INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Turn-taking Estimation Model Based on Joint Embedding of Lexical and Prosodic Contents

Chaoran Liu[1], *Carlos Ishi*[1], *and Hiroshi Ishiguro*[1,2]
[1]ATR Hiroshi Ishiguro Lab, Japan.
[2]Graduate School of Engineering Science, Osaka University.
chaoran.liu@atr.jp, carlos@atr.jp, ishiguro@sys.es.osaka-u.ac.jp

Abstract

A natural conversation involves rapid exchanges of turns while talking. Taking turns at appropriate timing or intervals is a requisite feature for a dialog system as a conversation partner. This paper proposes a model that estimates the timing of turn-taking during verbal interactions. Unlike previous studies, our proposed model does not rely on a silence region between sentences since a dialog system must respond without large gaps or overlaps. We propose a Recurrent Neural Network (RNN) based model that takes the joint embedding of lexical and prosodic contents as its input to classify utterances into turn-taking related classes and estimates the turn-taking timing. To this end, we trained a neural network to embed the lexical contents, the fundamental frequencies, and the speech power into a joint embedding space. To learn meaningful embedding spaces, the prosodic features from each single utterance are pretrained using RNN and combined with utterance lexical embedding as the input of our proposed model. We tested this model on a spontaneous conversation dataset and confirmed that it outperformed the use of word embedding-based features.
Index Terms: turn-taking, recurrent neural network, word embedding, LSTM

1. Introduction

Human conversation is formed by giving and taking turns in a sequential structure, where one person speaks, followed by another person who takes his turn. This behavior has been researched since the early 1970s. Sacks et al. [1] and Duncan et al. [2] showed that turn-giving signals consist of lexical, prosodic, and gestural cues. Recognizing these signals seems simple enough, especially since people manage to do so in everyday conversation. However, Heldner et al. [3] argued that even in human-human conversations, fewer than 1% of turn transitions are actual zero gaps (less than a 10-ms silence) and over 40% of the transitions are overlapped.

Most current dialog systems are either system-leading or user-leading: a system that gives the initial question or guide and waits for the user's responses or vice versa. A natural dialog style is different. A natural dialog system requires the ability to allow both systems and users to take the lead at any time and correctly recognizes such situations [4].

The most common turn-taking strategy of a dialog system is to detect a silence longer than a hand-coded threshold (typically longer than $0.5s$) and/or follow a strict ping-pong-like fashion where user and system sequentially take the conversational floor. A system using this strategy can ignore such important conversational behaviors of users as interruptions, overlapping, and the co-completion of utterances [5][6].

In a natural conversation, perhaps one turn constitutes more than one utterance. Estimating turn-giving points properly is a requisite feature for a dialog system. To deal with this difficulty, research has deployed such non-verbal modalities as gestures [7] or gazes [8] as turn release signals. Other research used complex models that extensively rely on hand-coded expert knowledge [9][10]. However, expert knowledge is culture-dependent and difficult to transfer to other languages/cultures/tasks.

Data-driven methods [11] have also been proposed. Instead of using a fixed threshold of silence, these studies deploy statistic approaches. The use of silence to detect turn-giving points faces a trade-off between gaps and overlaps. With a short silence-threshold setting, the system reduces the gaps at the cost of increasing the chance of interrupting users. The statistic approach learns optimal thresholds from data. In a similar approach, Raux et al. used the partial lexical feature conversation history to improve system performance [12][13]. They reported an estimation error rate of 38% on a dataset in a system that provides bus schedule information by dialogs. Considering this reported performance on a question-and-answer style dataset, we view turn-ending point estimation as an unsolved problem.

In this work, we built a model that takes both lexical and prosodic features as input to predict turn-ending points. Our model also needs the ability to memorize conversation histories. For example, consider the following utterances:

"I want to go to Kyoto." "Kyoto?"

Without knowing the previous utterance, a system could not determine if the second single word utterance is a turn-ending point. In our proposal, a Recurrent Neural Network (RNN) processes a time series of utterances to memorize the context. Each RNN unit takes a previous state and a joint embedding of the lexical and prosodic contents as input and classifies utterances into different classes. The details are described in the Section 3.3.

