

Attention-based RNN modelを用いた日本語対話モジュールの意図クラス推定とスロットフィリングにおける未知語対策

長井敦[†]呉剣明[‡]加藤恒夫[†]山本誠一[†]同志社大学[†] KDDI 総合研究所[‡]

ctwb0127@mail4.doshisha.ac.jp

1 はじめに

近年、音声認識技術の発達に伴い、家庭向けに AI スピーカーが発売されるなど、音声対話システムの開発が活発化している。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、入力表現が利用者によって異なったり、同じ利用者でも時によって揺らいだりするため、利用者の意図を的確に捉えるのは簡単ではない。的確な意図推定には意図クラス推定とスロットフィリングの2種類のタスクを高い精度で行うことが求められる。意図クラス推定はあらかじめ定義した意図クラスへの発話分類タスクとして、スロットフィリングはあらかじめ定義したラベルへの単語系列ラベリングタスクとして扱われる。この2種類のタスクは通常、個別のモデルにより別々に処理される。

しかし近年、これらの意図推定タスクを同時に行うニューラルネットワークベースのモデルが報告されている。Liu らにより提案された Attention-based RNN model[1] や joint online RNN model[2] はスロットフィリングのベンチマークテストに用いられる英語データセット ATIS(Air Travel Information System) における性能評価で良い性能を示している。さらに、意図推定を行うにあたって、過去の発話も考慮する Memory Network を構築するモデル [4][5] も報告されている。本稿では、語順の制約が比較的緩い日本語のデータでも、同手法が有効に働くかを検証し、word2vec による汎用単語分散表現 [3] を用いることで、未知語への対策を目指した手法を提案する。

2 意図クラスとスロット

対話システムにおいてエージェントがユーザから課される何らかのタスクを実行する際、必要な情報としては意図クラスとスロットが挙げられる。意図

クラスとはユーザ発話の意図の種類を表すものである。意図クラス推定はあらかじめ定義した意図クラスへの発話分類タスクとして扱われる。スロットとはタスクを実行する際に必要となるパラメータであり、発話意図を構成する要素となる。スロットは一般的に、in/out/begin(IOB) フォーマットで記述され、発話の各形態素に対して与えられる。この時、発話意図を構成しない形態素にはスロット「O」が付与される。このスロットを推定する処理をスロットフィリングと呼び、あらかじめ定義したラベルへの単語系列ラベリングタスクとして扱われる。表1に発話、意図クラス、スロットの例を示す。

表1: 発話, 意図クラス, スロットの例

発話	京都	から	金沢	まで
スロット	B-departure	O	B-destination	O
意図クラス	AskTransportation			

3 モバイルアプリ向け対話モジュール

本稿で、意図クラス推定とスロットフィリングの評価対象としたのは、KDDI 総合研究所が収集したモバイルアプリ向け対話モジュールのログデータである。対話モジュールの主な機能は、天気予報、アラーム、スケジュール登録・管理、連絡先登録・管理、経路検索、アプリ起動などであり、本データにおける意図クラスは基本的に、この機能を示す。頻度上位 20 種類の意図クラスと発話例を表2に示す。また、本データにおけるスロットは地名、日付、氏名、スケジュール内容、アプリ名、などの機能を実行するためのパラメータであり、それぞれのスロットは特定の意図クラスに紐付けられている。発話数は 34210 発話、1 発話あたりの平均形態素数は約 4.9 形態素であり、短い発話が多い。

データを学習用と評価用に 9:1 に分割した場合の未知語率は約 2.9%であり、スロットが付与されている単語に限定すると約 12.4%である。データには 41 種類の意図クラスと 41 種類のスロットが定義されている。意図クラスごとの発話数の分布には偏りがあり、発話数が 100 に満たない意図クラスが半数を占めている。

