2ツイートを用いた対話モデルの構築

東中竜一郎[1]　川前徳章[2]　貞光九月[1]　南泰浩[3]　目黒豊美[3]　堂坂浩二[3]　稲垣博人[1]
[1] NTTサイバースペース研究所
[2] NTTコムウェア株式会社
[3] NTTコミュニケーション科学基礎研究所

1　はじめに

対話モデルとは，対話における発話の遷移を有限状態オートマトンなどの状態と遷移を持つ構造で表したものである．このようなモデルがあると，対話において話者がどのような流れで話をしているのかを分析できたり，対話システムが次にどのような発話を行えばよいかに関する知見を得ることが可能である．対話モデルとしては，隠れマルコフモデル（HMM）が用いられることが多く，たとえば，目黒らは，人同士が傾聴を行っている対話をHMMによりモデル化している[2].

対話のモデル化をするためには元となる対話データが必要である．しかし，被験者を雇って対話データを収集することは，コストが高い．そして，大量に収集できないため，データの内容も特定の話題に限られてしまう．

この解決策として，近年流行しているツイッターのデータを用いる手法が提案されている．ツイッターではユーザ同士がやりとりを行うため，そのデータは対話的な側面を持つ．ツイッターの投稿数は一日に数億とも言われ，非常に量が多く，また，多くのユーザが投稿するため，内容に多様性があり，従来のデータ収集では不可能だった対話データの量と質を確保できる可能性がある．Ritterらは，ツイッターのデータから対話モデルの構築に取り組んでおり[3]，ツイッターのデータにおいて3回以上のやりとりを抽出し，それらを用いてHMMを学習することで対話モデルを構築している．3回以上のやりとりを用いているのは，単純な一問一答ではない対話をモデル化するためには3回以上のやりとりを用いることが直接的だからである．なお，本研究では，やりとりは返信関係にある一連のツイートとし，一連のツイートを「会話」と呼ぶ．

しかしながら，Ritterらの手法を用いて，ツイッターのデータから対話モデルを構築する場合，大きな問題がある．それは，ツイッターデータには，そもそも，会話のデータが非常に少ないことである．そして，その少ない会話のうち，ほとんどが2つのツイートから成り立っている．

表1: ツイッターコーパスの統計情報

	food	sports	all
会話数	63312	37292	1211725
ツイート数	132203	78123	2500918
単語数	2517179	1870382	40098705
ユニーク単語数	74865	75309	452099

表2: Nツイートを含む会話数

N	food	sports	all
2	58269 (92.03%)	34114 (91.48%)	1140201 (94.10%)
3	4565 (7.21%)	2865 (7.68%)	66223 (5.47%)
4	426 (0.67%)	273 (0.73%)	4729 (0.39%)
5	46 (0.07%)	34 (0.09%)	506 (0.04%)
6	6 (0.01%)	5 (0.01%)	62 (0.01%)
7	0 (0.00%)	0 (0.00%)	3 (0.00%)
8	0 (0.00%)	1 (0.00%)	1 (0.00%)
3 ≤	5043 (7.97%)	3178 (8.52%)	71524 (5.90%)

表1に，我々が独自にクロールしたツイッターデータ（ツイッターコーパスと呼ぶ）の統計情報を示す．アクセスレベルはSpritzerである．foodとsportsのカラムについては詳しくは後述するが，食事，および，スポーツに関するツイート集合を表し，allがコーパス全体を表す．全部で1,211,725の会話がある．なお，収集したツイートは全部で95,501,894あったが，そのうち，会話であったものは2,500,918ツイートであることから，会話は全体の2.62%しかないことが分かる．

表2は，会話がいくつのツイートから構成されるかを示しているが，90%以上が2つのツイート（2ツイートと呼ぶ）からなっていることが分かる．このように，ツイッターにおいて会話は非常に少なく，また，そのほとんどが2ツイートからなっている．一般に，対話コーパスには，長い会話が含まれることが望ましいが，そのようなデータは非常に少ないことになる．Ritterらの方法では，非常に数が少ない3ツイート以上の会話（ロングツイートデータとも呼ぶ）からしか

