Toward Multi-lingual Natural Language Understanding using Deep Neural Networks

Abstract

Recent approaches based on deep neural networks have shown promising results for natural language understanding. However, those models are limited on a monolingual corpus. In this work we present a multilingual semantic decoder which combines two neural networks, convolutional neural network and long-short term memory network. To achieve multilinguality we mainly focus on universal aspects of how language is structured; first, multiple *signifiant*(s) for a single *signifie,* and secondly an utterance depends on the serious of previous utterances in a dialogue. Training and evaluation are evaluated for two corpora established in different languages: SGSDS corpus and the publicly available DSTC2 corpora. Two corpora are identically annotated by the slot-filling framework, which enables to represent the conceptual meaning from an utterance irrespective of certain morpho-syntactic features of languages such as word order or morphemic variation. The architecture of our concatenated model encompasses these universal features by classifying a set of dialogue act-slot-value triplets to a corresponding utterance and by utilizing previous system's utterances. It is observed that our model is multi-linguistically applicable by classifying a set of dialogue act-slot-value triplets on both corpora built in Korean and English, respectively.

1 Introduction

Natural language understanding (NLU) has been one of the most rudimentary components in human-machine conversation (Wang et al., 2005). In order to achieve NLU, there is a need to capture pragmatic intention and to extract semantic meanings from an almost infinite variety of a user's utterances in the middle of dialogues. In the light of this situation, a semantic decoder should be able to consider both the current user's utterances and the previous conversation. Although Rojas-Barahona et al. (2016) proposes a joint model that is capable of capturing both sentence- and context- representation, their work is only evaluated on English corpora (i.e., DSTC2 and in-car datasets). This naturally raises a question whether this joint model could be extended across different languages.

The aims of this study are two folds: first, to build a robust semantic decoder by concatenating two deep neural networks to exploit multiple inputs, and secondly to conduct a dialogue act-slot pairs classification task on two corpora – one for Korean (i.e., SGDSG[1]) and the other for English dialogues (i.e., DSTC2). To fulfill this objective, we will briefly review previous studies of NLU in section 2. Section 3 will present the details of the architecture of our concatenated model, and section 4 will summarize the experimental set up. Section 5 will provide the experiment result that shows the robustness of our semantic decoder. In Section 6, we will conclude the paper and remarks on main findings of our study with implication of our future research.

2 Related Works

With the development of deep learning, typical deep learning models such as convolutional neural networks (CNN) and recurrent neural networks (RNN) have achieved remarkable results in several natural language processing tasks (Kim 2014; Mikolov 2010; Socher et al., 2012). Recently, several researches have been conducted by combining CNN and RNN models. Kim (2016) proposes a recurrent convolutional network (RCNN) model, in which the penultimate layer of CNN is connected

[1] Sogang Dialogue System Group. Since there had been no dialogue corpus in Korean, building a dialogue corpus with transcribed texts is the highest-priority task in our research.

Corpus	Speaker	Utterance	Act	Slot	Value
SGDSG	System	무엇을 도와드릴까요? *mwuesul towatulilkkayo*? "How can I help you?"	Hello	N/A	N/A
	User	나 내일 회의일정 등록해줘. *na nayil hoyuyilceng tunglokhaycwu.* "Schedule a tomorrow's meeting."	Inform	SYSTEM_ACTION	create
			Inform	DATE	tomorrow
			Inform	EVENT_TITLE	meeting
DSTC2	System	What kind of food would you like?	Request	FOOD	N/A
	User	Cheap Indian food	Inform	FOOD	Indian
			Inform	PRICE_RANGE	cheap

Table 1: Example of utterances of two corpora annotated with set of dialogue act-slot-value triplets.

to the recurrent layers in the RNN model to track a topic of a dialogue in human-human conversations.

Another jointed CNN and RNN model proposed by Rojas-Barahona et al. (2016) is different from previous CNN-RNN models, in that they optimize the model with two distinctive inputs: a current user' utterance and act-slot pairs of previous system utterances. In the task of decoding semantic meaning of spoken languages each input is utilized in sentence representation and context representation, respectively.