2. Data

The database used in the present work includes sessions of two-party free-topic casual conversations between native Japanese speakers. Dialog data of 29 speakers (13 males, 16 females) were used. The length of each session was about $10 - 15$ minutes. The speech data were segmented in utterance units, and text transcriptions were provided for each utterance. We used a total of 27,656 utterances in the present work, including 6,186 turn-giving phrases ("g"), 7,190 turn-keeping phrases ("k"), 5,861 interrogative phrases ("q"), and 8,419 backchannels ("bc"). We categorized utterances into four target classes, defined by the following list:

- **k** (turn-keeping): the speaker is keeping her turn; a short pause or a clear pitch reset is accompanied at strong phrase boundaries.
- **g** (turn-giving): the speaker finished making her statement and is giving her turn to the interlocutor.

http://dx.doi.org/10.21437/Interspeech.2017-965

q (question): the speaker is making a question or asking for a confirmation from the interlocutor.

bc (backchannels): the speaker is producing backchannels to the interlocutor, e.g., "un", "hai" ("uh-huh", "yes")

The term "phrase boundary" used here refers to a linguistic boundary. These utterances might or might not be followed by a clear pause and cannot be correctly recognized just using the speech sound power. Backchannels are considered a different category from the other turn-giving utterances because their linguistic/phonetic characters are unique. Furthermore, recognizing whether a phrase is backchannel is critical for a dialog system since it means a user has no intention to interrupt and requires no response; the system should continue its turn.

We used the lexical and prosodic information of the dialog data as features for classification tasks. The feature preprocessing and tuning of the learning algorithm are described in the following section.

3. RNN for Turn-taking Estimation

3.1. Lexical features

Turn-ending point prediction is closely related to dialog comprehension. Without understanding the conversation context, people cannot predict when to take the conversational floor. Thus, considering the task of estimating turn-ending points, the most powerful information is lexical features.

To feed text into a computational model, a vector representation of texts must be found. In this classification task, we considered three vector representations of words: Word2Vec [14], GloVe [15] and FastText [16].

3.1.1. Word segmentation

Word segmentation is a critical first step in Japanese text analysis. In such Latin alphabet-based languages as English, space is a good approximation for word segmentation. However, in such eastern Asian languages as Japanese, Chinese, and Korean, no equivalent for this characteristic exists. In the present work, we use MeCab [17], a Japanese morphological analyzer that uses conditional random fields (CRF) as its learning algorithm [18], and a pre-trained IPA dictionary provided by the Information-technology Promotion Agency of Japan.

3.1.2. Sentence embedding

All three word embedding methods we used in this work learn vector representations of words from their co-occurrence information. Both Word2Vec and FastText based on skip-gram with negative-sampling training method. A problem affects Word2Vec is that this method can not represent a rare word not included in training set. FastText represents each word as a bag of character n-grams to overcome this out of vocabulary problem. A vector representation of word is the sum of these character n-gram representations. By and large, the training process of these two methods is similar to the training a language model to predict words around the given input words. With GloVe, a co-occurrence matrix of the entire training set is built first, then factorize it to yield matrices for vector representation of words and context. It takes into account the entire vocabularies' bias terms. These learnable bias terms gives an extra degree of freedom over Word2Vec and FastText. However, all these process requires a large dataset to achieve an accurate embedding space. For GloVe and FastText, since no pre-trained 100-dimension word embedding data are available, we used Japanese Wikipedia data plus our dataset to train it. For Word2Vec, Japanese word embedding is available for download from their GitHub page. The dimension of word embedding for GloVe and FastText is set to 100 in concert with those given by the Word2Vec team for small dataset. We tested three word embeddings with a simple, one hidden layer, feed-forward neural network whose output is a SoftMax activation function for four classes. The number of neurons in the hidden layer is set to 10 to keep the number of parameters less than one-tenth of the number of training samples. Joulin et al. [16] reported that a neural network of this size provides reasonable performance. The input is the sentence embedding given by Eq. 1.

$$E_s = \frac{1}{n}\sum_{i=1}^{n} E_w^{(i)} tf\text{-}idf_w^{(i)} \tag{1}$$

E_s and E_w are the embeddings for sentences "s" and word "w." n is the number of words in "s," and $tf\text{-}idf_w$ is the weight for "w," which was originally proposed by Salton et al. [19]. The number of occurrences of each word (i.e., term frequency) in a document is compared to the number of its occurrences in the dataset (i.e., inverse document frequency). The main idea here is to give discriminative words relatively more weight. The result is shown in Section 4.1.

