Utterance Intent Classification of a Spoken Dialogue System with Efficiently Untied Recursive Autoencoders

Tsuneo Kato
Doshisha University

Atsushi Nagai
Doshisha University

Naoki Noda
Doshisha University

Ryosuke Sumitomo
KDDI Research, Inc.

Jianming Wu
KDDI Research, Inc.

Seiichi Yamamoto
Doshisha University

Abstract

Recursive autoencoders (RAEs) for compositionality of a vector space model were applied to utterance intent classification of a smartphone-based Japanese-language spoken dialogue system. Though the RAEs express a nonlinear operation on the vectors of child nodes, the operation is considered to be different intrinsically depending on types of child nodes. To relax the difference, a data-driven untying of autoencoders (AEs) is proposed. The experimental result of the utterance intent classification showed an improved accuracy with the proposed method compared with the basic tied RAE and untied RAE based on a manual rule.

1 Introduction

A spoken dialogue system needs to estimate the utterance intent correctly despite of various oral expressions. It has been a basic approach to classify the result of automatic speech recognition (ASR) of an utterance into one of multiple predefined intent classes, followed with slot filling specific to the estimated intent class.

There have been active studies on word embedding techniques (Mikolov et al., 2013), (Pennington et al., 2014), where a continuous real vector of a relatively low dimension is estimated for every word from a distribution of word co-occurence in a large-scale corpus, and on compositionality techniques (Mitchell and Lapata, 2010), (Guevara, 2010), which estimate real vectors of phrases and clauses through arithmetic operations on the word embeddings. Among them, a series of compositionality models by Socher, such as recursive autoencoders (Socher et al., 2011), matrix-vector model which models the dependencies explicitly (Socher et al., 2012), compositional vector grammar which combines a probabilistic context free grammar (PCFG) parser with compositional vectors (Socher et al., 2013a) and the neural tensor network (Socher et al., 2013b) are gaining attention. The methods which showed effectiveness in polarity estimation, sentiment distribution and paraphrase detection are effective in utterance intent classification task (Guo et al., 2014), (Ravuri and Stolcke, 2015). The accuracy of intent classification should improve if the compositional vector gives richer relations between words and phrases compared to thesaurus combined with a conventional bag-of-words model.

Japanese, an agglutative language, has a relatively flexible word order though it does have an underlying subject-object-verb (SOV) order. In colloquial expressions, the word order becomes more flexible. In this paper, we applied the recursive autoencoder (RAE) to the utterance intent classification of a smartphone-based Japanese-language spoken dialogue system. The original RAE uses a single tied autoencoder (AE) for all nodes in a tree. We applied multiple AEs that were untied depending on node types, because the operations must intrinsically differ depending on the node types of word and phrases. In terms of syntactic untying, the convolutional vector grammar (Socher et al., 2013a) introduced syntactic untying. However, a syntactic parser is not easy to apply to colloquial Japanese expressions.

Hence, to obtain an efficient untying of AEs, we propose a data-driven untying of AEs based on a regression tree. The regression tree is formed to reduce the total error of reconstructing child nodes with AEs. We compare the accuracies of utterance intent classification among the RAEs of a single tied AE, AEs untied with a manually defined rule, and AEs untied with a data-driven split method.

Proceedings of the SIGDIAL 2017 Conference, pages 60–64,
Saarbrücken, Germany, 15-17 August 2017. ©2017 Association for Computational Linguistics

Table 1: Relative frequency distribution of utterance intent classes

intent class tag	freq	sample utterance (translation)
CheckWeather	20.4	How's the weather in Tokyo now?
Greetings	16.5	Good morning.
AskTime	11.3	What time is it now?
CheckSchedule	7.2	Check today's schedule.
SetAlarm	5.7	Wake me up at 6AM tomorrow.
Thanks	3.6	Thank you.
Yes	3.1	Yes.
Goodbye	2.4	Good night.
WebSearch	2.2	Search (keyword)
Praise	2.2	You are so cute.
Time	1.9	Tomorrow.
MakeFun	1.6	Stupid.
GoodFeeling	0.9	I'm fine.
BadFeeling	0.8	I am tired
CheckTemp	0.8	What is the temperature today?
BackChannel	0.7	Sure.
AddSchedule	0.7	Schedule a party at 7 on Friday.
FortuneTeller	0.7	Tell my fortune today.
Call	0.6	Ho.
No	0.6	No way.

freq. : relative frequency distribution in percent.

