INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

A Neural Turn-taking Model without RNN

Chaoran Liu[1], *Carlos Ishi*[1], *Hiroshi Ishiguro*[1,2]

[1]ATR Hiroshi Ishiguro Labs, Japan
[2]Graduate School of Engineering Science, Osaka University
chaoran.liu@atr.jp, carlos@atr.jp, ishiguro@sys.es.osaka-u.ac.jp

Abstract

Sequential data such as speech and dialogs are usually modeled by Recurrent Neural Networks (RNN) and derivatives since the information can travel through time with such architecture. However, disadvantages exist with the use of RNNs, including the limited depth of neural networks and the GPU's unfriendly training process.

Estimating the timing of turn-taking is a critical feature of dialog systems. Such tasks require knowledge about past dialog contexts and have been modeled using RNNs in several studies. In this paper, we propose a non-RNN model for the timing estimation of turn-taking in dialogs. The proposed model takes lexical and acoustic features as its input to predict a turn's end. We conducted experiments on four types of Japanese conversation datasets and show that with proper neural network designs, the long-term information in a dialog could propagate without a recurrent structure. The proposed model outperformed canonical RNN-based architectures on a turn-taking estimation task.
Index Terms: Turn-taking, Deep learning, Capsule network, CNN, Dilated ConvNet

1. Introduction

For a spoken dialog system, the ability to take turns at appropriate timing is critical. A commonly used strategy in dialog systems is detecting silences longer than a heuristically designed threshold (e.g., 0.5s). However, such strategies are insufficient for smooth turn-taking [1].

Studies since the 1970s have shown that a speaker's turn-release signals consist of lexical, prosodic, and gestural cues in human conversations [2, 3]. Some works deployed such non-verbal modalities as gestures [4] or gazes [5] to estimate the endpoint of the conversational turns. Hand-crafted expert knowledge has also been used in many researches [6]. Such hard-coded models are difficult to transfer to other languages/cultures since they are culture-dependent. Data-driven methods such as finite state machine-based [7] and neural network-based models [8, 9, 10] have also been proposed in recent years. These works use feature sequences extracted from both text and speech signals. To deal with the time dependency of feature sequences, all of the above neural network-based works use recurrent neural network (RNN) or variants (e.g., long-short term memory [11], gated recurrent unit [12], etc.). Several also adopted nested hierarchical architecture and learnable speaker ID [8] or a speaker dependent feature set [9] to address the long-term dialog context.

To the best of our knowledge, all currently proposed neural network-based turn-taking estimation systems are built on the basis of RNN-related architectures. On the other hand, the state of the art performances regarding speech synthesis [13], machine translation [14], and language modeling [15] have been achieved by convolutional neural networks (CNN) [16]. The successes of CNNs on these sequential modeling tasks suggest that RNN-based architectures are not necessarily optimal solutions for the endpoint estimation for conversational turns.

Inspired by several recent studies, we proposed a RNN-free architecture to perform turn-taking estimation. The reason to avoid using RNNs is twofold. One is that the computation of each RNN state needs to be performed sequentially, parallelizing them is difficult. Training a RNN-related network generally requires much longer time than a CNN with the same number of parameters on an equivalent device. The other is that RNNs cannot be deeply stacked, and many works argued that deep networks are better generalized than shallow ones [17, 18]. With a RNN-free architecture, we can easier explore a different network without access to rich computational resources.

In the proposed model, the lexical information is processed by a capsule network [19] with a convolutional layer, where the acoustic information is handled by a dilated convolutional network with ResNets [17] as its building blocks. We experimentally compared our proposed model with two RNN-based models (nested [8] and stacked [10]) on a Japanese conversational corpus that included four types of conversations: dating, job interviews, attentive listening, and at reception-counter conversations. Our experimental results show that the proposed non-RNN model outperformed RNN-based networks in a turn-taking estimation task.

