複数化した再帰型自己符号化器による音声対話システムの意図クラス推定

加藤 恒夫[1]　長井 敦[1]　野田 直希[1]　住友 亮翼[2]　呉 剣明[2]　山本 誠一[1]

[1]同志社大学　[2]KDDI 総合研究所

tsukato@mail.doshisha.ac.jp

1　はじめに

音声対話システムは，簡便に対話的な検索などができるインターフェースである一方，利用者により，あるいは，同じ利用者でも時により異なる入力表現や，省略を含む表現に対して，利用者の入力の意図を的確に推定できることが求められる．このため，利用者の発声に対する音声認識結果を予め定義した有限個の意図クラスに分類し，適切な応答を返すのに必要な変数を取得するのが通常である．例えば，天気予報の問い合わせの場合，天気予報の意図クラスを推定し，適切な応答を返すために必要な変数として，場所と日時の情報を取得する．

近年，大規模テキストコーパスにおける単語間の共起情報をもとに，単語を比較的低次元の数百次元の実数ベクトルで表現する単語分散表現 [1], [2] の研究，また，その単語分散表現をもとに句や節，文のベクトル表現を獲得する構成性の研究が盛んに行われている．なかでも．Socher らによる再帰型自己符号化器 (Recursive Autoencoder, 以下 RAE と記す) [3]，係り受けの関係を明示的にモデル化した行列・ベクトルモデル [4]，構文解析に文法範疇依存の再帰型ニューラルネットワークを組み合わせたモデル [5]，ニューラルテンソルネットワーク [6] 等の一連の提案は注目されている．

これらの手法の有効性は極性判別や感情識別，言い換え検出のタスクで示されたが，ベクトル表現は文意の獲得を狙いとしているため，音声対話システムの意図クラス推定にも適用可能である [7]．ことに，本研究で対象とするスマートフォン音声対話アプリにおいて語彙のバリエーションは大きいため，従来のシソーラスでは対応しきれない同義語や類義語に対して連続的なベクトル表現により適切な演算ができれば意図クラスの推定精度が改善できると考えられる．

本研究では Socher の RAE[3] を適用した．また，単語とより大きな単位である句や述語項ではベクトル表現の作用の仕方が異なると考え，文法範疇依存の再帰型ニューラルネットワークを導入した [5] に倣い，RAE を複数の自己符号化器で構成されるように拡張した．複数化の手法として，日本語文法知識に基づく手動設定ルールによる手法と，自己符号化器の再現誤差を目的変数として回帰木を用いてデータドリブンに自動連続分割を行う手法の 2 種類を検討する．

2　スマートフォン音声対話アプリ

意図クラス推定の評価対象としたのは，日常生活の中で恒常的に音声入力の使用をねらいとしたスマートフォン上の音声対話アプリである [8]．アプリには，使用を促すためゲーム的な要素が盛り込まれている．アプリは初め応答できる内容が限られているが，対話を繰り返すうちに成長し，様々な応答を返すようになる．主な機能は，天気予報，スケジュール登録・管理，アラーム設定，ウェブ検索，雑談等である．

利用者の発声内容に込み入ったものは少なく，短文や単語の発声が多い．本評価に伴い，サーバに蓄積された利用者の発声ログのうち，約 13 万 9 千件に対して意図クラスを再設計し，タグを付与した．「今何時」，「おはよう」，「こんにちは」，「ありがとう」，「今日の天気は」をはじめ頻度の高い発声内容も含まれるため，頻度順にソートし，上位 3,000 種類に対して筆者 3 名で意図クラス分類作業を行った．発声総数に対するカバー率は 70.0%，約 9 万 7 千発声分である．作業の結果，分類が困難な発声内容を集めた「その他」クラスを含め，169 種類の意図クラスを定義した．

タグを付与した発声についての意図クラスの相対度数分布を表 1 に示す．

表 1: 意図クラスの相対度数分布

意図クラスタグ	割合	発声内容の例
CheckWeather	20.4%	「明日の東京の天気は」
Greetings	16.5%	「おはよう」
AskTime	11.3%	「今何時」
ConfirmSchedule	7.2%	「今日の予定は」
SetAlarm	5.7%	「明日 6 時に起こして」
Thanks	3.6%	「こちらこそありがとう」
Yes	3.1%	「いいよ」
Goodbye	2.4%	「おやすみ」
WebSearch	2.2%	「検索」
Praise	2.2%	「かわいいね」
Time	1.9%	「明日」
MakeFun	1.6%	「ばーか」
GoodFeeling	0.9%	「元気」
BadFeeling	0.8%	「疲れた」
CheckTemperature	0.8%	「今日の気温は」
BackChannel	0.7%	「そうだね」
AddSchedule	0.7%	「明日 5 時からパーティ」
FortuneTeller	0.7%	「今日の運勢は」
Call	0.6%	「おーい」
No	0.6%	「いやだ」

