

大規模テキストコーパスを用いて未知語に対処する対話システム

安田宜仁 堂坂浩二 マティアス・デネック

日本電信電話(株), NTT コミュニケーション科学基礎研究所

〒619-0237 「けいはんな学研都市」精華町光台 2-4

yasuda@atom.br1.ntt.co.jp

あらまし

従来, 対話システムに追加することは, 対話システムが持っている複雑なデータベースの変更を伴うため困難であった. 本稿では, 未知語をデータベースに追加するのではなく, 未知語とデータベース内の要素を関連づけることによりタスクを達成するような手法を提案する. 適切な関連付けの候補を見つけるために, 質問応答技術を用いる. しかし, すべての未知語をこの手法で解決することは不可能であり, データベースへの追加が不可避な語も存在する. そのような語によって無駄な対話が行われることを避けるため, 我々はさらに未知語の種類分類と, その分類器を提案する. 被験者実験を通じて得られたパラメータに基づいたシミュレーションの結果, 対話を重ねた場合に, より短いやりとり回数で対話を終えられるようになること, および適切に対話を終了できる割合が増加することを確認した.

Handling Unknown Words in Dialogue Systems Using Unannotated Large Text Corpora

Norihito YASUDA Kohji DOHSAKA Matthias DENECKE

NTT Communication Science Laboratories, NTT Corp.

2-4 Hikaridai, Seika-cho, "Keihanna Science City" Kyoto, 619-0237, Japan

yasuda@atom.br1.ntt.co.jp

Abstract

In conventional dialogue systems, handling new words is difficult because their complex back-end databases are hard to change. In this paper, we propose a method to complete tasks by relating unknown words to database entries instead of incorporating them into the database. To find candidates of the proper relationship between an unknown word and database entries, open-domain QA technology is used. However, not all unknown words can be resolved by this method, and additions to the database are required for such words. To avoid unnecessary dialogue by such words, we also propose types of unknown words and a classifier for unknown words. We conducted simulations based on the parameters derived from the preliminary experiments with human subjects, and their result shows that our method enables the system to decrease the number of turns for each dialogues and to increase the ratio of correctly ending dialogues over experienced dialogues.

1 はじめに

従来の音声対話システムでは、主に音声認識の制限とシステムの問題解決能力の制限から、ひとつのシステムが扱うことのできる語彙は百～数千語に限られていた [1, 2, 3]. 音声認識に関しては近年、超大語彙での連続音声認識を実時間でを行うような技術が開発されており、デコーダの速度を理由に語彙を限る理由はなくなってきた。たとえば文献 [4] では、重みつき有限状態トランスデューサを用いて 180 万語の語彙で実時間での認識を実現している。このような技術を前提とすれば、音声認識での未知語（語彙外の語）よりも、システムの知識についての未知語、つまりシステムが問題解決を行う上で対処できないような語の扱いが問題となってくる。これまで音声対話システムは一般的に、バックエンドに高度に構造化されたデータベースを持っており、語彙を追加するのは困難であった。新しい語に関して、データベースで必要な多数の属性についての情報を与える必要があったからである。しかし、タスク遂行の観点からは、データベースへの語彙の追加は必ずしも必要ではないと言える。未知語に対して、その未知語の代替となるような既知のデータベース内の要素を見つけることができれば、システムはその要素を用いてタスクを遂行することができる場合があるからである。ユーザからの目的地と出発地に関する条件を要求として受け付け、列車の乗り換え経路を案内するような、列車情報案内システムを例に説明する。ここで、システムの目的地や出発地に関する知識は駅名に限定されているとする。もし、ユーザがシステムの知識の範囲内の駅名を用いて案内を要求した場合には、従来の対話システムは情報を案内することができる。一方、目的地が「東京タワー」のようなランドマーク名の場合、従来の対話システムではそのランドマーク名がデータベースに列挙されていない限り対応不可能であった。しかし、仮にシステムがそのランドマーク名に対する最寄駅を知ることができれば、その最寄駅に基づいてユーザの要求を処理することにより、「適切な列車の経路を案内する」というタスクを遂行することはできるはずである。

本稿ではこのように、システム問題解決上のための知識を持ち合せていないような単語をデータベース内の既知の要素と関連付けることによって、対話システムが未知語を取り扱う方法について述べる。