対話モデルが学習できない．これでは，せっかくのツイッターデータのデータ量や多様性を活用しきれていない．

2 提案手法

ツイッターデータにおけるやりとりの大部分を占める，2 ツイートを用い，対話モデルを学習することを考える．ツイッターの会話データで従来使用されていなかったデータを使用できるようになることで，対話モデルの性能を向上させることができ，より質の高い対話分析にも繋がると考えられる．

2.1 基本的な考え方

課題は 2 ツイートのデータから 2 回を超えるやりとりをモデル化することである．ここに，A→B という会話があり，B'→C という 2 つの会話があったとする．ここで，A，B，B' ，C はそれぞれ発話であり，矢印は返信関係を示す．B と B' は内容が近い発話である．近さは，含まれる単語の一致度などから計算できるとする．こうした状況で，B と B' を一つにまとめると，A→{B,B'} →C というロングツイートが構成できる．このように，似た発話をクラスタリングし，2 ツイートからロングツイートを構成していくことで，2 ツイートからでも 2 回を超える会話のモデル化が実現できることが分かる．データをクラスタリングし，状態間の遷移を求めていくことは，HMM における学習過程と同じである．そのため，2 ツイートデータから HMM を学習することになる．

2.2 無限 HMM の学習

HMM の学習には無限 HMM を用いる．無限 HMM はノンパラメトリックベイズの手法の一つであり，状態数がデータ依存で決定されるモデルである [4].

HMM の学習には EM アルゴリズムが用いられることが多いが，文献 [3] でも触れられているように，ベイズ学習を用いた手法の方が性能がよいことが分かっている．また，ツイッターデータは内容が多様であるため，予め状態数を決定してモデル化することは難しい．そこに，無限 HMM を用いる利点がある．

無限 HMM の学習では，会話中のツイートが一つずつ処理される．最初のツイートはまず最初のクラスタ（状態）にアサイン（割り当て）される．最初は一つのクラスタしか存在しない．そして，次のツイート t_i はすでに何らかのツイートがアサインされたクラスタ c_j か新しいクラスタ $c_{j=new}$ に次の確率でアサインされる．

$$\mathrm{P}(c_j|t_i) \propto \mathrm{P}(c_j|c_{t_{i-1}}) \cdot \mathrm{P}(c_{t_{i+1}}|c_j) \cdot \mathrm{P}(t_i|c_j),$$

ここで，c_t はツイート t がアサインされたクラスタを指す．会話においては，ツイートは順序を持っている；t_{i-1} と t_{i+1} はそれぞれ，t_i の会話における直前，直後のツイートを指す．$P(c_k|c_j)$ はクラスタ間の遷移確率を表し，以下のように定義される．

$$\mathrm{P}(c_k|c_j) = \frac{\mathrm{transitions}(c_j, c_k) + \beta}{\sum_{l=1}^{K} \mathrm{transitions}(c_j, c_l) + K \cdot \beta + \alpha},$$

ここで，α はツイートが新しいクラスタにアサインされる度合いを示すハイパーパラメタである．この値が，大きければ大きいほど新しいクラスタが生成されやすくなる．$\mathrm{transitions}(c_j, c_k)$ は c_j から c_k への遷移回数を返す．c_j に含まれるツイートの直後のツイートが c_k にアサインされていると，この回数が多くなる．β はハイパーパラメタである．$\mathrm{P}(t_i|c_j)$ は t_i が c_j から生成される確率であり，以下の式で得られる．

$$\mathrm{P}(t_i|c_j) = \prod_{w \in W} \mathrm{P}(w|c_j)^{\mathrm{count}(t_i, w)},$$

$$\mathrm{P}(w|c_j) = \frac{\mathrm{count}(c_j, w) + \gamma}{\sum_{w \in W} \mathrm{count}(c_j, w) + |W| \cdot \gamma},$$