2. Background

For modeling sequential data that were formed as a time series, RNNs enjoyed a long period of dominance due to their nature inherited from differential equations. In such applications as language modeling and machine translation, RNNs achieved tremendous success especially after the introduction of LSTM and GRU [20, 21], since they overcame a difficulty known as the vanishing gradient with which plain RNN architectures continue to struggle. All RNN variants shared the same concept where the hidden state represents everything that has passed through this network so far, and the computation of the current state depends on these hidden states. This fact prohibits the training process of RNNs from doing parallelized computing. The performance of eight RNN variants has been benchmarked on speech recognition, handwritten English sentence recognition, and music modeling [22]. Our results showed that none of these variants significantly outperformed the original LSTM model proposed in the late 1990s [11].

At the same time, many attempts have also applied CNNs to sequential data. In natural language processing (NLP) tasks, many outstanding results have been reported using CNNs, including part-of-speech tagging [23], sentence classification [24], document classification [25], machine translation [14], and language modeling [15]. In CNN approaches, local information (i.e., word co-occurrence in a local neighborhood in

http://dx.doi.org/10.21437/Interspeech.2019-2270

NLP cases) was aggregated by multiple pattern detectors and passed to a higher layer for higher-level representation. These translated replicas of pattern detectors (i.e., convolutional kernels), which can translate acquired knowledge from one position to another, are extremely helpful for pattern interpretation. However, some argue that a replicating pattern detector is inefficient because it grows exponentially with the number of dimensions [19].

To address this CNN inefficiency, a model named *capsule network* was recently proposed [19]. In it, the scalar output of a pattern detector was replaced with vector output (i.e., a capsule), and the max pooling process was replaced with a routing-by-agreement process that sends the capsule of each output of a lower layer to an appropriate parent in the upper layer. The routing process is computed as follows. First, a set of coefficients is initiated equally and added to 1. Then by multiplying a learnable weight matrix, every capsule output of a lower layer is projected to the same dimensional space with the capsules of the upper layer and the projected vector is scaled down with the coefficients, which are updated and renormalized using the dot-product results of the projected vectors and the capsules in the upper layer. They argued that this model automatically encodes the intrinsic relationship between different parts regardless of the change of the viewpoint. In the lexical information processing of our proposed model, we use a capsule network to handle the context dependent viewpoint of the words in high-dimensional embedding space.

3. Proposed model

In this section, we describe our proposed model for the endpoint estimation of conversational turns. It is composed of two subnets: one for lexical feature processing and the other for acoustic feature processing. The outputs of two subnets are combined with linear and softmax layers to produce the final estimation. For the lexical subnet, we used ten utterances as input. In our previous work, we used learnable speaker ID vectors to inform the model about the speaker for a particular utterance [8]. In this work, we found that a fixed speaker ID vector satisfied the same purpose. By adding a different speaker ID vector, the input is shifted to a different part of the hidden space, and the information regarding the utterances spoken by different speakers is preserved. For the acoustic subnet, we used the features for the current utterance as input. The outputs of the two subnets were combined to produce the final estimation.

3.1. Feature extraction

The proposed model uses both lexical and acoustic features to perform turn-taking estimation. We extracted spoken utterances from conversations based on the speech activity detection (SAD) results. A voiced interval between two pauses longer than 200ms is defined as an utterance (i.e., an inter-pause unit, IPU). Text transcriptions provide lexical features.

3.1.1. Lexical features

We first applied word segmentation using MeCab [1] since Japanese has no natural separators like *space* in Latin alphabet-based languages. We used Word2Vec [26] to convert tokenized words from a one-hot vector to distributed representation. Japanese Wikipedia data is used to train the word embedding. The embedding space's dimension was set to 300. By stacking the word embeddings, the lexical features for an utterance are a $300 \times \#$ *of words* matrix.

[1] A conditional random field-based Japanese morphological analyzer. http://taku910.github.io/mecab/

3.1.2. Acoustic features

For the acoustic features, we used 40 mel-frequency spectral coefficients (MFSCs, without de-correlating discrete cosine transform), computed these 40-dimensional vectors, and used them as frame representations. We used MFSCs instead of the more widely used mel-frequency coefficients (MFCCs) because the discrete cosine transform has been argued to project the spectral energies into a new basis to reduce the performance of deep neural networks since locality is not well maintained [27]. For a single utterance, the acoustic features can be arranged as a 2D feature map ($40 \times \#$ *of frames*).

3.2. Capsule network for lexical information

In the proposed model, lexical features are processed by a Capsule network that consists of convolutional, primary capsule layer, and a full connected capsule layer. Its architecture is depicted in Figure 1.