関連付けのための適切な候補を見つけるために、本稿では、主に情報検索および自然言語処理の分野で研究されてきた質問応答技術を用いる。質問応答技術とは、大量のテキストコーパスを知識源として用い、自然言語によって尋ねられた質問に対して回答する技術で

ある。システムは、未知語の関連付けのための質問を行って関連付けの候補を得る。質問応答によって常に正しい回答が得られるわけではないので、システムは関連付けが正しいかどうかをユーザに確認する。もし正しい関連付けを得ることができれば、システムはその情報に基づいてタスクを進める。

しかし、すべての未知語をこの方法で解決できるわけではない。そのため、どのような未知語であればデータベース内要素への関連付けにより解決できる可能性があるのかを知ることが重要となる。文脈に合致しているかという観点から、未知語を 2 つの種類に分けることができる。まず、現在の対話の文脈では意味がないような未知語（文脈非合致未知語）がある。これは、ユーザの誤解や、音声認識誤りなどによってシステムへ入力された未知語であると考えられる。このような未知語の場合、ユーザは何か変であることを気付くことができるはずなので、システムからの確認により、問題を避けことができると考えられる。つまり、システムが常に直接確認を行っているとしたら、この種類の未知語をシステムが処理する必要はない。たとえば、列車情報案内システムの目的地として「肩たたき機」という語が入力されたとする。システムが「あなたの目的地は肩たたき機ですか?」といった直接確認を行うことによって、ユーザが問題に気付くことができる。

一方で、対話システムが扱う話題の範囲として不自然でない未知語（文脈合致未知語）の場合は、文脈非合致未知語とは異なり、直接確認を行ったとしても問題を避けることはできない。なぜなら、ユーザが、システムへ入力された語がシステムの能力を越えていることに気付くことは困難であり、そのような場合、確認に対しては単に肯定すると考えられるからである。文脈合致未知語はさらに 2 つのクラスに分類することができる。まず、1 つ目のクラスはシステムのデータベースにその語を追加しない限り本質的にタスクを達成できないような語から構成される。列車情報案内システムの例では、システムの取り扱い範囲外の駅名が該当する。このような駅名が入力された場合、ユーザの要求に応じた情報提供を行うためには、その駅に関する時刻表や料金や接続している列車といった、データベース内の各属性の情報なしにはタスクの達成は不可能である。そのため、このような未知語を要求に含んでいるような対話を継続したとしても、ユーザにとって無益な対話にしかならない。したがって、このような未知語をシステムが一早く察知し、対話の継続を断念することが望ましい。このような語のクラスを、システムが「すみません (Sorry)」と謝って断念すべきことから S クラスと呼ぶ。このクラスの語は、システムの既存知識と同種だけれども、データベースに入っていない

分類 例	文脈非合致 製品名	文脈合致	
		データベースと同種 駅名	データベースと異種 ランドマーク名
取扱い方法	直接確認	S-クラス (断念)	G-クラス (関連付けに挑戦)

表 1: 未知語種別とそれぞれの取扱い方法

い語といえる。つまり、S クラスはデータベース内の既知の要素によって外延的に定義されたクラスであると考えることができる。残りのクラスは、関連付けが成功するのか失敗するのか事前には分からないため、関連付けを試みる価値があると言える。たとえば、もし列車情報案内システムに対して「浅草寺」という語が入力され、最寄駅が「浅草」であることを知ることができれば、「浅草」に基いて情報を提供することができる。このような語のクラスを、システムがさらに「頑張る (Go on)」べき意味があることから G クラスと呼ぶ。以上の未知語の分類について表 1 に示す。

開語彙を扱う手法はこれまでも提案されてきた。一般的な質問応答技術 (文献 [5] など) は、開語彙を扱うことができる。文献 [6, 7] では、対話型の開語彙質問応答が提案されている。これらの手法と異なり、開語彙の質問応答自信を取り扱うのではなく、我々はタスク向型の対話システムでの未知語の問題を取り扱う。また、文献 [8, 9] では、ユーザからの補助による対話システムの知識獲得の手法が提案されている。しかしこれらの手法はユーザの知識とユーザからシステムへの教示に完全に依存している。