3 複数化した再帰型自己符号化器 (RAE) による意図クラス推定

3.1 RAE の半教師あり学習

Socher の RAE は，単語系列をリーフノードとして，自己符号化器 (Autoencoder, 以下 AE と記す) をボトムアップに繰り返し適用することで木を形成し，中間ノードとして句や節のベクトル表現，ルートノードとして文のベクトル表現を獲得する．さらに各ノードのベクトル表現にもう一つのニューラルネットワークを適用することで，有限個の意図クラスのうちの一つを推定する．

AE は，木の各層で隣り合う 2 つのノードのベクトルに対して重み行列 $W^{(1)}$，バイアスベクトル $b^{(1)}$，活性化関数 f からなるニューラルネットワークを適用し，親ノードの分散表現として 2 つの子ノードのベクトルと同じ次元数のベクトル $y_{(i,j)}$ を獲得する．

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

この $y_{i,j}$ に対して 2 つの子ノードのベクトルをできるかぎり再現する逆変換を構成する．すなわち

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

誤差関数は式 (3) で与えられる再現誤差 E_{rec} である．

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

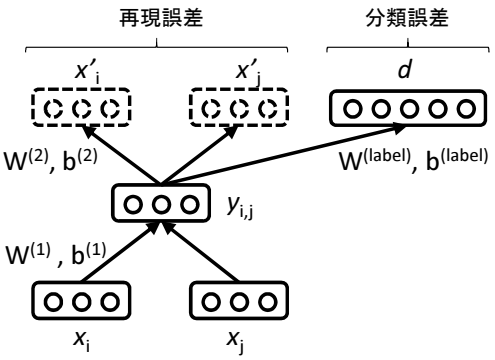

図 1: RAE のモデルパラメータと誤差関数

木の形成は理想的には構文木に従うが，実際には木の末端の隣り合う 2 ノードの組み合わせのうち，再現誤差 E_{rec} が最小になるノードペアをグリーディに探索する．

ここで，全ノードに適用される AE は共通の AE，すなわち共通の $W^{(1)}$ と $b^{(1)}$ が用いられる．RAE の学習は，全学習データに対して E_{rec} の総和を最小化するように $W^{(1)}$，$b^{(1)}$ のモデルパラメータを調整する．

一方，意図クラス分類用のニューラルネットワークは，分類するクラス数 K と同数の出力ユニットを並べ，ノードのベクトル表現 y を入力として，以下の出力 d_k を計算する．

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

正解ラベル t は，各文に対し属する意図クラスに 1 を，それ以外の意図クラスに 0 を与える．すなわち

$$t = [0, \ldots, 0, 1, 0, \ldots, 0]^t \tag{5}$$

誤差関数は再現誤差と同様に式 (6) の 2 乗誤差とした．

$$E_{ce} = \frac{1}{2}|t - d_k|^2 \tag{6}$$

RAE のモデルパラメータと 2 種類の誤差関数を図 1 に示す．AE は，隣り合う単語や句の組み合わせに対して再現性の高い縮約表現を得る一方，RAE 全体としては高精度の意図クラス推定を目的とするため，これらを総合した誤差関数として式 (7) のとおり再現誤差 E_{rec} と分類誤差 E_{ce} の加重和を取る．

$$E = \alpha E_{rec}(x_i, x_j) + (1 - \alpha) E_{ce}(x_i, x_j, t) \tag{7}$$

RAE の半教師あり学習は，全学習データに対する誤差関数の総和を最小化するようにモデルパラメータを調整する．

3.2 手動設定ルールに基づく AE の複数化

ここまでの RAE は，木の全てのノードに対して単一の AE を適用したが，組み合わせる単語の品詞や句・節の種類によって，ベクトル表現の作用の仕方は異なると考えられる．そこで，ノードの意味的な複雑さに応じて，複数の AE を使い分けることとした．

初期検討として，手動で設定したルールにより 2 種類の AE を使い分ける方法を検討した．大語彙の単語辞書においては名詞が大多数を占める．一方，節や文は述語を中心に，主語，目的語，修飾語等が接続する．名詞を中心とする単語および名詞句の集合と，述語を中心とする節や文の集合の分散表現には大きな隔たりがあると考え，その 2 種類の集合用に 2 つの AE を使い分けることとした．RAE により形成される木に沿って考えると，リーフに近いノードには単語・名詞句用の AE が用いられ，ルートに近づくにつれて述語項・文用の AE が用いられることを狙いとしている．