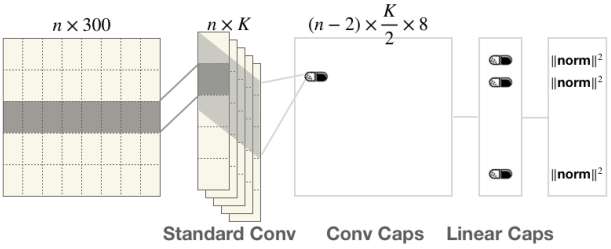

Figure 1: *Capsule network for text processing.*

3.2.1. Convolutional layer

The first layer in this capsule network is a standard convolutional network with 64 kernels, a stride of 1, and ReLU activation. This layer's kernel size was set to $1 \times V$ to reduce the dimensionality of the word embedding to the number of kernels, and V is the word embedding dimension (300 in this work). This layer conducts non-linear mapping from $x \in \mathbb{R}^{V \times n}$ to $\mathbb{R}^{(n) \times K}$, where n is the number of words in an utterance and K is the number of kernels. The word embeddings are converted to the activities of the local pattern detectors in this layer and used as the input of the next primary capsule layer.

3.2.2. Capsule layers

The second and third layers are a convolutional 8D capsule layer and a fully connected 16D (i.e., a capsule is a 8D or 16D vector) capsule layer. The convolutional layer has 32, $3 \times K$ kernels, where K is the number of kernels in the previous layer. In the convolutional capsule layer, the output of the previous standard convolutional layer is flattened to form a 2D input. The computation of the convolutional capsule layer resembles the one in the standard convolutional layer except that the kernel output is rearranged as 8D vectors, and non-linearity is induced by a *squash function*, which takes vectors as its input, instead of standard activation (i.e., ReLU, sigmoid, etc.). It limits the length of the input vectors between 0 and 1 but preserves their direction. The computation of *squash* is described as follows

where $\boldsymbol{s}_j$ is the input vector and $\boldsymbol{v}_j$ is the output vector:

$$\boldsymbol{v}_j = \textbf{Squash}(\boldsymbol{s}_j) = \frac{\|\boldsymbol{s}_j\|^2}{1+\|\boldsymbol{s}_j\|^2}\frac{\boldsymbol{s}_j}{\|\boldsymbol{s}_j\|}$$

We used the output capsules (8D vectors) as the input for a fully connected capsule layer to produce a set of 16D capsules. This process was called *dynamic routing* in the original paper [19]. Its computation is shown in Algorithm 1. s

Algorithm 1 Dynamic routing algorithm

procedure ROUTING($\hat{\boldsymbol{u}}_{j|i}, l, r = 3$)
  Initialize b_{ij} for all i in layer l and j in layer $(l+1)$: $b_{ij} \leftarrow 0$
  for r iterations **do**:
    for all capsule i in layer l: $\boldsymbol{c}_i \leftarrow \textbf{Softmax}(\boldsymbol{b}_i)$
    for all capsule j in layer $(l+1)$: $\boldsymbol{s}_j \leftarrow \sum_i c_{ij}\hat{\boldsymbol{u}}_{j|i}$
    for all capsule j in layer $(l+1)$: $\boldsymbol{v}_j = \|\boldsymbol{s}_j\| \cdot \boldsymbol{s}_j/(1+\|\boldsymbol{s}_j\|^2)$
    for all i and j: $b_{ij} \leftarrow b_{ij} + \hat{\boldsymbol{u}}_{j|i} \cdot \boldsymbol{v}_j$
  return $\boldsymbol{v}_j$

In the routing procedure, $\hat{\boldsymbol{u}}_{j|i}$ is linearly projected vector computed using output capsule $\boldsymbol{u}_i$ in the previous layer and learnable weight $\boldsymbol{W}_{ij}$. Every $\boldsymbol{s}_j$ before *squash* is a weighted combination of $\hat{\boldsymbol{u}}_{j|i}$, and the weight is the cosine similarity score between each $\boldsymbol{v}_j$ and $\hat{\boldsymbol{u}}_{j|i}$. Since $\boldsymbol{v}_j$ is unknown beforehand, the routing procedure uses an iterative manner to approximate it by starting from the average of $\hat{\boldsymbol{u}}_{j|i}$. After several iterations, $\hat{\boldsymbol{u}}_{j|i}$, which has a higher cosine similarity score with $\boldsymbol{v}_j$, is assigned a larger c_{ij} and contributes more in the decision procedure of $\boldsymbol{v}_j$. Finally, we combined the norm of the 16D capsules with the acoustic subnet's output.