2 Spoken Dialog System on Smartphone

The target system is a smartphone-based Japanese-language spoken dialog application designed to encourage users to constantly use its speech interface. The application adopts gamification to promote the use of interface. Variations of responses from an animated character are largely limited in the beginning, but variations and functionality are gradually released along with the use of the application. Major functions include weather forecast, schedule management, alarm setting, web search and chatting.

Most of user utterances are short phrases and words, with a few sentences of complex contents and nuances. The authors reviewed ASR log data of 139,000 utterances, redifined utterance intent classes, and assigned a class tag to every utterance of a part of the data. Specifically, three of the authors annotated the most frequent 3,000 variations of the ASR log, which correspond to 97,000 utterances i.e. 70.0 % of the total, redefined 169 utterance intent classes including an *others* class, and assigned a class tag to each 3,000 variations.

Frequent utterance intent classes and their relative frequency distribution are listed in Table 1. A small number of major classes occupy more than half of the total number of utterances, while there are a large number of minor classes having small portions.

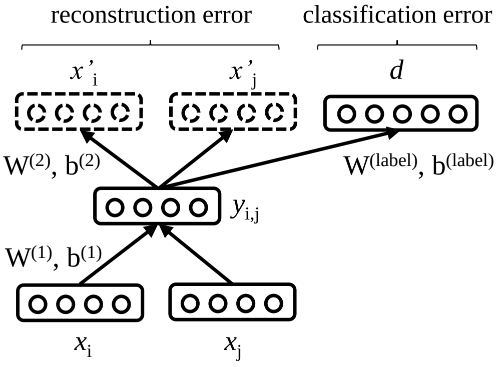

Figure 1: Model parameters and error functions of the recursive autoencoder

3 Intent Class Estimation based on Untied RAE

3.1 Training of Basic RAE

Classification based on RAE takes word embeddings as leaves of a tree and applies an AE to neighboring node pairs in a bottom-up manner repeatedly to form a tree. The RAE obtains vectors of phrases and clauses at intermediate nodes, and that of a whole utterance at the top node of the tree. The classification is performed by another softmax layer which takes the vectors of the words, phrases, clauses and whole utterance as inputs and then outputs an estimation of classes.

An AE applies a neural network of model parameters: weighting matrix $W^{(1)}$, bias $b^{(1)}$ and activation function f to a vector pair of neighboring nodes x_i and x_j as child nodes, and obtains a composition vector $y_{(i,j)}$ of the same dimension as a parent node.

$$y_{(i,j)} = f(W^{(1)}[x_i; x_j] + b^{(1)}) \tag{1}$$

The AE applies another neural network of an inversion which reproduces x_i and x_j as x'_i and x'_j from $y_{(i,j)}$ as accurately as possible. The inversion is expressed as equation (2).

$$[x'_i; x'_j] = f(W^{(2)} y_{(i,j)} + b^{(2)}) \tag{2}$$

The error function is reconstruction error E_{rec} in (3).

$$E_{rec} = \frac{1}{2} \| [x'_i; x'_j] - [x_i; x_j] \|^2 \tag{3}$$

The tree is formed in accordance with a syntactic parse tree conceptually, but it is formed by greedy search minimizing the reconstruction error in reality. Among all pairs of neighboring nodes at a time, a pair that produces the minimal reconstruction error E_{rec} is selected to form a parent node.

Here, the AE applied to every node is a single common one, specifically, a set of model parameters $W^{(1)}$, $b^{(1)}$, $W^{(2)}$ and $b^{(2)}$. The set of model parameters of the tied RAE is trained to minimize the total of E_{rec} for all the training data.

The softmax layer for intent classification takes the vectors of nodes as inputs, and outputs posterior probabilities of K units. It outputs d_k expressed in equation (4).

$$d_k = f(W^{(label)}y + b^{(label)}) \qquad (4)$$

The correct signal is one hot vector.

$$t = [0, \dots, 0, 1, 0, \dots, 0]^t \qquad (5)$$

The error function is cross-entropy error E_{ce} expressed in (6).

$$E_{ce}(y, t) = -\sum_{k=1}^{K} t_k \log d_k(y) \qquad (6)$$

Figure 1 lists the model parameters and error functions of RAE. While AE aims to obtain a condensed vector representation best reproducing two child nodes of neighboring words or phrases, the whole RAE aims to classify the utterance intent accurately. Accordingly, the total error function is set as a weighted sum of two error functions in equation (7).

$$E = \alpha E_{rec} + (1 - \alpha)E_{ce} \qquad (7)$$

The training of RAE optimizes the model parameters in accordance with a criterion of minimizing the total error function for all training data.