表 2: 意図クラス毎の発話数の割合と発話例

意図クラス名	割合	発話例
SearchPhonebook	11.4%	お父さんの電話番号
SendMail	6.3%	お母さんにメール
CheckSchedule	6.1%	スケジュール確認
CheckWeather	6.1%	明日の大阪の天気
EditAlarm	6.1%	明日 8 時にアラーム
CallPhone	5.9%	警察に電話
AskPlace	5.9%	地図
AskTransportation	5.8%	大阪駅から京都駅まで
AskAround	5.8%	この辺りのお店
Application	5.8%	テキストアプリ
CheckNews	5.7%	今日のニュース
PlayMusic	5.7%	音楽再生
Horoscope	5.6%	占い
SearchWiki	5.5%	ジnkスの意味
AddSchedule	5.2%	スケジュール登録
AddMemo	1.9%	牛乳買うとメモ
AddPhonebook	1.2%	アドレス帳に登録
ViewPicture	1.1%	写真を見る
Camera	0.7%	カメラ起動
Cancel	0.2%	中止

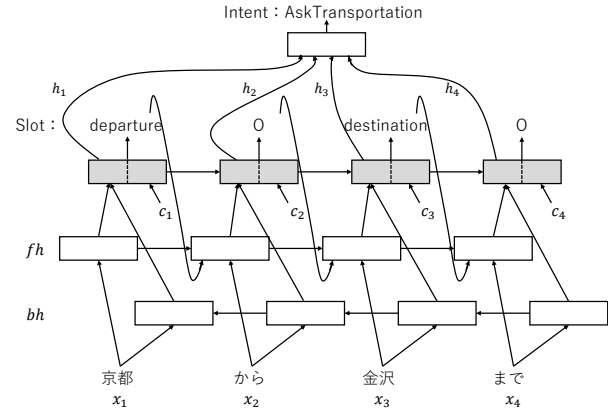


図 1: Attention-based RNN model[1]

$$bh_i = LSTM_{bw}(x_i) \quad (2)$$

$$fh = (fh_1, \dots, fh_T) \quad (3)$$

$$bh = (bh_T, \dots, bh_1) \quad (4)$$

得られた fh と bh を $h_i = [fh_i, bh_i]$ となるように連結し、単語系列を読み込んだ隠れ状態 $h = (h_1, \dots, h_T)$ とする。

$$h = (h_1, \dots, h_T) \quad (5)$$

$$h_i = [fh_i, bh_i]$$

4 手法

4.1 Attention-based RNN model

本研究では Liu らにより提案された Attention-based RNN model をベースとして用いる [1]。このモデルは双方向性 RNN で発話の音声認識結果を形態素単位に読み込み、得られた隠れ層からスロットと意図クラスを推定する。加えて、スロットを推定するには現在注目している単語にフォーカスするために Attention を用いて現時点までの情報を集約したコンテキストベクトル c を学習し用いる。モデルを図 1 に示す。

意図クラス推定とスロットフィリングは入力となる発話、すなわち単語インデックス系列 $X = (x_1, \dots, x_T)$ から出力として意図クラス $Intent$ とスロット系列 $Y = (y_1, \dots, y_T)$ を得る処理である。まず順方向 RNN により単語インデックス系列を読み込み、順方向隠れ状態 $fh = (fh_1, \dots, fh_T)$ を得る。同様に逆方向 RNN により逆順に単語インデックス系列を読み込み、逆方向隠れ状態 $bh = (bh_T, \dots, bh_1)$ を得る。この時 RNN は LSTM(Long short-term memory) ユニットを使用する。

$$fh_i = LSTM_{fw}(x_i) \quad (1)$$

各隠れ状態 h_i は発話全体の情報を含み、特に i ステップ目の単語周辺の情報に焦点を当てている。この隠れ状態 h_i をコンテキストベクトル c_i と連結し、意図クラス推定とスロットフィリングを行う。コンテキストベクトル c_i は隠れ状態 h の加重平均として計算される。 g はフィードフォワードニューラルネットワークである。

$$c_i = \sum_{j=1}^T \alpha_{i,j} h_j \quad (6)$$

$$\alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_{k=1}^T \exp(e_{i,k})} \quad (7)$$

$$e_{i,k} = g(h_{i-1}, c_{i-1}, h_k)$$

4.2 汎用単語分散表現を用いた未知語対策

Attention-based RNN model の入力となる発話中の単語は one-hot ベクトルとして入力され、ランダムに初期値を与えられた単語埋め込み行列を通すことで、予め定義した次元数の実数単語分散表現へと変換される。しかし、これでは学習データに存在しない未知語