2 システム概要

2.1 入力解析と確認

ユーザからの入力は正規表現を用いたパーザによって解析される。限られた解析しか行えない反面、未知語を含んでいたり、断片的な入力などに対しても動作するようなものを一般のプログラマが短期間で作成することが可能である。パーザは、入力とルールを比較し、属性と値の対を出力する。この値が既知語でなければ以後未知語として取り扱う。例として、入力が「渋谷駅から東京オペラシティまで何分で着きますか」であるとすると、この入力に対して、

```
{(&from= “.+" ) から (&to= “.+" ) まで何分で }
というルールが適用されれば、以下の出力を行う。
( to . “東京オペラシティ” )
( from . “渋谷駅” )
( intention . “TIME” )
```

もし解析によって未知語の属性を判定できない場合は、システムはユーザに問合せを行う。たとえば、「行き先は東京タワーですか?」や、あるいはもっと慎重に「目的地はどちらですか?」といった問合せを行う。

2.2 SG 分類器

文脈合致未知語の 2 つのクラスである S クラスと G クラスを分類するために、SG 分類器という分類器を作成した。この分類器は、外延的に定義されたクラスに属する語同士は通常 of 文の中で似たような文脈で出現するという仮定に基いている。注釈なしの大量のテキストコーパスでの bag-of-words モデルを用いる。通常の bag-of-words モデルでは、文やウェブページの特徴を捉えるために、ひとつの文やひとつのウェブページ含まれるすべての単語によってバッグを構成するが、ここでは、各単語の特徴に注目しているため、対象単語の前後の固定された幅に入っている単語によってバッグを構成する。なお、ここでの状況は、網羅的でないクラス中の語の集合 (たとえば、列車情報案内システムのバックエンドデータベース内にある駅名の集合) が与えられ、それらが文書中では正例であると見なせるだけである。そのため、ある閉じた集合内ですべての正例および負例が付与された訓練データを必要とするような分類手法 (たとえば SVM など) を直接は用いることはできない。以下に示す方法により語の重みを計算し、その値が閾値を越えるかどうかによって、当該語のクラスを分類する。

1. 大規模なテキストコーパスを用いて (たとえば新聞など) 固定幅の単語の並びから作成されたバッグをすべて作成する
2. システムデータベースに含まれている語 (既知語) を含んでいるようなバッグをすべて集める
3. ステップ 2. で集めたバッグに含まれるような各単語 t について、その重み w_t を以下の式により求める:

$$w_t = \frac{1}{n_t} \sum_{b \in B} t f_{t,b}$$

ここで、 B は単語 t を含んでいるようなバッグの集合であり、 $tf_{t,b}$ は、単語 t がバッグ b に出現した回数である。また、 n_t は t を含むようなバッグがコーパス中に出現した回数である。

4. 分類器の対象となっている未知語を含むようなバッグをすべて集め、この中に含まれるすべての単語の集合 T とした場合、当該未知語のスコアを以下の式で定義する:

$$\sum_{t \in T} tf_t \times w_t,$$

ここで、 tf_t は単語 t がこれらのバッグの中で出現した回数であり、 w_t はステップ 3. で求めた単語 t の重みである。

5. ステップ 4. のスコアが実験的に定められた閾値よりも大きければ、当該未知語は S クラスと判定し、そうでなければ G クラスと判定する。

2.3 質問応答を用いた知識の関連付け

関連付け対象の候補を見つけるために、システムは質問応答エンジンに対して、未知語の属性に応じた質問を行う。質問応答エンジンは人間によって使われることを前提に作成されているので、自然言語を入力として受け付ける。本手法では、システム自体が質問応答エンジンを用いるので、タスクと未知語の属性に応じて事前に与えられたテンプレートを用いて質問文を生成する。列車情報案内システムの例において、入力が「金閣寺へはどう行けば良いですか」だとする。「金閣寺」は目的地に関しての未知語なので、目的地用のテンプレートの

{ の駅はどこですか? }

を用いて「金閣寺の駅はどこですか?」という質問を生成する。なお、未知語が複合語やフレーズである場合も同様に取り扱う。たとえば「東京タワーの隣の花屋へはどう行けば良いですか」に対しては、システムは「東京タワーの隣の花屋」を解決しようと試みる。