3 Models

In this section, we will introduce the architecture of our semantic decoder, as illustrated in Figure 1. To capture the semantic meaning from utterances regardless of the distinct type of languages, we are grounded on universal nature of how language is structured. All the linguistic sign is made up of the matched pair of *signifiant* (i.e. phonetic or image form) and *signifié* (i.e. the concept of the meaning indicated by the *signifiant*), and the their elationship between is arbitrary, so there could be various forms of *signifiant* to denote a single *signifié* (Saussure 1916). To extent to the dialogues, s*ignifiant* is analog to actual sentences uttered in different languages and *signifié* is so to be a propositional meaning of each utterances. Additionally, an utterance is always dependent on the previous utterances in a single dialogue.

Our model is built to predict a correct set of dialogue act-slot-value triplets annotated to a corresponding user utterance, as shown in Table 1. The act-slot-value triplets are expected to represent the core concept of propositional meaning irrespective of diverse word orders and morphology of utterances across the languages, since this annotation framework thoroughly consists of two categories (i.e. dialogue *act* and *slot-value* pair(s), which are divided into two roles (i.e. '*pragmatic intention'* and '*semantic information'*). In the light of this situation, we concatenate CNN and LSTM in that the former is specialized in extracting necessary information regardless of different word orders and the latter is good at retrieving previous dialogue history.

3.1 Convolutional Neural Network

In this CNN architecture, a sentence of length n is represented as a $n \times k$ matrix, and each row of the matrix is a k dimensional morpheme embedding vector $x_i \in \mathbb{R}^k$ representing the i-th word in a sentence. Each word in a sentence is segmented into several morphemes by *Komoran* (Park and Cho, 2014), which are initialized into embedding vectors for an input layer.

A convolutional operation involves a filter $\boldsymbol{m} \in \mathbb{R}^{h \times k}$ is applied to a window of h rows to produce a feature:

$$c_i = f(\boldsymbol{m} \cdot x_{i:i+h-1} + b)\,, \qquad (1)$$

where f is a hyperbolic tangent function and $b \in \mathbb{R}$ is a bias term. These series of convolutional operations are applied to all the possible windows and generate a feature map:

$$\boldsymbol{c} = [c_1, c_2, \dots, c_{n-h+1}]\,. \qquad (2)$$

Then a max pooling is operated to take the maximum value $\hat{c} = \max\{\boldsymbol{c}\}$ as a representative feature for the filter.

In our model, multiple filters with varying window size h are integrally engaged to obtain multiple adjacent features. These features are then concatenated to form the 'top-level' feature vector s_t, which embeds features of a user utterance at a dialogue turn t.

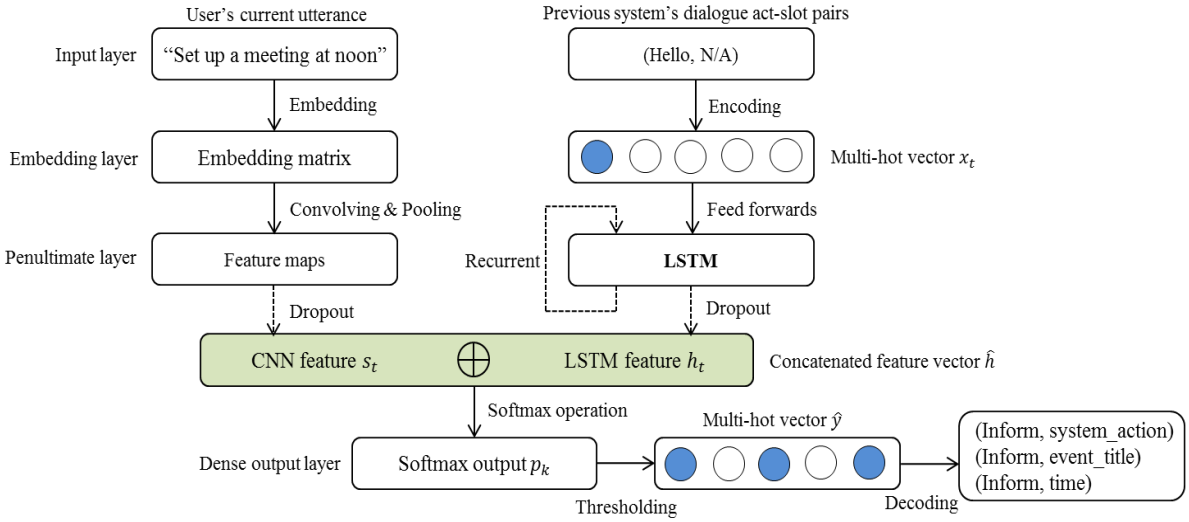

Figure 1. The Architecture of our concatenated CNN-LSTM model.