3.2. Prosodic features

We believe that the intonation of utterances changes when the speaker intends to release his turn or to keep it. Thus, prosodic features provide additional clues for turn-ending point estimation. In this work, we used fundamental frequency and speech power as prosodic features. A time-series processing algorithm is needed since both are time sequences. There are several traditional algorithms for time-series classification.

The first algorithm we tested is a linear classifier. This algorithm's performance depends on the feature set used as the classifier's input. If we simply treat a time-series instance as a high-dimension vector, a slight time-shifted one might be recognized as a completely different instance. Hand-crafted feature extraction is needed to tune such algorithms.

The most wildly used algorithm for handling time-series data is HMM, which has been scrutinized in many kinds of applications. It consists of two stages: a continuous stage that divides the data observed in each time step into several clusters, and a discreet stage that takes a sequence of clusters and classifies them into discreet hidden states.

We trained four HMM model to classify a time sequence of the fundamental frequency, speech energy, and their derivatives (Δ) into the four turn-taking related classes mentioned in Section 2. These features are normalized to 5-mean unit-variance for each speaker/conversation. We used 5-mean instead of 0-mean since 0 is used to represent those periods no fundamental frequency detected. The class C a prosody-series belong to is decided by the maximum likelihood estimation $C = \underset{C\in\{k,g,q,bc\}}{\arg\max}\ P(\mathbf{\Theta}_{hmm}^{C}|\text{Prosody})$.

In addition, to complete the same task, we built two small neural networks: one with RNN and another with Gated Recurrent Units (GRUs [20]). A hidden layer in a traditional RNN is calculated using Eq. 2.

$$h_t = \sigma(\mathbf{W}^{hh} h_{t-1} + \mathbf{W}^{hx} x_t) \tag{2}$$

$\sigma(\cdot)$ is the activation function (sigmoid function), x_t is the input layer, h_t and h_{t-1} are the hidden layers in time steps t and $t-1$,

Σ is the sigmoid function, and $\mathbf{W}^{hh}$ and $\mathbf{W}^{hx}$ are the weights of the hidden-hidden and input-hidden connections.

In an RNN unit, the hidden layer of the previous time step is completely exposed to the current step. In contrast, a GRU uses update gate z_t and reset gate r_t to control the exposures of the previous step, following Eq. 3.

$$
\begin{aligned}
z_t &= \sigma(\mathbf{W}^z x_t + \mathbf{U}^z h_{t-1}) \\
r_r &= \sigma(\mathbf{W}^r x_t + \mathbf{U}^r h_{t-1}) \\
\tilde{h}_t &= tanh(\mathbf{W} x_t + r_t \odot \mathbf{U} h_{t-1}) \\
h_t &= z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t
\end{aligned} \tag{3}
$$

$\mathbf{W}^{(\cdot)}$ and $\mathbf{U}^{(\cdot)}$ are learnable parameters, $\tilde{h}_t$ is the memory content, and $\odot$ is the element-wise production.

We believe that the long-term movements of prosody depend more on lexical information than intonation, and our purpose here is to find the correlation of the turn-ending point and the intonation. Thus, we did not test the more complex Long-Short-Term Memory (LSTM [21]) units that can exploit long-term dependency. The test results are shown in Section 4.2.

3.3. Joint embedding

In the simple example in Section 1, if a model memorized the first utterance, it is easy to determine that there is a turn release point after the word "Kyoto" in the second utterance. In this subsection, we use RNN variants to realize the memories of a dialog history.

In previous subsections, we described the use of the lexical and prosodic features. Here we introduce a joint embedding of them and how to use them to train the RNN variants. In contrast to prosody, long-term memory in a dialog history might be more important. In a normal RNN, the effects of the information observed in a given time step will gradually fade after a certain number of time steps. To retain long-term memory, we used GRU units that were described in previous subsections and LSTM units.