通常、質問応答エンジンは複数の回答を出力する。この回答すべてがシステムが想定している属性に合致するとは限らないので、データベースと比較することによって当該属性では受け付けられないような回答を排除し、データベース内の要素として現れるような回答のみを関連付けの候補とする。こうして得られた関連付けの候補について、システムは、関連付けが正しいかどうかをユーザに対して確認する。もしユーザが正しい関連付けかどうかを知らず、「知りません」といった言葉によって、その旨を伝えてきた場合、システムはそ

れ以上対話を続けることを断念する。もし否定的な応答を得た場合は、別の候補をユーザに対して確認する。もし肯定的な応答を得ることができれば、関連付けられたデータベース要素に基づいて対話を継続する。肯定、否定の応答については、今後の対話で再び同じ未知語が現れた場合に備えて、関係に関する正の関係、負の関係として記録しておく。

2.4 対話制御

対話を重ねるにつれて未知語を効率的に扱えるようにするために、システムはこれまでの対話で獲得した情報を再利用する。未知語を含んだ入力がある度にこれまでに獲得した知識をチェックし、もしその未知語に関して正の情報があれば、関連付けられた既知語に基づいてタスクを継続する。もし負の情報があれば関連付けの候補から除外し、ユーザに対する無駄な確認を避ける。質問応答エンジンからの候補すべてに負の情報が登録されていて、しかもユーザが正しい関連付けの候補を知らない場合には、その時点で対話の継続を断念する。再利用できる正の情報がない場合、2.2 節で述べた SG 分類器を用いてクラスを判定する。もし、分類器が S クラスであると判定した場合はシステムは対話を終了し、それ以上継続できないことをユーザに伝える。G クラスであると判定した場合は、2.3 節に述べたように関連付けを試みる。

3 評価

列車情報案内システムを用いて提案手法を評価した。この列車情報案内システムは時刻表のような固定された情報だけでなく、要求された目的地に関して、料金や時間、あるいは乗り換え回数に応じた適切な経路を計算することができる¹。本列車情報案内システムは関東の 7 都県を扱うことができる²。質問応答エンジンには、日本語の新聞をコーパスに用いた SAIQA [10] を用いた。

対話を重ねることによるシステムの改善を検証するには多数の対話実験を要する。このような実験を被験者を用いて行うのは困難なため、実験は 2 段階に分けて行う。まず少数の対話を予備実験として被験者を用いて行い、次に、多数の対話を模擬ユーザを用いてシミュレーションにより行う。模擬ユーザの動作に必要なパラメータは予備実験での実際の被験者のふるまい

¹バックエンドのシステムにはジョルダン株式会社製「乗換案内サーバ SDK」を用いた。

²「乗換案内サーバ SDK」自体は日本全国の駅を取り扱うことができるが、本稿ではシステム知識が制限されている条件下での実験を行うため、このような設定とした。

種類	(数)	分類器の判定クラス		精度
		G	S	
地名	(52)	46	6	88.5%
大学名	(85)	83	2	97.6%
駅名	(912)	178	734	80.5%

表 2: SG 分類器の精度

を元に決定する。

3.1 予備実験

被験者を用いた実験 被験者を用いた予備実験では、すべての機能を実装したシステムではなく、初期システムを用いた。この実験は、次節で述べるシミュレーションでの模擬ユーザの動作に必要なパラメータを得ることを目的としている。SG 分類器は用いず、すべての未知語は G クラスとして取り扱った。すなわち、ユーザが発話した未知語に対しては必ず関連付けを試みる。実験は音声対話システムを模した Wizard-of-Oz 方式で行った。ただし、ウィザード役が行うのは、ユーザが発話した音声をキーボードでタイプするのみであり、入力に対する処理や応答はすべてシステム自体が行う。システム応答には、NTT サイバースペース研究所メディア処理プロジェクトのテキスト合成システム(開発コードネーム FinalFLUET)を使用した [11]。

被験者はシステムとの対話にとって出発地から目的地までの適切な経路を得る。目的地として駅以外の場所を選択しても良いこと知らせ、具体的には、「過去に観光や出張に訪れた場所」「今後訪れてみたいと思っている観光地」「過去にどのような列車に乗れば良いのか困った区間」「その他行きたいと思っている区間」を自由に選択するように指示した。