ここで，W は特徴量の集合であり，$\mathrm{count}(*, w)$ はツイートまたはクラスタにおいて，特徴量 w が何回生起したかを表す．γ はハイパーパラメタである．$\mathrm{P}(t_i|c_{new})$ には一様分布を用いる．新しいクラスタが作られる場合の確率は

$$\mathrm{P}(c_{new}|c_{t_{i-1}}) \cdot \mathrm{P}(c_{t_{i+1}}|c_{new}) \cdot \mathrm{P}(t_i|c_{new}),$$

であるが，そのときの $\mathrm{P}(c_{new}|c_{t_{i-1}})$ と $\mathrm{P}(c_{t_{i+1}}|c_{new})$ は以下のように導出される：

$$\mathrm{P}(c_{new}|c_{t_{i-1}}) = \frac{\alpha}{\sum_{l=1}^{K} \mathrm{transitions}(c_{t_{i-1}}, c_l) + \alpha},$$

$$\mathrm{P}(c_{t_{i+1}}|c_{new}) = \frac{1}{K+1},$$

ここで，$\mathrm{P}(t_i|c_{new})$ には一様分布を用いる．

すべてのツイートを順番に配置した後，ギブスサンプリングによりツイートを再配置する．各ツイートにつき十分な回数のサンプリングが行われたら，収束したとみなし，そのときのツイートのクラスタにおける配置をクラスタリング結果とする．全体の構造が学習された HMM となる．

3 実験

ツイッターコーパスのすべてのデータを用いて HMM を学習するのは計算コストが高いため，今回はその部分集合を実験データとした.「食事」と「スポーツ」に

関するキーワードを含む会話集合を抽出し，それぞれ，Food-Set，Sports-Set とした．これら部分集合の詳細は表 1，表 2 に示した通りである．

Food-Set と Sports-Set のデータから，前節のアルゴリズムにより，無限 HMM を学習した．α, β, および γ には，すべて 0.01 を用いた．特徴量には bag-of-unigrams を用い，文献 [3] や文献 [1] に倣って，2 ツイートデータにおける最頻の 5,000 単語を用いた．ギブスサンプリングのイタレーション数は 1,000 とした．

2 ツイートから作成される対話モデルの有効性を評価するために，ロングツイートから作成した対話モデルと比較を行った．まず，Food-Set と Sports-Set のそれぞれを，2 ツイートデータとロングツイートデータに分けた．そして，ロングツイートデータをランダムに 2 分割した．片方を，ロングツイート学習データ，もう片方を，ロングツイートテストデータと呼ぶ．つまり，各セットは，2 ツイートデータ，ロングツイート学習データ，ロングツイートテストデータの 3 つに分けられたことになる．

評価は，2 ツイートデータから作ったモデル（2 ツイートモデル）と，ロングツイート学習データから作ったモデル（ロングツイートオープンモデル）が，ロングツイートテストデータをどれだけ説明できるかを調べることにより行った．

加えて，ロングツイートテストデータから学習したモデル（ロングツイートクローズドモデル）を使って，自分自身をどれだけ説明できるかも評価した．これは，上限を確かめるためである．また，2 ツイートデータの量によってどのようにモデルが改善するかを確かめるため，2 ツイートデータを 1,000 会話ごとのブロックに区切り，ブロックを一つずつ加えて学習していくことで，性能改善を確かめた．

学習したモデルがどれだけテストデータを説明するかの評価尺度として，対数尤度とケンドールの τ（タウ）を用いた．対数尤度はテストデータを生成する確率であり，テストデータを生成しやすいモデルがいいモデルだと考えられることから採用した．τ は発話の並び替えの尺度であり，会話中のツイートを適切に並び替えられるようなモデルが会話の流れを理解した良いモデルであると考えられることから採用した．τ は具体的に以下のような流れで計算する：