Several LSTM variations exist. In this work, we use a widely tested model, described by Zaremba et al. [22]. The hidden layer is calculated based on Eq. 4.

$$
\begin{aligned}
i_t &= \sigma(\mathbf{W}^i x_t + \mathbf{U}^i h_{t-1}) \\
f_t &= \sigma(\mathbf{W}^f x_t + \mathbf{U}^f h_{t-1}) \\
o_t &= \sigma(\mathbf{W}^o x_t + \mathbf{U}^o h_{t-1}) \\
\tilde{c}_t &= tanh(\mathbf{W}^c x_t + \mathbf{U}^c h_{t-1}) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\
h_t &= o_t \odot tanh(c_t)
\end{aligned} \tag{4}
$$

i_t, o_t, and f_t are respectively the input, output, and forget gates in time step t. $\tilde{c}_t$ and c_t are the new and final memory cells. Other denotations are the same as those in Eqs. 2 and 3.

We also tested an architecture using an RNN with max pooling ($h^{(i)}_{pooling} = \max_n\{0, h^{(i)}_n\}$). A pooling layer is generally used with the CNN layer. Johnson et al. [23] used it with an LSTM layer and reported an improved result in a sentiment analysis task. We use a simplified version in this work that replaced the LSTM layer with a normal RNN layer. A normal RNN cannot satisfy long-term dependency due to the absence of a memory unit. Intuitively, if a pooling layer is added to a RNN layer, it can behave as a long-term memory unit. This network's architecture is shown in Figure 1.

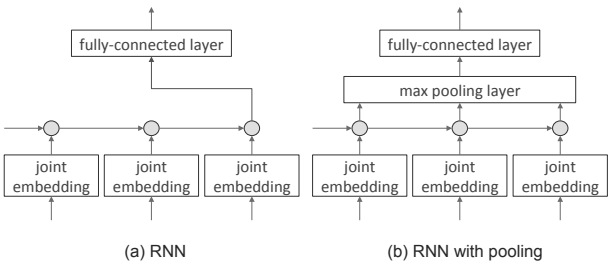

Figure 1: *Traditional RNN and RNN with a max pooling layer.*

The overall process flow is shown in 2. In the training process of these networks, one utterance is used at each time step. The lexical and prosodic features are combined to form a joint embedding of utterances. When training modes through time,

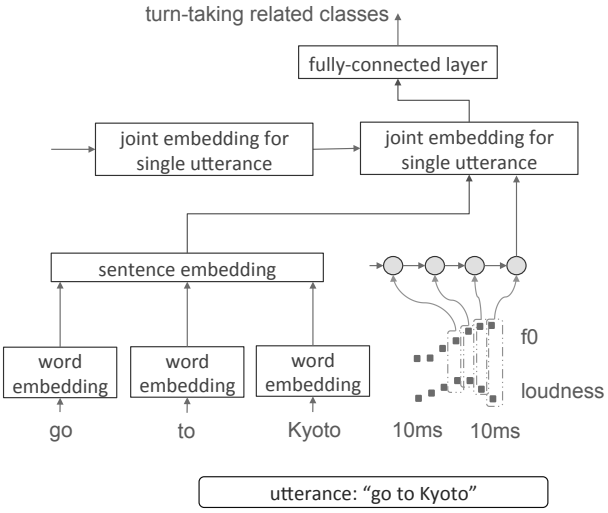

Figure 2: *Proposed turn-taking estimation model.*

utterances that are used as input are spoken by different speakers. Since we use a two-party conversation in this work, the speaker ID can be described using a two-dimensional one-hot vector. We modified the activation function in Eq. 2, 3 and 4, replaced $\mathbf{W}^{(\cdot)} x_t$ with $\mathbf{W}^{(\cdot)} x_t + \mathbf{V}^{(\cdot)} h_{ID}$. h_{ID} is the vector representation of speaker ID. This method was first used in "WaveNet" [24] to allow a neural network to learn the speaker dependent features. We used it here to indicate which speaker made the utterance.

4. Experiment and discussion

We randomly divided our dataset into three parts: 80% as a training set, 10% as a validation set, and 10% as a test set.

4.1. Sentence embedding

First, we test a simple feed-forward neural network with sentence embeddings as its input. The target output is the turn-taking related classes described in Section 2. Given word vector representations, a sentence embedding is calculated using Eq. 1. This feed-forward neural network takes one sentence at a time and predict which class it belongs to. Table 1 shows the classification results on the validation set for three different

word embeddings. The dimension of word vector is 100.