3.3. Dilated convolutional network for acoustic information

Regarding acoustic information processing, a dilated convolutional network is used in the proposed model. The size of the CNN kernels was set to 3×3 with stride 3 in the time axis and stride 1 in the frequency axis since the acoustic features are computed on the basis of time-overlap windows. This subnet employed a dilated architecture to extend a receptive field that can look back through time. To transmit the order of the frames to the model, a previously proposed position encoding (PE) was added to the input vectors [28]:

$$\text{PE}(p, 2i) = \sin(p/10000^{2i/d})$$
$$\text{PE}(p, 2i+1) = \cos(p/10000^{2i/d})$$

where p is the index of the current frame and d is the dimension of the input vectors. We also added *Squeeze-and-Excitation* (SE) [29] blocks to the network to rescale the weight between the convolutional channels.

3.3.1. Dilated convolution

One limitation of CNNs in time series processing is that it can only cover a fixed range over time. Increasing the stride and/or the pooling size can linearly extend the receptive field with the growth in depth. However, this solution did not completely cover the historical data needed in acoustic feature processing (thousands of frames vs. tens of layers). A dilated architecture was proposed to address this problem [13]. The following is the computation of 1D convolutional filtering with dilation factor d:

$$\boldsymbol{F}(t) = \sum_{i=0}^{k-1} \boldsymbol{f}(i) \times \boldsymbol{x}(t - d \times i)$$

where $\boldsymbol{f}$ is a filter of length k and $\boldsymbol{x}$ is the signal we want to process. In a conventional CNN, dilation factor d is fixed number 1, and in this work d is increased exponentially with the layer number in the network (i.e., $d = 2^i$ in ith layer). We used 2D instead of a 1D dilation (which was used in the original work [13]) since we used 2D kernels. Unlike the common CNNs used in image processing, the inception field of a kernel moves only in the negative direction of the timeline since we cannot use information from the future. The left image in Fig. 2 illustrates the dilated convolution. We used 3×3 kernels instead of 2×2:

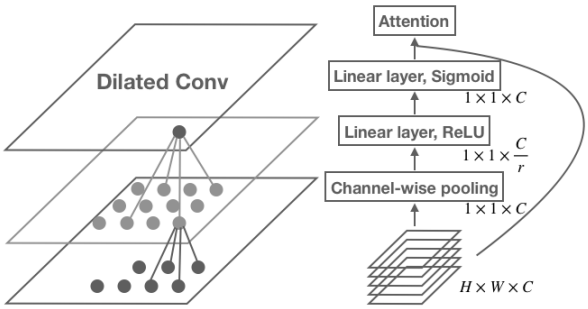

Figure 2: *Architectures of Dilated ConvNet and Squeeze-and-Excitation block.*

3.3.2. Channel-wise attention

The SE-block, which we added to this subnet, resembles a channel-wise attention mechanism. As depicted in the right image of Fig. 2, the channels in a hidden layer are rescaled after passing a SE-block. The attention weight for channels in a $H \times W \times C$ hidden layer is computed as follows. First, in the **squeeze** operation, an average pooling is conducted for each channel to produce a $1 \times 1 \times C$ vector:

$$w_c = \frac{\sum_i^H \sum_j^W h(i,j,c)}{H \times W}$$

where w_c is the weight for channel c and $h(i,j,c)$ is the neuron value at position (i,j,c). Then in the **excitation** operation, this vector passes through two linear layers with activations to reduce and resume the dimensionality. The result vector is used as the weight to rescale the channels in this hidden layer. This operation closely resembles the computation of self-attention [28] except that the parallel linear layers in it are replaced with serial ones. The SE-block gives the network the ability to learn the importance of different channels in a hidden layer. SE-net achieved the state of the art in an image classification task. We found that it also improved the performance of the neural networks in sequential data processing.