具体的には，リーフノードとなる形態素に mecab を用いて 10 種類の品詞ラベルを付与するとともに，中間ノードについては，2 つの子ノードの品詞ラベルあるいは名詞句・動詞句などの種別の組み合わせにより親ノードの種別を決定するテーブルに従って，ノード種別を決定する．テーブルの規定には「基礎日本語文法」[9] を参照した．さらに，ノード種別により適用する AE を切り替えるテーブルを規定する．

学習時，評価時とも RAE による木の形成においては，各ノードのノード種別に応じて 2 種類の RAE を切り替えて適用する．なお，意図クラス識別用のニューラルネットワークは複数化せずに一種類である．

3.3 自動連続分割に基づく AE の複数化

AE 複数化の効果を高めるために，再現誤差 E_{rec} に基づく自動連続分割を行う．複数の AE の学習手順を図 2 に示す．リーフノードとなる形態素には，手動設定ルールの場合と同様に 10 種類の品詞ラベルを付与する．中間ノードについては，2 つの子ノードの品詞ラベルの組み合わせにより，品詞ラベルに相当するノード種別を一意に定めることができる．最初に，全てのノード種別を含む集合に対して単一の AE を学習する．これを学習データに適用したときの再現誤差 E_{rec} をノード種別毎に集計する．次に，再現誤差 E_{rec} を用いてノード種別の集合を 2 分割する．分割には，E_{rec} を目的変数として CART[10] により回帰木を学習して用いた．説明変数には，左右それぞれの子ノードの品詞ラベル（ノード種別）を用いる．2 分割したノード

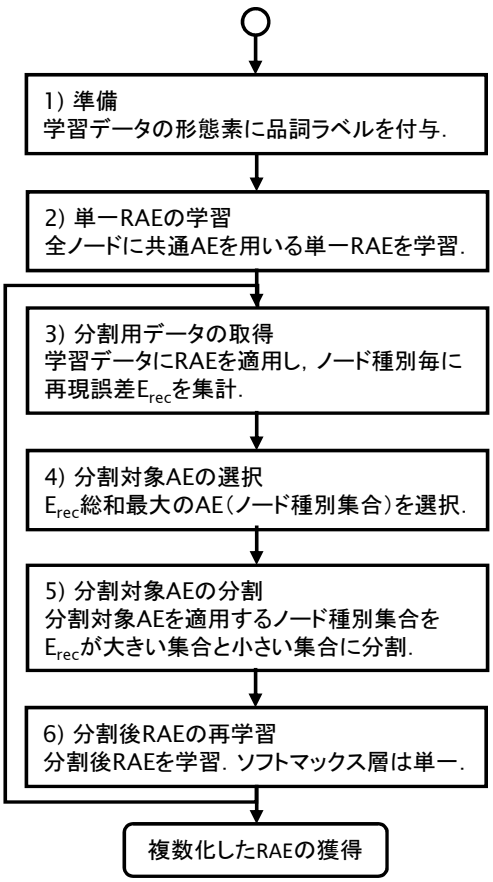

図 2: 自動連続分割による RAE 学習手順

種別集合それぞれの AE を半教師あり学習により学習する．このとき，2 種類の AE によって生成されるベクトル空間がかけ離れないように，意図クラス識別用ニューラルネットワークは分割せずに単一の種類を使用する．また，AE の初期値は分割前のモデルパラメータとし，L2 正則化を掛けて学習を行う．分割後の AE の学習を終えたら，これを学習データに適用し，再現誤差 E_{rec} をノード種別毎に集計し直す．再現誤差の総和が最大となるノード種別の集合を対象にして再び 2 分割を行い，その後はノード種別集合毎の AE の学習と分割を繰り返す．

4 意図クラス推定実験

4.1 学習・評価データ

意図クラスタグを付与した発声ログを用いて意図クラス推定実験を行った．発声内容の種類が少ない意図クラスは，意味の近い意図クラスもしくは「その他」クラスにマージし，65 種類のクラス分類とした．発声内容の頻度の考慮として，高頻度の発声の重みを増や