Table 1: *Classification accuracies for different lexical features.*

word embedding	Word2Vec	FastText	GloVe
classification acc.	54.2%	57.3%	**59.1%**

In this test, we found that GloVe slightly outperformed Word2Vec and FastText while Word2Vec gave the lowest classification accuracy. Levy et al. [25] pointed out that in word similarity-oriented tasks, the difference for neural network-based approaches like Word2Vec and GloVe are insignificant. Even compared to those traditional count-based approach such as Pointwise Mutual Information (PMI) matrices and SVD-based Latent Semantic Analysis (LSA), with carefully tuned hyper-parameters (context distribution smoothing, eigenvalue weighting, vector normalization, etc.), there is no global advantage to any method over the others. We think that the difference in our test results came mainly from the difference in the training dataset.

4.2. Prosody

Similar to the lexical features, we also tested prosodic features on the validation set. Time series of normalized fundamental frequency, speech energy (loudness), and their derivatives are used as input. Four classifiers (HMM, linear SVM, RNN, GRU) are used in this experiment. For linear SVM, since it can not accept feature vectors with variant length, a fixed dimension set to 20 (200-ms, 10-ms per frame) is employed with end point aligned. Duncan argued that the final syllable of an utterance can be considered as a marker of the turn-ending point [2]. Therefore, we chose the length of 200-ms as it is the average mora length in Japanese. Table 2 shows the experimental results for different classifiers.

Table 2: *Classification accuracies for different classifiers using prosodic features.*

classifier	HMM	SVM	RNN	GRU
classification acc.	41.4%	33.8%	40.8%	**42.2%**

The results in Table 2 are lower than those reported in previous works due to the following three reasons. 1) In our experiment, the classification targets are four classes instead of two (turn-ending point or not). Thus, the chance level is decreased from 50% to 25%. 2) We do not use silence as prosodic feature since our purpose is to build a turn-taking estimation model with minimal gap. 3) Most of previously reported results are on Q&A like conversation data. Our dataset consists of spontaneous conversation with largely varied intonations.

With the same prosodic features, SVM gave the poorest performance due to its limitation when dealing with time series data. Surprisingly, HMM achieved nearly the same result with GRU and outperformed RNN. A neural network based method might need a larger dataset to fully utilizing its potential. These results indicated that using prosodic features only is not efficient for turn-taking estimation.

4.3. Joint embedding

We proposed a model that takes joint embedding of lexical and prosodic features as input, as well as dialog histories (Figure 2). The hidden layers in feed-forward/GRU network described in the previous subsections was concatenated as a joint embedding for each utterance. The time series of joint embedding is used to perform turn-taking estimation. GRU, LSTM and RNN with pooling layer (see Section 3.3) are used in this experiment. The experimental results on the test set is shown in Table 3.

Table 3: *Classification accuracies using variants of RNNs.*

GRU	73.7%
LSTM	**74.6%**
pooling-RNN(1)	72.2%
pooling-RNN(2)	72.9%
pooling-RNN(3)	73.8%
pooling-RNN(4)	73.0%

The number in $(\cdot)$ indicates the number of time-steps that are connected to the max-pooling layer, where pooling-RNN(1) is equivalent to normal RNN. The best classification accuracy was given by LSTM. The score is significantly higher than using lexicon or prosody only.

4.4. Discussion

The vocabularies used in spoken language are distinct from those in written language. Although the results in the present work are promising, the conversation dataset used is still small. In order to represent lexical context accurately, a large conversation dataset is needed. We are considering to construct larger conversation dataset and build accurate conversational word embedding based on it in our future work.

The effect of long term memory of dialog history remains ambiguous. We will further analyze it by changing pooling size or/and using new models with external memories to find an optimal strategy to estimate turn-taking behaviors.

5. Conclusions

In the present work, we analyzed a dataset of two-party spontaneous conversations and proposed a model to classify utterances into 4 prime turn-taking related classes (turn-keeping, turn-giving, backchannel and question).

With lexical context only, by presenting sentence using word embedding, we obtained a best classification accuracy of 59.1% with GloVe model. However, with prosody information only, a lower classification accuracy of 42.2% is obtained by a recurrent neural network with gated recurrent unit using fundamental frequency and speech loudness as its input. Therefore, we considered that in a turn-taking behavior estimation task, the lexical context is a more powerful feature than prosody.

Finally, we proposed a model using joint embedding of both lexical and prosodic features. The proposed model also takes dialog histories into account to better predict turn-taking behaviors. A significantly higher result of 74.6% is obtained on the proposed model by using a long short-term memory recurrent neural network compared to the ones using lexical features or prosodic features only.