222 対話を収集した結果、60 対話は少なくとも 1 つの未知語を含んでおり、全体で 52 の異なる未知語が含まれていた。32% のランドマークについて最寄駅を知っており、22% の目的地はシステムの取り扱い範囲である関東 7 都県以外の駅であった。

SG 分類器の評価 提案した未知語クラスの分類器の動作を確認するため、判定精度を調べた。重み算出のためのテキストとして毎日新聞 11 年分(1991 年から 2001 年まで)を用い、システムデータベースに含まれている単語として、関東 7 都県内の駅名すべてを用いた。単語幅は 5、閾値は 0.01 とした。判定の対象とした未知語は、前節の被験者実験に現れた 52 の地名、次節の実験に用いる 85 の大学名、関東地方以外の 912 の駅名である。このうち、地名と大学名は、関連付けに

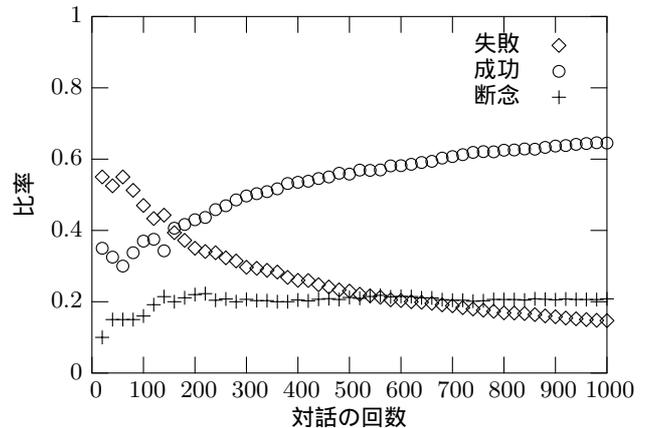


図 1: 対話の終了方法の比率(累積値)

よってタスクを遂行できる可能性があり、G クラスに分類されれば正解であり、駅名は S クラスに分類されれば正解である。結果を表 2 に示す。SG 分類器は単純な構成ではあるが、表から、未知語の種類によらず 8 割以上の精度を実現していることが分かる。

3.2 模擬ユーザとの実験

模擬ユーザとのシミュレーションでは、模擬ユーザは目的地としては駅名あるいは関東地方のほぼすべての大学 85 校の中から選択し、出発地として駅名を選択する。予備実験で被験者が用いた駅名の選択の比率と同様に、模擬ユーザは 22% の駅は関東地方内から選択し、残りは関東以外の地方の駅を選択するとした。予備実験の結果に基づき、目的地として駅名ではなくランドマークを用いた対話のうち 32% の確率で模擬ユーザは最寄駅を知っているとした。上記の条件で対話を 1000 回繰返した。

図 1 に対話の終了方法の比率の累積値を示す。対話の終了方法とは、各対話が成功で終わったのか、失敗で終わったのか、システムが対話の続行を断念したのかの 3 通りである。図より、対話を重ねることによって、失敗で終わった対話の比率が減り、成功および断念の比率が増えていることが分かる。成功で終わった対話が増える主な理由は、ユーザがランドマークを用いて問合せを行う場合に、ランドマークの最寄駅を知らないようなユーザに対しても、これまでに獲得した関連付けに関する正の情報をを用いて情報を提供できる場合によるものである。断念の比率が増えていくのは、関連付けを試みても質問応答エンジンからの回答がすべて関連付けに関する負の情報を持っている場合に、システムから対話を断念することによるものである。また、図 2 に 1 対話をあたりの対話を終了するまでのターン