3.2 Long-Short Term Memory Network

Since each utterance is dependent on the previous utterances in a conversation, it is necessary to refer to dialogue act-slot instances of a system's utterance right before a user says. To receive assistance from the previous system's utterances at an inter-utterance level as well, we employ a long-short term memory (LSTM) (Hochreiter and Schmidhuber, 1997), a special kind of recurrent neural network (RNN) which is much better for preserving in formation over long periods of time than other kinds of traditional RNN.

The structure of LSTM is divided into a memory cell c_t and three gates: a forget gate f_t, an input gate i_t and an output gate o_t. Three kinds of gates functions to decide which amount of information the memory cell should keep or forget at a time step t. The input x_t and the output h_t of LSTM are updated as follows:

$$i_t = \sigma(W^i \cdot x_t + U^i \cdot h_{t-1} + b^i) \quad (3)$$
$$f_t = \sigma(W^f \cdot x_t + U^f \cdot h_{t-1} + b^f) \quad (4)$$
$$o_t = \sigma(W^o \cdot x_t + U^o \cdot h_{t-1} + b^o) \quad (5)$$
$$g_t = tanh(W^g \cdot x_t + U^g \cdot h_{t-1} + b^g) \quad (6)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad (7)$$
$$h_t = o_t \odot tanh(c_t) \quad (8)$$

where x_t is the input at the current time step, h_t is the hidden unit at time step t, b is a bias term, $\sigma(\cdot)$ is a logistic sigmoid function and $\odot$ denotes a point-wise multiplication operation.

Unlike the model proposed by Rojas-Barahona et al. (2016), where the word vectors are fed into LSTM as inputs, we encode corresponding dialog act-slot pairs of previous system's utterances at a time step t into a single multi-hot vector, and stipulate them as the input x_t. As the single multi-hot vector represents multiple labels of dialogue act-slot pairs as a whole and is fed into LSTM at a time t, it facilitates the network to grasp semantic information at an inter-utterance level more straightforward.

3.3 Concatenating CNN and LSTM

We notice that the contextual flow of a conversation works as just an auxiliary information in extracting the semantic meaning from the current user's utterances. While feature vectors in the penultimate layer of CNN and RNN are merged by a tangent function in Rojas-Barahona et al. (2016), we concatenate the hidden unit h_t of LSTM to the 'top-level' feature vector s_t modeled by the CNN. Then, the concatenated vector $\hat{h}_t = s_t \oplus h_t$ is passed to a fully connected softmax layer whose output is the probability distribution over all labels of dialogue act-slot pairs as described in Figure 1.

The softmax operation over each prediction is calculated as follows:

$$P(y_k = 1 | \hat{h}, W, b) = \frac{\exp(W_k \cdot \hat{h} + b_k)}{\sum_j \exp(W_j \cdot \hat{h} + b_j)} \quad (8)$$

where k denotes the index of the multi-hot vector y, which represents the dialogue act-slot pairs of user's utterance.

3.4 A Threshold Predictor

In both datasets of SGDSG and DSTC2, more than one label of a dialogue act-slot pair is annotated to a given single user's utterance. To perform this multi-label classification task, we use

the output probability distribution p_k for a given utterance $\boldsymbol{x}$ from the softmax layer. The predicted multiple label of act-slot pairs $\hat{y}$ for an utterance $\boldsymbol{x}$, is determined by a threshold t as follows:

$$\hat{y} = \{y_k | p_k > t;\ k \in L\} \quad (9)$$

The threshold learning mechanism used in the literature (Elisseeff and Weston, 2001; Nam et al., 2014) is adopted, which models t with a linear regression model.