3.2 Rule-based Syntactic Untying of RAE

To relax the difference of the nonlinear operation depending on types of nodes, we designed a rule to switch two AEs depending on types of two child nodes manually. At the leaf level of a tree, most of words are nouns, while a sentence or a phrase is composed of a predicate with a subject or an object or a complement. The operation of vectors between words and noun phrases, and that between phrases and clauses are assumed to differ considerably. Hence, the manual rule switches two AEs, one for words and noun phrases, and the other for phrases and clauses. Along a tree, the AE for words and noun phrases is applied at lower nodes around leaves, and the AE for phrases and clauses is applied at upper nodes close to the root node.

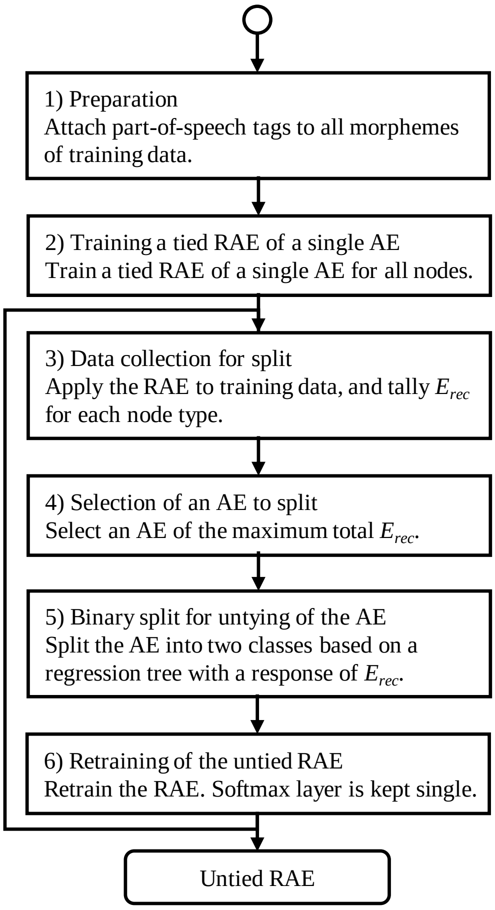
1) Preparation
Attach part-of-speech tags to all morphemes of training data.
2) Training a tied RAE of a single AE
Train a tied RAE of a single AE for all nodes.
3) Data collection for split
Apply the RAE to training data, and tally E_{rec} for each node type.
4) Selection of an AE to split
Select an AE of the maximum total E_{rec}.
5) Binary split for untying of the AE
Split the AE into two classes based on a regression tree with a response of E_{rec}.
6) Retraining of the untied RAE
Retrain the RAE. Softmax layer is kept single.
Untied RAE

Figure 2: Procedure for training RAE of multiple AEs with data-driven untying

The node type is determined as follows. At leaf nodes, every word of a sentence is given a part-of-speech tag as a node type by Japanese morpheme analyzer (Kudo et al., 2004). The number of tags is set at 10. At upper nodes, the node type is determined by the combination of node types of two child nodes. A look-up table of the node type is defined on the basis of Japanese grammar. Another look-up table determining which AE to apply on the basis of the node type is defined as well.

3.3 Data-driven Untying of RAE

To obtain a more effective untied RAE, we designed a training method including data-driven untying of RAE. The method is based on sequentially splitting an AE with regression trees to reduce the total reconstruction error E_{rec}. Specifically, the method splits an AE into two on the basis of a re-

Table 2: Precision, recall, and accuracy of utterance intent classification of 65 classes

.

method	training set			test set		
	prec.	recall	**acc.**	prec.	recall	**acc.**
(1) Cosine similarity of bag-of-words (BoW)	-	-	-	76.0%	74.2%	**85.1%**
(2) Tied RAE based on random word vectors	37.2%	33.2%	**70.6%**	32.0%	65.6%	**66.4%**
(3) Tied RAE based on word2vec vectors	81.2%	78.8%	**88.7%**	74.7%	70.5%	**82.7%**
(4) RAE of two AEs untied by manual rule	65.9%	68.3%	**88.1%**	63.0%	62.5%	**84.0%**
(5) RAE of two AEs untied by data-driven split	80.3%	79.8%	**91.3%**	72.4%	72.3%	**85.6%**
(6) RAE of three AEs untied by data-driven split	73.9%	75.2%	**90.3%**	70.8%	67.9%	**84.8%**

gression tree with the response of the reconstruction error E_{rec}, and optimizes the model parameters of split AEs alternatively.