や、学習データ中での出現頻度が少ない単語の単語分散表現は類似の文脈で使用される頻出単語に近いベクトルとして適切に学習されず、スロットに未知語が入る場合に、対応できない。そこで、word2vecを用い、あらかじめ大規模コーパスで学習した汎用単語分散表現を、ランダムに与えられていた単語埋め込み行列の初期値と置き換えることとした。これにより、未知語や、学習データ中での出現回数が少ない単語の単語分散表現にもある程度意味を持つ値を与えることができると考えられる。すなわち、未知語にも対応した単語分散表現を獲得できることになる。手法としては、単語埋め込み行列を用いず、大規模コーパスで学習した汎用単語分散表現をそのまま入力の単語分散表現とする手法も考えられるが、本タスクにおいては、学習とともに意図クラスとスロットの推定に向いている単語分散表現の学習を期待するため、単語埋め込み行列の初期値を変更することとした。手法の手順を図2に示す。

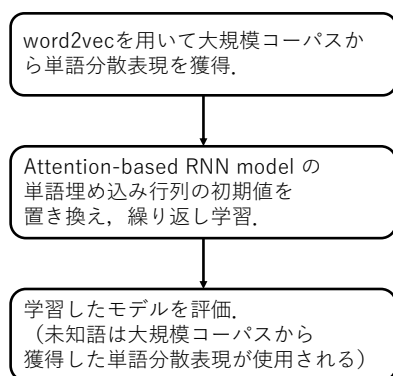


図 2: 提案手法

5 実験

5.1 実験条件

モバイルアプリ向け対話モジュールデータを用いて意図クラス推定とスロットフィリングの実験を行った。学習データは 30789 発話、評価データは 3421 発話である。まず、ベースラインとなる ATIS データとモバイルアプリ向け対話モジュールデータにおいて通常の Attention-based RNN model を用いた場合の評価結果を比較する。使用する ATIS データの言語は英語、意図クラス種類数 22、スロット種類数は 79、1 発話あたりの平均携帯素数は約 11.1 形態素である。また、学習データは 5284 発話、評価データは 587 発話であ

る。次に単語埋め込み行列の初期値をランダムに与えた場合と、word2vec を用いてあらかじめ大規模コーパスで学習した単語分散表現を与えた場合の評価結果を比較することで提案手法の有効性を検証する。この時、大規模コーパスとしては日本語 wikipedia テキスト約 11 億語を使用し、word2vec の学習モードは Skip-gram とする。評価指標は意図クラス分類に関しては accuracy、スロットフィリングに関しては F1-score を用いる。

5.2 実験結果

実験結果を表 3 に示す。ATIS データを使用した際の結果と比べると、日本語であるモバイルアプリ向け対話モジュールデータを使用しても、意図クラス推定、スロットフィリングともに性能の大きな違いは見られなかった。単語埋め込み行列の初期値に word2vec を用いてあらかじめ大規模コーパスで学習した単語分散表現を与えた場合、意図クラス推定精度はわずかに向上したが、スロットフィリング精度はわずかに低下した。

表 3: 意図クラス分類とスロットフィリング精度

評価データ	RNN 初期値	意図クラス accuracy	スロット F1-score
ATIS	random	97.0	95.7
対話モジュール	random	97.75	95.15
	w2v	97.78	95.08

次に、モバイルアプリ向け対話モジュールデータ使用時の単語埋め込み行列の初期値を変更する前後でのスロット毎のスロットフィリング精度の変化をスロットの出現頻度順に表 4 に示す。単語埋め込み行列の初期値に word2vec による汎用単語分散表現を使用したことにより、未知語数が約 83% 減少している。また、スロット毎にスロットフィリング精度が向上しているスロットと、低下しているスロットがある。

6 考察

実験の結果から日本語であるモバイルアプリ向け対話モジュールデータでも Attention-based RNN model が有効に働いた。また、word2vec による汎用単語分散表現を使用した本手法による精度の向上は見られなかったが、スロット毎にスロットフィリング精度を見ると、評価データ中の出現頻度が高い上位 7 スロットのスロットフィリング精度が向上している。特