表 2: 学習データ・評価データに対する 65 意図クラス分類精度

実験条件	学習データ			評価データ		
	prec.	recall	**acc.**	prec.	recall	**acc.**
1) Bag-of-Words	-	-	-	76.0%	74.2%	**85.1**%
2) 乱数単語ベクトルに基づく単一の RAE	37.2%	33.2%	**70.6%**	32.0%	65.6%	**66.4%**
3) word2vec に基づく単一の RAE	81.2%	78.8%	**88.7%**	74.7%	70.5%	**82.7%**
4) 手動設定ルールに基づく AE2 種類の RAE	65.9%	68.3%	**88.1%**	63.0%	62.5%	**84.0%**
5) 自動連続分割に基づく AE2 種類の RAE	80.3%	79.8%	**91.3%**	72.4%	72.3%	**85.6%**
6) 自動連続分割に基づく AE3 種類の RAE	73.9%	75.2%	**90.3%**	70.8%	67.9%	**84.8%**

す一方，低頻度の発声，未知の発声に対する意図クラス推定精度も重視するため，頻度を平方根で平滑化して学習・評価データに反映した．学習データが 7,833 発声，評価データが 870 発声である．評価データに占める未知発声内容の割合は約 15%である．

4.2 実験条件

発声内容の最小構成単位である形態素のベクトル表現として以下の 2 種類を用意した．

1. ランダムベクトルを設定
2. 日本語 Wikipedia テキスト約 11 億語を用いて word2vec で学習した約 108 万語の単語分散表現

ベクトルの次元数は 100 とし，word2vec の学習モードは予備実験の結果 Skip-gram を選択した．また，形態素解析の結果，数詞は一形態素とした．

RAE については，以下の 3 種類を比較した．

1. 単一の AE で構成される RAE
2. 手動設定ルールに基づく単語・名詞句用 AE と述語項用 AE の 2 種類で構成される RAE
3. 自動連続分割によって獲得した複数の AE で構成される RAE

また，ベースライン手法として Bag-of-Words のコサイン類似度に基づく手法も評価した．

4.3 実験結果

学習データ，評価データに対する意図クラス分類の precision，recall，accuracy を表 2 に示す．実験条件 2) の乱数単語ベクトルに基づく単一の RAE に比べ，3) の word2vec に基づく単一 RAE の方が大幅に精度を改善した．複数化する場合には，4) の手動設定ルールに基づく AE2 種類の RAE では殆ど改善していないが，5) の自動連続分割に基づく AE2 種類の RAE では accuracy で約 3%改善した．ただし，6) の AE3 種類の RAE とすると精度が悪化した．過学習が発生していると考えられる．

5 おわりに

スマートフォン音声対話アプリの意図クラス推定に RAE を適用するとともに，句や節の種別に応じて複数の AE を使い分ける手法として，手動設定ルールに基づく手法と自動連続分割に基づく手法を評価した．

手動設定ルールに基づく手法よりも自動連続分割に基づく手法の方が分類精度が高く，再現誤差を目的関数とする回帰木の学習がある程度有効であることがわかった．しかし，学習データが未だ少ないためか，AE の増加に伴う連続的な改善効果までは確認できていない．今後は，日本語の係り受け構造を正確に反映した処理を行うとともに，低頻度の発声内容に対して意図クラスタグを付与し，効果検証を進める予定である．

参考文献

[1] J.Pennington, et al. Glove: Global vectors for word representation. *Proc. of EMNLP 2014*, pp. 1532–1543, 2014.

[2] T. Mikolov, et al. Distributed representation of words and phrases and their compositionality. *Proc. of NIPS 2013*, pp. 3111–3119, 2013.

[3] R. Socher, et al. Semi-supervised recursive autoencoders for predicting sentiment distributions. *Proc. of EMNLP 2011*, pp. 151–161, 2011.

[4] R. Socher, et al. Semantic compositionality through recursive matrix-vector spaces. *Proc. of EMNLP 2012*, pp. 1201–1211, 2012.

[5] R. Socher, et al. Parsing with compositional vector grammars. *Proc. of ACL 2013*, pp. 455–465, 2013.

[6] R. Socher, et al. Recursive deep models for semantic compositionality over a sentiment treebank. *Proc. of EMNLP 2013*, pp. 1631–1642, 2013.

[7] D. Guo, et al. Joint semantic utterance classification and slot filling with recursive neural networks. *Proc. of SLT Workshop 2014*, pp. 266–271, 2014.

[8] X. Xu, et al. Hey peratama: a breeding game with spoken dialogue interface. *Proc. of Mobile and Ubiquitous Multimedia 2014*, pp. 266–267, 2014.

[9] 益岡, 田窪. 基礎日本語文法 －改訂版－. くろしお出版, 1992.

[10] L. Breiman, et al. *Classification and Regression Trees*. Chapman & Hall CRC, 1984.