4 Experimental Setup

4.1 Corpus Development

Evaluation of model is conducted on two datasets: SGDSG in Korean and DSTC2 (Henderson et al., 2014b) in English. For the case of Korean there has been no dialogue corpus in Korean, we first establish an annotated dialogue corpus. We set up a forum of conversation with 20 test subjects using a Wizard of Oz methodology; we created an environment where a test subject believes to interact with a computer system which is actually a hidden operator[2] (Rieser and Lemon, 2008). We asked each test subjects to perform 15 dialogues with accordance to the precise tasks on the topic of schedule management. The user is able to create, read, update, and delete schedules with the details such as start date, alert, event title, and location to update or access to database.

The DSTC2 corpus is collected for the purpose of providing restaurant information in the city of Cambridge. The system searches a restaurant by constraints such as area, price range, and food type, and specifies a certain place with other information such as address and phone number.

Both corpora are annotated by the identical framework (i.e. dialogue act-slot-value triplets of the DSTC2), as illustrated in Table 3.

4.2 Annotation Framework

As mentioned in Section 3 a theoretically universal tagging framework is in demand to create a simple and direct mapping between a sentence of user's utterance and a dialogue acts. To build an annotation set which is more appropriate for a system to respond naturally, we adopted the framework of DSTC2, which are effective enough to extract users' intention and are specialized enough to determine a system operation.

Corpus	U	D	L	C
SGDSG	6,480	1,529	33	1.48
DSTC2_train	11,677	3,934	101	1.23
DSTC2_test	1,612	506	76	1.20

Table 2: Statistics of SGDSG and DSTC2 corpus. **U**: Number of utterances spoken by a user. **D**: Number of total dialogues. **L**: Size of all possible dialogue act-slot pairs in the corpus. **C**: Average number of dialogue act-slot pairs tagged per utterance.

The table 3 summarizes our revised version of dialogue act set for user utterances. The dialogue act set is designed by reducing unnecessary several tagging layers and directly representing the user's intention in terms of the most appropriate response for a system in the next turn. For example, utterances in SGSDS corpus are collected as follows: What's going on tomorrow?, Would you read my schedule on tomorrow?, I need to check things to do tomorrow, *I was wondering if I have plans for tomorrow*, *Read my tomorrow's schedule.* These utterances tagged into a set of dialogue act-slot-value triplets: inform {(system_action, read), (date, tomorrow)}. A system can achieve 2 goals at a time; it can both directly understand the meaning of those 5 users' utterances and prepare to operate READ SCHEDULE action.

Consequently, this tagging set is expected to applied in a wide variety of domains for detecting the intentions of users, since this framework consists of two types of objects; it thoroughly divides two roles *pragmatic intention* and *semantic information* into two categories: *dialogue act* and *slot-value pair*(s). When it comes to expand a topic of dialogues, a new ontology is only needed to superinduce additional semantic information to the primitive knowledge of a system, without adding new categories of dialogue act.

4.3 Hyper-parameters and Training

In our experiments, we use: filter windows (h) of 2, 3, 4 with 200 feature maps each for the CNN, dimension of 128 for the hidden unit of LSTM and a batch size of 60. As a means of regularization, we apply Dropout on the penultimate layers of both the CNN and the LSTM with dropout rate of 0.2. Those values are chosen by performing a rough

[2] Specifically, a test participant may think he or she is communicating with a computer using a speech interface; while the participant's sayings are being secretly transmitted into the computer of an operator (i.e. "wizard") in another room, and the operator types proper response in the texts which are transformed into acoustic waves into the computer of the test subject. This methodology helps gain relevant training data of human-to-robot dialogues.

