('sổ đỏ riêng', 'LEGAL'), ('đã tách 4 sổ riêng . \n
Giá :', 'O'), ('450 triệu / sổ', 'PRICE'), ('. \n', 'O'), ('Đất', 'REAL_ESTATE_TYPE'), ('sổ đỏ',
'LEGAL'), ('- thổ cư 100 % , đường đã trải nhựa \n Vui lòng liên hệ chính chủ : 0903 995
824 - 0902 969 278', 'O')
```

Figure 5: Result on the model without WordCNN (above) and with WordCNN (below). Notice the long phrase in red is captured by the latter model, but is ignored by the former.

the skip connection between BiGRU layers and word-level CNN to capture n-grams. Our approach yielded significant improvement on the F1 metric when applied to real dataset.

ACKNOWLEDGEMENT

We would like to thank K35 GROUP JSC for funding this research.

REFERENCES

[1] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” in *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001, pp. 282–289.

[2] J. Kupiec, “Robust part-of-speech tagging using a hidden markov model,” *Computer Speech & Language*, vol. 6, no. 3, pp. 225 – 242, 1992.

[3] A. Ratnaparkhi, “A maximum entropy model for part-of-speech tagging,” in *Conference on Empirical Methods in Natural Language Processing*, 1996, pp. 133–142.

[4] Z. Huang, W. Xu, and K. Yu, “Bidirectional LSTM-CRF Models for Sequence Tagging,” *arXiv:1508.01991 [cs.CL]*, Aug. 2015.

[5] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, “Neural architectures for named entity recognition,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 260–270.

[6] X. Ma and E. Hovy, “End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 1064–1074.

[7] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.

[8] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in Neural Information Processing Systems 26*, 2013, pp. 3111–3119.

[9] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *Neural Computation*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[10] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN Encoder–Decoder for statistical machine translation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734.

[11] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[12] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” *arXiv:1412.3555 [cs.NE]*, Dec. 2014.

[13] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching word vectors with subword information,” *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[14] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *arXiv:1412.6980 [cs.LG]*, Dec. 2014.

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.