6. Acknowledgements

This research was supported by JST, ERATO, Grant Number JPMJER1401.

7. References

[1] Sacks, H., Schegloff, E., and Jefferson, G. (1974). A simplest systematics for the organization of turn-taking in conversation. Language 50, 696-735.

[2] Duncan, S. D. (1974). On the structure of speaker-auditor interaction during speaking turns. Lang. Soc. 2, 161-180.

[3] Heldner, M., and Edlund, J. (2010). Pauses, gaps and overlaps in conversations. J. Phon. 38, 555-568.

[4] Horvitz, E. (1999). Principles of mixed-initiative user interfaces. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 159-166, Pittsburgh, USA, 1999.

[5] Str?m, N. and Seneff, S. Intelligent barge-in in conversational systems. In Proceedings of the 6th International Conference on Spoken Language Processing, pages 652-655, Beijing, China, 2000.

[6] Baumann, T. Incremental Spoken Dialogue Processing: Architecture and Lower-level Components. PhD thesis, Universitat Bielefeld, Germany, 2013.

[7] Stiefelhagen, R., Fugen, C., Gieselmann, R., Holzapfel, H., Nickel, K. and Waibel, A. Natural human?robot interaction using speech, head pose and gestures. In Proceedings of the IEEE-RSJ International Conference on Intelligent Robots and Systems, pages 2422-2427, Sendai, Japan, 2004.

[8] Koller, A., Garoufi, K., Staudte, M. and Crocker, M.W. Enhancing referential success by tracking hearer gaze. In Proceedings of the 13th Annual SIGDIAL Meeting on Discourse and Dialogue, pages 30-39, Seoul, South Korea, 2012.

[9] Thorisson, K. R. (2002). Multimodality in Language and Speech Systems, chapter Natural Turn-Taking Needs No Manual: Computational Theory and Model, From Perception to Action, pages 173-207. Kluwer Academic Publishers.

[10] Kronild. F. (2006). Turn taking for artificial conversational agents. In Cooperative Information Agents X, Edin- burgh, UK.

[11] Ferrer, L., Shriberg, E. and Stolcke, A. (2003). A prosody-based approach to end-of-utterance detection that does not require speech recognition. In ICASSP, Hong Kong.

[12] Raux, A. and Eskenazi, M. (2008). Optimizing endpoint- ing thresholds using dialogue features in a spoken dialogue system. In Proc. SIGdial 2008, Columbus, OH, USA.

[13] Raux, A. and Eskenazi, M. (2009). A Finite-State Turn-Taking Model for Spoken Dialog Systems. In Proc. NAACL 2009, Stroudsburg, PA, USA.

[14] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J. (2013). Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems.

[15] Pennington, J., Socher, R. and Manning, C. GloVe: Global Vectors for Word Representation. Empirical Methods in Natural Language Processing (EMNLP). 2014.

[16] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, (2016). "Bag of Tricks for Efficient Text Classification". arXiv preprint arXiv:1607.01759

[17] MeCab: Yet Another Part-of-Speech and Morphological Analyzer, http://taku910.github.io/mecab/

[18] Kudo, T., Yamamoto, K., Matsumoto, Y. ApplyingConditional-Random Fields To Japanese Morphological Analysis. In Proc. of Conference on Empirical Methods in Natural Language Processing, 2004.

[19] Salton, G. and McGill, M. editors. IntroductiontoModernInformation Retrieval. McGraw-Hill, 1983.

[20] Cho,K., Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In proc. of EMNLP, 2014.

[21] Hochreiter, S., Schmidhuder, J. Long short-term memory. Neural Computation, 9(8): 1735-1780, 1997.

[22] Zaremba, W., Sutskever, I. Learning to execute. arXiv:1410.4615, 2014

[23] Johnson, R., Zhang, T. Supervised and Semi-Supervised Text Categorization using LSTM for Region Embeddings. In Proc. ICML. 2016.

[24] Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K. WaveNet: A Generative Model for Raw Audio. arXiv:1609.03499. 2016.

[25] Levy, O., Goldberg, Y. and Dagan, I. Improving Distributional Similarity with Lessons Learned from Word Embeddings. Transactions of the Association for Computational Linguistics. Vol. 3. 2015.