4. Evaluation

We experimentally evaluated the proposed model on a Japanese conversation corpus (identical as that previously used [10]) that consists of four types of conversations and over $30,000$ utterances. The type and the number of the sessions are shown in Table 1.

Table 1: *Corpus used in current work.*

Type (Sessions)	**Contents**
Dating (33)	Speakers talk about their personal interests and preferences to try to impress the interlocutor.
Interview (29)	One speaker is a job interviewer and the other is a candidate for the job.
Listening (19)	One speaker talks at length about a topic such as a memorable trip. The other mainly listen and occasionally responds with questions.
Receptionist (46)	One speaker is the receptionist of professor's office. The other is a visitor came w/o appointment.

Each type of conversation is randomly divided into three subsets: 70% as a training set, 10% as a validation set, and 20% as a test set. Since the proposed model takes a fixed input length, all the utterances are zero padded twice the length of the longest one in the training set in case an utterance exists with an unseen length in the validation or test sets.

We prepared two RNN-based models as baselines. One is from a previous work [10] that used two separated stacked LSTMs for lexical and acoustic features. The final hidden states of the stacked LSTMs are concatenated and used as input of a final fully connected layer. The other is a nested LSTM-based model (from our previous work [8]) that was originally designed for utterance classification using weighted word embedding as lexical features and the time series of the fundamental frequency as prosodic features. We slightly modified it so that it fits the format of the inputs and the outputs in this work.

The accuracy of the endpoint estimation results for all four types of conversations are shown in Table 2. On *dating*, *interviews*, *receptionist* types of conversation data, the proposed model significantly outperformed the two baselines. For the lengthy conversations in the *listening* type, the proposed model still showed slightly higher performance than the LSTM-based model. This result shows that CNN-based networks are also suitable for time series modeling.

Table 2: *Accuracy of turn-taking estimation on 4 conversation types.*

	Nested RNN	Stacked RNN	Proposed
Dating	79.78%	76.69%	**81.34**%
Interview	83.23%	83.50%	**86.37**%
Listening	79.06%	77.14%	**79.43**%
Receptionist	77.74%	76.83%	**82.61**%

5. Discussion

We attempted to completely replace conventional architectures with a capsule network. Unfortunately, unlike lexical subnets, all the combinations of hyper-parameters we tried for the acoustic subnet were outperformed by RNNs and CNNs. One explanation is that the distributed representation of words in the embedding space is easier to manipulate with a vector-based capsule network. Further works are required to apply this architecture to pure time series data like acoustic features.

The original work [29] used SE-block along with ResNet [17]. However, in this work, we did not find any performance improvement by adding a bypass to the network. Since we are using dilated networks, the upper layer has a different dimension than the lower layer. Thus, values from a lower layer go through the bypass and need a dimensional adjustment to meet the upper layer's requirement. We performed a 1×1 convolution on the lower layer and added it to the upper layer. This operation might be the reason that residual block lost its effectiveness in our CNNs.

The proposed model takes fixed length utterances as input. If an utterance exceeds the maximum length that the model can handle, then part of the information will be skipped. This remains a limitation of CNNs compared to RNNs. Although the capability of LSTMs for memorizing long-term information has not been fully investigated. It will be interesting to compare the length of memory that different neural networks can retain with identical parameter numbers, computational complexity, and so on.

6. Conclusions

In this paper, we investigated the possibility of replacing RNN variants (LSTM, GRU, etc.) with CNN variants (CapsNet, dilated ConvNet) in time series data processing.

We proposed a deep neural network-based model for estimating the endpoints of conversational turns. The proposed model handled sequential spoken data that include lexical and acoustic features with CNN variants. We used two subnetworks for the lexical and acoustic information respectively. One is a capsule network for the text processing and the other is a dilated convolutional network for the acoustic feature processing. Without conventional RNN-based architecture, the proposed model can be trained much faster than models using recurrent architectures. We experimented on a Japanese corpus containing four types of conversations: dating, a job interview, listening, and a receptionist situation. The experimental results showed that our proposed model outperformed the two currently proposed strong baselines.