	Act	Slot	Definition
1	ack	empty list	An acknowledgement e.g. "okay"
2	affirm	empty list	An affirmative reply to the system's previous utterance e.g. "yes"
3	bye	empty list	A goodbye message of the user indicating that the conversation is finished
4	negate	empty list	A negative statement or a denial of the system's previous utterance e.g. "no"
5	null	empty list	Something not understandable to the system; outside its domain e.g. ″pineapple″
6	repeat	empty list	A request for the system to repeat what it just said e.g. ″please repeat that″
7	reqalts	one pair (s, v)	Requesting for changing the existing value of the slot; asking for alternative suggestions of the given value of the slot e.g. "are there any others"
8	restart	empty list	Asking the system to restart from the beginning e.g. ″let′s start again″
9	thankyou	empty list	User thanking the system e.g. ″thanks″
10	deny	one pair (s, v)	Informing that the user does not want the value v for a certain slot s. s must be an informable slot and v a possible value for s as specified in the ontology.
11	request	one pair ("slot", s)	Asking the system for the value of the requestable slot according to the ontology.
12	inform	one pair (s, v)	Notifying that the value of slot s to be accepted as v
13	hello	empty list	Greeting the system e.g. ″hi″
14	reqmore	empty list	Asking the system if the user can request more information

Table 3: The list of Dialogue acts for User's utterances.

grid search (Zhang and Wallace, 2016). The model undergoes training through stochastic gradient descent over shuffled mini-batches with RMSprop update rule. Our model stops the iterant processes of learning by an early stopping mechanism.

4.4 Model Variations

To evaluate the classification performance of our CNN-LSTM combined model, we compare the performance of three models:

- CNN (multiclass): The model that predicts only one dialogue act-slot for given user's utterance.
- CNN (thresholding): The CNN model with a threshold predictor that classifies multiple labels of dialogue act-slot pairs
- CNN-LSTM (thresholding): The model which concatenates CNN and LSTM. It allows to extract information from both a current user's and previous system' utterances.

5 Results and Discussion

In this section we show the evaluation results on two corpora by conducting a 5-fold cross validation task. All models iterate the evaluation process 20 times, and the mean scores of each evaluation metric are calculated. Table 4 summarizes the performing results of each model on the two corpora. Compared two CNN models, we observe that the performance of the CNN model is significantly improved with the help of a threshold predictor, which enables the semantic decoder to predict multiple pairs of dialogue act-slot. It is observed that the CNN model maintains the desirable performance[3] on both Korean and English[4]. Further improvements are achieved by CNN-LSTM model on Korean corpus. It points that this conjoined model more effectively capture the semantic meaning by referring to contextual information: previous system's utterances.

Table 5 summarizes the performance of CNN and CNN-LSTM models on classifying act-slot-value triplets per each slot of SGSDS corpus. Overall, our semantic decoder shows promising results of classifying correct sets of act-slot-value triplets except the slot *event title* and *time_new*[5]. It is worth noting that though Korean and English are morphologically and syntactically far different, our CNN-LSTM model steadily predicts correct label set of

[3] We only exploit top-1 ASR hypothesis in the DSTC2 corpus at this stage.

[4] Compared to the best-performing model on the DSTC2 corpus (Rojas-Barahona et al. 2016), they achieve the following results in the CNN+LSTM_w4 configuration: P=89.77%, R=86.40% F1=88.03%.

[5] For the case of the slot *event title*, it requires for a semantic decoder to extract the meaning from a variety of agglutinated verbal forms of Korean. It is concerned to utilize unsupervised word embedding algorithms such as GloVe or word2vec in the future research.

Model	SLU Task	SGSDS			DSTC2		
		Precision	Recall	F-measure	Precision	Recall	F-measure
CNN(multiclass)	Act-slot	73.43	49.54	59.16	-	-	-
CNN (Threshold)	Act-slot	92.27	85.58	88.79	89.18	82.00	85.44
	Act-slot-value	87.01	74.96	80.52	89.38	78.37	83.51
CNN-LSTM	Act-slot	96.75	88.98	92.63	90.61	80.50	85.26
	Act-slot-value	90.90	78.71	84.31	84.75	75.63	79.93

Table 4: Overall evaluation results of the proposed models on SGSDS and DSTC2 corpus