表 4: スロット毎のスロットフィリング精度の変化

スロット名	出現頻度	未知語数	スロットフィリング精度 F1-score	
SearchPhonebook-name	334	16 → 6	96.9 → 98.5	+1.6
SearchWiki-keyword	225	96 → 7	98.4 → 98.4	0.0
CallPhone-name	206	32 → 2	97.8 → 98.3	+0.5
AskAround-facility	188	14 → 0	94.3 → 94.7	+0.4
AskPlace-position	183	41 → 0	88.5 → 89.6	+1.1
CheckSchedule-datebegin	182	2 → 1	99.2 → 99.5	+0.3
SendMail-name	181	28 → 1	93.7 → 95.6	+1.9
Application-name	181	51 → 34	82.8 → 77.9	-4.9
PlayMusic-keyword	173	32 → 10	93.9 → 94.8	+0.9
AddSchedule-begindate	159	1 → 0	97.8 → 97.8	0.0
AskTransportation-destination	155	18 → 1	96.5 → 96.5	0.0
AskTransportation-departure	152	20 → 1	98.0 → 96.4	-1.6
EditAlarm-date	144	0 → 0	99.3 → 99.6	+0.3
CheckWeather-place	139	2 → 0	98.5 → 97.5	-1.0
Horoscope-date	135	0 → 0	100.0 → 100.0	0.0
Horoscope-constellation	106	1 → 0	99.5 → 99.0	-0.5
CheckWeather-date	104	0 → 0	99.0 → 97.1	-1.9
AddSchedule-subject	67	20 → 2	81.2 → 75.7	-5.5
AddMemo-body	52	0 → 1	92.3 → 92.3	0.0
CheckNews-date	52	23 → 0	99.0 → 98.0	-1.0
All	3201	397 → 66	95.1 → 95.0	-0.1

にスロットフィリング精度の向上しているスロットとして、SearchPhonebook-name, SendMail-name や AskPlace-position 挙げられる。これらのスロットとなる単語は名前や地名などの日本語 wikipedia のデータに多く含まれる単語が多いため、効果的に学習されていると考えられる。対して、出現頻度下位のスロットに関してはスロットフィリング精度の低下しているスロットが多い。特にスロットフィリング精度が大きく低下しているスロットとして、Application-name と AddSchedule-subject が挙げられる。Application-name に該当するアプリの名前は正しく形態素解析されていないものが多く、日本語 wikipedia データ中にもほとんど現れない。今後、実施すべき未知語対策としては、word2vec で学習した単語分散表現を初期値として用いるだけでなく、単語埋め込み行列の再推定における正則化項の調整が有効ではないかと考えられる。

7 おわりに

本稿では、日本語対話モジュールにおける意図クラス推定とスロットフィリングに Attention-based RNN model を用いるとともに、未知語への対応を目指し、word2vec による汎用単語分散表現を用いる手法を評価した。評価結果からスロットによって、適切に学習されるスロットと学習されないスロットがあることが分かった。

今後、性能向上を図るため、Attention-based RNN model における学習時に、単語埋め込み行列にお

ける正則化項の調整を検討している。また、日本語 wikipedia データで学習した単語分散表現が持つ汎用性と、Attention-based RNN model で学習した単語分散表現が持つドメイン指向性を両立させるような単語分散表現を獲得するために、両単語分散表現の同時最適化、もしくは両単語分散表現の変換機能の実現を検討している。

参考文献

- [1] Bing Liu, et al. Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling, Interspeech 2016, pp. 685-689, 2016.
- [2] Bing Liu, et al. Joint online spoken language understanding and language modeling with recurrent neural networks, SIGDIAL 2016, pp. 22-30, 2016.
- [3] T. Mikolov, et al. Distributed Representations of Words and Phrases and their Compositionality. NIPS 26, pp. 3111-3119, 2013.
- [4] Yun-Nung Chen, et al. End-to-End Memory Networks with Knowledge Carryover for Multi-Turn Spoken Language Understanding, Interspeech 2016, pp. 3245-3249, 2016.
- [5] Ankur Bapna, et al. Sequential Dialogue Context Modeling for Spoken Language Understanding, SIGDIAL 2017, pp.103-114, 2017.