7. Acknowledgements

This work was supported by JST, ERATO, Grant Number JPMJER1401. The authors wish to thank Dr. Divesh Lala for providing the dataset and the baseline model. We also thank Mr. Taiken Shintani and Ms. Taeko Murase for their help with the data preprocessing.

8. References

[1] R. Hariharan, J. Häkkinen, and K. Laurila, "Robust end-of-utterance detection for real-time speech recognition applications," in *Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2001, pp. 249–252.

[2] H. Sacks, E. A. Schegloff, and G. Jefferson, "A simplest system-

atics for the organization of turn-taking for conversation," *Language*, vol. 50, no. 4, pp. 696–735, 1974.

[3] S. Duncan, "On the structure of speaker-auditor interaction during speaking turns," *Language in Society*, vol. 3, no. 2, pp. 161–180, 1974.

[4] R. Stiefelhagen, C. Fugen, R. Gieselmann, H. Holzapfel, K. Nickel, and A. Waibel, "Natural human-robot interaction using speech, head pose and gestures," in *Proceedings. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 01 2004, pp. 2422 – 2427 vol.3.

[5] A. Koller, M. Staudte, K. Garoufi, and M. Crocker, "Enhancing referential success by tracking hearer gaze," in *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, ser. SIGDIAL '12, Stroudsburg, PA, USA, 2012, pp. 30–39.

[6] K. R. Thòrisson, *Multimodality in Language and Speech Systems*. New York: Kluwer Academic Publishers, 2002.

[7] A. Raux and M. Eskenazi, "A finite-state turn-taking model for spoken dialog systems," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, ser. NAACL '09, Stroudsburg, PA, USA, 2009, pp. 629–637.

[8] C. Liu, C. Ishi, and H. Ishiguro, "Turn-taking estimation model based on joint embedding of lexical and prosodic contents," in *Proc. Interspeech 2017*, 2017, pp. 1686–1690. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-965

[9] R. Masumura, T. Tanaka, A. Ando, R. Ishii, R. Higashinaka, and Y. Aono, "Neural dialogue context online end-of-turn detection," in *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 224–228.

[10] D. Lala, K. Inoue, and T. Kawahara, "Evaluation of real-time deep learning turn-taking models for multiple dialogue scenarios," in *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, ser. ICMI '18. New York, NY, USA: ACM, 2018, pp. 78–86.

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

[12] K. Cho, B. van Merriënboer, Ç. Gülçehre, 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)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734.

[13] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *Arxiv*, 2016. [Online]. Available: https://arxiv.org/abs/1609.03499

[14] J. Gehring, M. Auli, D. Grangier, and Y. Dauphin, "A convolutional encoder model for neural machine translation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 123–135.

[15] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. International Convention Centre, Sydney, Australia: PMLR, 06–11 Aug 2017, pp. 933–941.

[16] 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 Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[18] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," *arXiv e-prints*, p. arXiv:1608.06993, Aug 2016.

[19] S. Sabour, N. Frosst, and G. E Hinton, "Dynamic routing between capsules," 10 2017.

[20] A. Graves, "Generating sequences with recurrent neural networks." *CoRR*, vol. abs/1308.0850, 2013. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1308.html

[21] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. [Online]. Available: http://arxiv.org/abs/1409.0473

[22] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey." *CoRR*, vol. abs/1503.04069, 2017. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1503.html

[23] C. D. Santos and B. Zadrozny, "Learning character-level representations for part-of-speech tagging," in *Proceedings of the 31st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, E. P. Xing and T. Jebara, Eds., vol. 32, no. 2. Bejing, China: PMLR, 22–24 Jun 2014, pp. 1818–1826. [Online]. Available: http://proceedings.mlr.press/v32/santos14.html

[24] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 2014, pp. 1746–1751. [Online]. Available: http://aclweb.org/anthology/D/D14/D14-1181.pdf

[25] R. Johnson and T. Zhang, "Deep pyramid convolutional neural networks for text categorization," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 562–570. [Online]. Available: https://www.aclweb.org/anthology/P17-1052

[26] 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*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 3111–3119. [Online]. Available: http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

[27] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 22, no. 10, pp. 1533–1545, Oct. 2014. [Online]. Available: http://dx.doi.org/10.1109/TASLP.2014.2339736

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5998–6008. [Online]. Available: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

[29] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.