- テストデータにおける会話のそれぞれについて，すべての可能なツイートの順列（順番）を列挙する．
- それぞれのツイートの順番について，対話モデルによって尤度を計算する．
- 最も尤度が高かった順番をそのシステムが最も適切と判断した順番とする．
- 上記適切と判断した順番と，テストデータでの順番（正解）を比較し，下記の式により τ を得る．

$$\tau(R,H) = \frac{n_+(R,H) - n_-(R,H)}{\mathrm{combination}(R)},$$

ここで， R と H はそれぞれ正解と仮説を表し，$n_+(R,H)$ は仮説中のツイートのペアのうち順番が正しかったものの数，$n_-(R,H)$ は仮説中のツイートのペアのうち順番が誤っていたものの数，$\mathrm{combination}(R)$ は仮説中のツイートが取り得るペアの数である．

3.1 結果

図 1 と図 2 はそれぞれ Food-Set と Sports-Set における対数尤度と τ であり，2 ツイートのデータを 1,000 会話ずつ増やした場合に性能がどう変化するかを示している．対数尤度はデータを増やすにつれ，ロングツイートオープンモデルに漸近したり，場合によっては，超えることもある．τ については，データを増やすにつれ，ロングツイートオープンモデルを超えて，ロングツイートクローズドモデルも超えていく傾向にあることが分かる．この結果は，2 ツイートデータの有効性を示すものである．また，どちらのデータでも，τ が急に上昇するポイントが見られる．我々は，これらのポイントで，HMM の構造が 3 ツイート以上を扱えるようになっているのではないかと考えている．

図 3 に，2 ツイートデータの 1,000 会話ずつ増やした場合の，無限 HMM の状態数の推移を示す．前述の通り，無限 HMM では予め状態数を決めず，データを最もよく表す状態数が自動的に決定される．この図から，状態数は 35〜40 程度が良いことが分かる．この状態数は文献 [3] において，性能が飽和する際の状態数に近く，このことは，2 ツイートから学習したモデルが，ロングツイートから学習したモデルと近い可能性を示していると言える．

図 4 に，状態遷移数の推移を示す．ここで，遷移数とは，状態の遷移テーブル（M 状態であれば，$M \times M$ のセルからなる）における，値が 0 でない（遷移が存在する）セルの数を指す．この値が大きくなればなるほど，各状態から多くの状態に遷移していることを示し，複雑なネットワークになっていると言える．図によれば，遷移数は，概ね線形に伸びている．図 3 によると，状態数は一定値で飽和しているが，学習データの増加に応じて，内部ではモデルが複雑になっている様子が見て取れる．

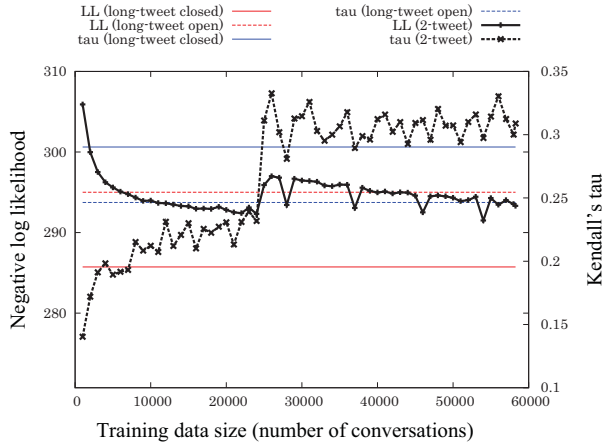

図 1: Food-Set 対する結果：2 ツイートの学習データを変化させたときの，テストデータに対する対数尤度（マイナスをかけており，値が低いほどよい）と τ．赤色の実線と点線はそれぞれロングツイートクローズドモデルとロングツイートオープンモデルの対数尤度．青色の実線と点線はそれぞれ，ロングツイートクローズドモデルとロングツイートオープンモデルの τ．