Figure 2 shows the procedure. The procedure starts with giving a part-of-speech tag to every word of a sentence. While forming a tree, a unique node type is given according to the node types of child nodes. To be precise, a new node type is given to an unseen combination of node types of two child nodes, whereas the same node type is given when the combination of node types has been seen before.

Initially, a single tied AE for all node types is trained. Applying the AE to all training data, reconstruction error E_{rec} is tallied for each node type. Then, a class of all node types is split into two classes based on a regression tree of CART (Breiman et al., 1984) with the response of E_{rec}. The predictor variables are the node types of the left and right child nodes. Then, the AEs are retrained with L2 regularization after every binary split. Note that the softmax layer is kept single in order not to make the generated vector space completely different.

4 Experiments

4.1 Experimental Setup

An experiment of utterance intent classification was conducted with the annotated data described in Section 2. The number of classes was reduced to 65 by merging classes with few pieces of data with a similar class or into the *others* class. Considering the balance of frequent utterances and less-frequent ones, the frequencies of utterances were smoothed by applying a square root function. The numbers of utterances in the training and test sets were 7,833 and 870, respectively. The ratio of unknown utterances in the test set was 15 percent.

4.2 Conditions of Experiments

Two types of word vectors, ramdom word vectors and word2vec vectors, were compared as the minimal elements of a tree. A total of 1.08 million word2vec vectors were trained with Japanese wikipedia texts of 1.1 billion words. The dimension of the vectors was fixed at 100. The word2vec vectors were trained by using skip-gram mode on the basis of results of preliminary experiments.

Three types of RAE, that is, a single tied AE, two AEs untied by the manual rule, and multiple AEs untied by the data-driven split, and a baseline method of cosine similarity of bag-of-words were evaluated.

4.3 Experimental Results

Table 2 shows the precision, recall, and accuracy of the classification for the training and test sets. The baseline method (1) showed relatively high performance, because the test set randomly chosen in consideration of the smoothed frequencies contained many known utterances and words seen in the training set. The tied RAE based on word2vec vectors (3) showed significantly better performance than the tied RAE based on random word vectors (2). While the RAE of two AEs untied by a manual rule (4) made a slight improvement, the RAE of two AEs untied by data-driven split (5) made more improvement. The resulting split was not simple, but one of the two AEs was to add a modifier, roughly speaking. However, the RAE of three AEs untied by data-driven split (6) showed a fall. We believe that the RAE was probably overlearned with thousands pieces of training data.

5 Conclusions

RAE was applied to utterance intent classification of a smartphone-based Japanese-language spoken dialogue system. To improve the classification accuracy, we examined the RAE of multiple AEs untied by a manual rule and RAEs of multiple AEs untied by data-driven split.

Comparing the untied RAEs of two AEs between the manual rule and data-driven split, the AEs untied by the data-driven split showed better accuracy. This means that splitting AEs based on a regression tree with the response of the reconstuction error is effective to some extent.

Reducing the model parameters effectively to circumvent overlearning, and utterance intent classification with more variations of utterances are future work.

References

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. 1984. *Classification And Regression Trees*. Chapman & Hall CRC.

E. Guevara. 2010. A regression model of adjective-noun compositionality in distributional semantics. *Proc. of the 2010 Workshop on Geometrical Models of Natural Language Semantics 2010* pages 33–37.

D. Guo, G. Tur, W. Yih, and G. Zweig. 2014. Joint semantic utterance classification and slot filling with recursive neural networks. *Proc. Spoken Language Technology Workshop 2014* pages 266–267.

T. Kudo, K. Yamamoto, and Y. Matsumoto. 2004. Applying conditional random fields to japanese morphological analysis. *Proc. EMNLP 2004* pages 230–237.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. 2013. Efficient estimation of word representation in vector space. *arXiv: 1301.3781* .

J. Mitchell and M. Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science* 34(8):1388–1429.

J. Pennington, R. Socher, and C. D. Manning. 2014. Glove: Global vectors for word representation. *Proc. EMNLP 2014* pages 1532–1543.

S. Ravuri and A. Stolcke. 2015. Recurrent neural network and lstm models for lexical utterance cllasification. *Proc. Interspeech 2015* pages 135–139.

R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng. 2013a. Parsing with compositional vector grammars. *Proc. ACL 2013* pages 455–465.

R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. *Proc. EMNLP 2011* pages 151–161.

R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. 2013b. Recursive deep models for semantic compositionality over a sentiment treebank. *Proc. EMNLP 2013* pages 1631–1642.

R. Socher et al. 2012. Semantic compositionality through recursive matrix-vector spaces. *Proc. EMNLP 2012* pages 1201–1211.