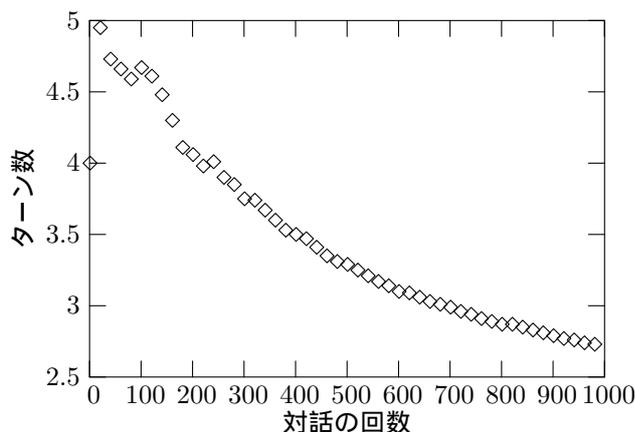


図 2: 1 対話あたりの平均ターン数の推移

数の平均値の推移を示す。図より、システムが対話を重ねることによって、1 対話あたりのターン数を減らすことができている。過去の対話において獲得した関連付けに関する情報を用いて効率的な対話制御を行えるからである。

4 まとめと今後の課題

本稿では、未知語を既知のデータベース内の要素と関連付けることによって対話システムが未知語に対処する方法を提案した。さらに、未知語の中には解決不能な無駄な対話を引き起してしまうものがあるので未知語の分類が重要であることを指摘し、システムが実際に対話中に検出するための分類器を提案した。実際の被験者のふるまいをパラメータに用いた模擬ユーザとの実験によって提案法の有効性を示した。

現在のところ、システムが対話中に獲得していく情報は、ユーザからの応答を信用して構築されている。しかし、現実にはユーザが誤った確認を行うことが発生する。このような、誤ったユーザからの応答に対処することは今後の課題である。

謝辞 質問応答エンジン SAIQA を利用させていただいた、佐々木裕氏、磯崎秀樹氏、前田英作氏に感謝します。分類器に関して有益な議論をしてくださった賀沢秀人氏、平尾努氏、鈴木潤氏に感謝します。

参考文献

[1] Nakano, M., Dohsaka, K., Miyazaki, N., Hirasawa, ichi J., Tamoto, M., Kawamori, M., Sugiyama, A. and Kawabata, T.: Handling Rich

Turn-Taking in Spoken Dialogue Systems, in *Proc. Eurospeech*, pp. 1167–1170 (1999).

[2] Weilhammer, K., Oppermann, D. and Burger, S.: The influence of scenario constraints on the spontaneity of speech. A comparison of dialogue corpora, in *Proc. LREC* (2000).

[3] Chou, W., Zhou, Q., Kuo, H.-K. J., Saad, A., Attwater, D., Durston, P., Farrell, M. and Scahill, F.: Natural Language Call Steering for Service Applications, in *Proc. ICSLP* (2000).

[4] Hori, T., Hori, C. and Minami, Y.: Fast On-The-Fly Composition for Weighted Finite-State Transducers in 1.8 Million-Word Vocabulary Continuous Speech Recognition, in *Proc. ICSLP* (2004), (to appear).

[5] Harabagiu, S., Pasca, M. and Maiorano, S.: Experiments with open-domain textual question answering, in *Proc. 18th COLING* (2000).

[6] Jonsson, A. and Merkel, M.: Some Issues in Dialogue-Based Question-Answering, in *New Directions in Question Answering*, pp. 45–48 (2003).

[7] Hori, C., Hori, T., Isozaki, H., Maeda, E., Katagiri, S. and Furui, S.: Study on Spoken Interactive Open Domain Question Answering, in *Spontaneous Speech Processing and Recognition*, pp. 111–114 (2003).

[8] Purver, M.: Processing Unknown Words in a Dialogue System, in *Proceedings of the 3rd SIGdial Workshop on Discourse and Dialogue*, pp. 174–183, Philadelphia (2002).

[9] Knight, K.: Learning Word Meanings by Instruction, in *Proc. 13th AAAI*, pp. 447–454 (1996).

[10] Sasaki, Y., Isozaki, H., Kokuryou, K., Hirao, T. and Maeda, E.: NTT’s QA Systems for NTCIR QAC-1, in *working notes, NTCIR Workshop 3, Tokyo* (2002).

[11] Takano, S., Tanaka, K., Mizuno, H., Abe, M. and Nakajima, S.: A Japanese TTS System Based on Multi-form Units and a Speech Modification Algorithm with Harmonics Reconstruction, *IEEE Transactions on Speech and Processing*, Vol. 9, No. 1, pp. 3–10 (2001).