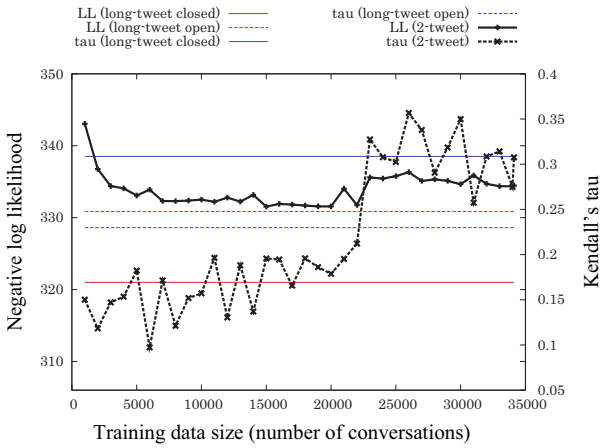

図 2: Sports-Set に対する結果．赤色と青色の実線と点線の意味は図 1 を参照．

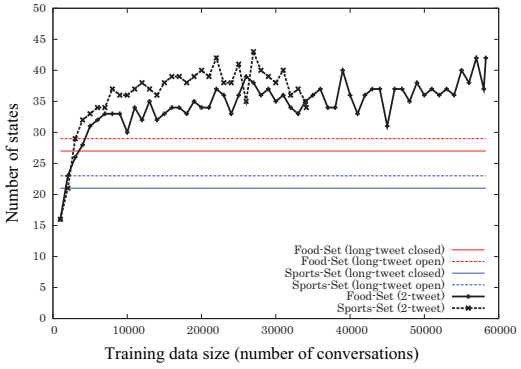

図 3: 状態数の推移：赤色の実線と点線はそれぞれ Food-Set のロングツイートクローズドモデルとロングツイートオープンモデルの状態数．青色の実線と点線はそれぞれ，Sports-Set のロングツイートクローズドモデルとロングツイートオープンモデルの状態数．

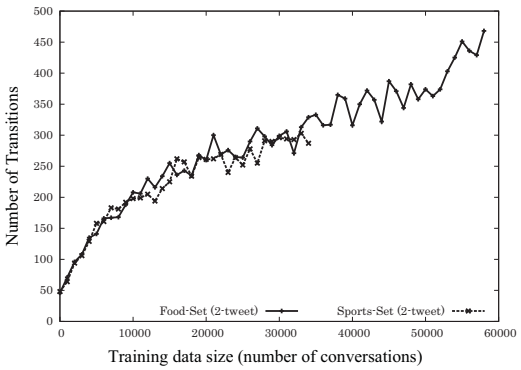

図 4: 遷移数の推移：状態間の遷移テーブルにおける，値が 0 でないセルの数．

4 まとめと今後の課題

本稿では，ツイッターデータから対話モデルの学習をする際に問題となる，長いやりとりが少ないということに対処するために，ワンショットのやりとりである，2 ツイートのみから対話モデルの構築を提案した．2 つのデータセットについて，その有効性を確認することができた．2 ツイートが有効に機能したということは，ツイートのさらに大部分を占める 1 ツイートの利用にも可能性が広がる．たとえば，HMM の構造に大きな影響は与えないと考えられるが，状態の出力分布を，より精緻なものにできる可能性がある．加えて，今後は，ツイッターデータから学習した対話モデルを元に，対話システムの構築に繋げていく予定である．

参考文献

[1] Shafiq Joty, Giuseppe Carenini, and Chin-Yew Lin. Unsupervised approaches for dialog act modeling of asynchronous conversations. In *Proc. IJCAI*, 2011.

[2] Toyomi Meguro, Ryuichiro Higashinaka, Kohji Dohsaka, Yasuhiro Minami, and Hideki Isozaki. Analysis of listening-oriented dialogue for building listening agents. In *Proc. SIGDIAL*, pp. 124–127, 2009.

[3] Alan Ritter, Colin Cherry, and Bill Dolan. Unsupervised modeling of Twitter conversations. In *Proc. NAACL-HLT*, pp. 172–180, 2010.

[4] Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei. Sharing clusters among related groups: Hierarchical Dirichlet processes. In *Proc. NIPS*, 2004.