Slots	CNN			CNN-LSTM		
	Precision	Recall	F-measure	Precision	Recall	F-measure
Domain	96.61	93.63	95.07	97.82	99.26	98.53
System_action	94.58	87.47	90.87	95.17	88.73	91.83
Date	93.30	88.99	91.07	93.85	88.65	91.15
Time	89.02	66.72	76.23	88.72	58.03	69.62
Event_title	55.05	43.47	48.53	52.52	40.27	45.56
Repeat	88.66	58.50	70.47	87.40	44.50	58.64
Alert	96.51	64.80	77.26	93.78	67.73	78.11
“Slot”	59.65	97.00	67.90	87.45	94.00	87.73
Read details	88.59	50.59	63.56	100.00	70.54	75.95
Date.from	93.33	70.59	78.40	98.46	76.47	86.06
Date.to	97.14	80.00	86.00	100.00	88.57	93.84
Date.new	73.33	40.00	47.94	90.00	46.67	60.00
Time.new	49.91	40.00	44.30	34.17	21.05	25.58

Table 5: Classification results on SGSDS corpus: Act-slot-value triplets per each slot

dialogue act-slot-value triplets well, without using any manually designed feature or preprocessing the data through delexicalization. The results suggest that our semantic decoder has a potential to conduct a NLU task across languages.

6 Conclusion and Future research

In this paper we aim at developing a novel approach to automatically acquire knowledge to detect the meaning from user’s utterances with going beyond the limitation of monolingual corpus. We have presented a multilingual semantic decoder based on CNN-LSTM approach. Based on primary and universal feature of language structure, the annotation set is designed to map between utterances and dialogue act-slot-value triplets. The architecture of our proposed model enables to represent the conceptual meaning from utterances without being limited by different word order or syntactic structure of each language.

We demonstrate that two networks facilitate classifying a correct set of labels for a given utterance by using two inputs at both utterance and inter-utterance level. The CNN architecture of our semantic decoder is appropriate to realize the classification task that fills the slots for corresponding utterances. The LSTM model helps to decode semantic meaning based on previous contextual information of a dialogue. Our model achieves outstanding results on multi-lingual corpora of dialogues, although there is still need to improve our model to be robust and solid enough.

The effectiveness of the proposed classifier model indicates good generality and a reasonable direction for achieving NLU. In future research we extend another topic domain of dialogues on both Korean and English corpora.

References

Lina M. Rojas Barahona, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, Stefan Ultes, Tsung-Hsien Wen, and Steve Young. 2016. Exploiting Sentence and Context Representations in Deep Neural Models for Spoken Language Understanding. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 258–267.

Ronan Collobert, Jason Weston, L´eon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Andre Elisseeff and Jason Weston. 2001. A kernel method for Multi-labelled classification. In *Advances in Neural Information Processing Systems (NIPS), Volume 14*, pages 681-687.

Ferdinand de Saussure. 1916. "Nature of the Linguistics Sign", in: Charles Bally & Albert Sechehaye (Ed.), Cours de linguistique générale, McGraw Hill Education.

Matthew Henderson, Blaise Thomson, and Jason Williams. 2014. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263-272.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long shor-term memory. *Neural computation, Volume 9, Issue 8*, pages 1735–1780.

Minwoo Jeong and Gary Geunbae Lee. 2006. Exploiting Non-local Features for Spoken Language Understanding. In *Proceedings of the COLING/ACL 2006 Main Conference*, pages 412–419.

Seokhwan Kim, Rafael E. Banchs, and Haizhou Li. 2016. Exploring Convolutional and Recurrent Neural Networks in Sequential Labelling for Dialogue Topic Tracking. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics,* pages 963–973.

Yoon Kim. 2014. Convolutional Neural Networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of Interspeech/ICSA*, pages 1045–1048.

Jinseok Nam, Jungi Kim, Eneldo Loza Mencía, Iryna Gurevych, and Johannes Fürnkranz. 2014. Large-scale Multi-label Text Classification - Revisiting Neural Networks. *Machine Learning and Knowledge Discovery in Databases, Springer*, pages 437–452.

Eunjeong L. Park and Sungzoon Cho. 2014. KoNLPy: Korean natural language processing in Python. In *Proceedings of the 26th Annual Conference on Human & Cognitive Language Technology*.

Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing(EMNLP)*, pages 1631–1642.

Ye-Yi Wang, Li Deng, and Alex Acero. 2005. Spoken language understanding. *IEEE Signal Processing Magazine*, pages 16–31.

Ye Zhang and Byron C. Wallace. 2016. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. *arXiv preprint arXiv:1510.03820.*