The University of Osaka
Institutional Knowledge Archive

Title	ユーザの意図理解を目的とした文書データからの知識獲得に関する研究
Author(s)	藤本, 拓
Citation	大阪大学, 2014, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/34562
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

ユーザの意図理解を目的とした
文書データからの知識獲得に関する研究

提出先　　大阪大学大学院情報科学研究科
提出年月　2014年1月

藤本　拓

関連発表論文

1. 学会論文誌発表論文

1. 藤本拓, 原隆浩, 西尾章治郎：自然な発話により操作可能なカーナビゲーションシステムの開発, 電子情報通信学会論文誌, Vol. J96-D, No. 11, pp. 2815–2824 (2013).

2. 藤本拓, 原隆浩, 西尾章治郎：時系列の最適平滑化と動的な語彙集合を考慮した時系列文書に対するトピック解析手法, 電子情報通信学会論文誌, Vol. J96-D, No. 5, pp. 1212–1221 (2013).

3. 藤本拓, 栄藤稔, 金野晃, 秋永和計：階層型 URL 辞書と LDA による Web 閲覧行動のモデル化, 電子情報通信学会論文誌, Vol. J95-D, No. 2, pp. 183–192 (2012).

4. 藤本拓, 鈴木敬, 中山雄大, 竹下敦, ラムザンズルフィカ, ジェントリイクレイグ, ジェインラビ：環境適応型コンテンツ配信におけるコンテンツ正当性保証の実現, 電子情報通信学会論文誌, Vol. J89-B, No. 3, pp. 324–336 (2006).

5. 春本要, 藤本拓, 中野賢, 西尾章治郎：パイプラインリクエストを考慮した Web キャッシングプロキシ上での配送順序変更方式, 情報処理学会論文誌, Vol. 44, No. 11, pp. 2806–2816 (2003).

2. 研究会等発表論文（査読付）

1. Fujimoto, H., Etoh, M., Kinno, A. and Akinaga, Y.: Topic Analysis of Web User Behavior Using LDA Model on Proxy Logs, in *Proceedings of 15th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Vol. 1, pp. 525–536 (2011).

2. Fujimoto, H., Etoh, M., Kinno, A. and Akinaga, Y.: Web User Profiling on Proxy Logs and its Evaluation in Personalization, in *Proceedings of 13th Asia-Pacific Web Conference*, pp. 107–118 (2011).

3. 藤本拓, 中山雄大, 関根和寿, 片桐雅二：携帯電話におけるアプリケーション実行ログ記録方式の設計と実装, 第 20 回コンピュータ・シンポジウム (2008).

4. Fujimoto, H., Nakano, T., Harumoto, K. and Nishio, S.: Response Order Rearrangement on a Caching Proxy for Reducing WWW Latency, in *Proceedings of 17th ACM Symposium on Applied Computing*, pp. 845–851 (2002).

5. 藤本拓, 中野賢, 春本要, 西尾章治郎：キャッシングプロキシにおける WWW コンテンツの効率的な転送方式, 第 12 回データ工学ワークショップ (2001).

3. その他の研究会等発表論文

1. 藤本拓, 栄藤稔, 金野晃, 秋永和計：LDA モデルを利用した Web ユーザプロファイリング方式のコンテンツ推薦への適用に関する研究, 第 3 回データ工学と情報マネジメントに関するフォーラム (2011).

2. 藤本拓, 中野賢, 春本要, 西尾章治郎：キャッシングプロキシにおける配送順序変更手法の理論的性能解析, 情報処理学会第 63 回全国大会講演論文集, Vol. 3, pp. 319–320 (2001).

以上

内容梗概

スマートフォンの普及やクラウドサービスの発展を背景として，Twitter や facebook などのソーシャルメディアがインターネットユーザに急速に普及し，多くのユーザにとってのインターネットは，コンテンツを視聴する手段から自ら発信する手段へと変化した．ユーザが発信することで，インターネット上に蓄積された文書データを分析することで，文書を生成したユーザの意図を理解し，レコメンド，ホットトピック抽出，対話システムといった様々なアプリケーションへと応用する取り組みが盛んに行われている．

本論文は，ユーザの意図理解を目的とした，文書データからの知識獲得技術を確立する．文書データからの知識の獲得は，その目的に応じて必要とされる分析技術が異なる．そこで本論文では，以下に示す二つの軸によって分析技術を分類し，各分類について，ユーザの意図理解を目的とした高精度な知識獲得手法を確立する．第一の軸は，文書分析の粒度である．文書集合を対象した分析か単一の文書を対象とした分析かによって，分析結果の適用先が大きく異なる．第二の軸は，時系列を考慮した分析を行うか否かである．時系列を考慮して文書を分析することで，静的な分析では得られないリアルタイムのホットトピック変化や異常の検知といった知識を獲得可能である．

本論文ではまず，文書集合に対する非時系列分析技術として，トピックモデルによる，ユーザの Web 閲覧行動の高精度なモデル化手法を提案する．トピックモデルは教師無し機械学習の一種であり，雑多な文書集合からトピックの集合であるモデルを生成する．この際，文書を生成したユーザをトピックの多項分布として表現することが可能であるため，文書を生成したユーザの意図理解を行う技術として優れている．ここで，モデルが扱う単語の抽象化を行うことが，モデルの精度向上に大きく寄与することが知られている．そこで提案方式は，Web ページの概念的な階層関係を表す Web ディレクトリ辞書を導入する．これにより，本方式においてトピックモデルの単語に置き換えられるユーザの Web 閲覧行動を抽象化し，最上位概念の行動を抽出する．本論文では，7537 人から得た 4 か月間の Web 閲覧ログを利用した大規模な実験により，提案方式の精度が既存手法を大きく上回ることを示す．また，コンテンツ推薦を例として実アプリケーションにおける

有効性を示し，さらにコミュニティ分析への適用結果を示す．

本論文では次に，文書集合に対する時系列分析技術として，時系列の文書集合の変遷を高精度にモデル化する手法を提案する．文書集合を対象とするため，上述した通り，トピックモデルの適用が有効であると考えられる．しかし，既存のトピックモデルは，時系列文書の分析を行うことが出来ない．これに対して提案方式は，トピックモデルを拡張し，時系列の文書集合を高精度にモデル化する．具体的には，トピックモデルにより各時刻の文書集合をモデル化し，これらを時系列に平滑化することで，時間変化にロバストなモデルを生成する．この際，時系列フィルタであるパーティクルフィルタを適用することで，最適な平滑化係数を導出する．さらに，時々刻々と登場する新語に対応してモデルが扱う語彙集合を動的に拡張する．本論文では，9ヶ月間 1 日 100 万のツイートデータを利用した大規模な実験により，提案方式の精度が既存手法を上回ることを示す．また，ホットトピック抽出を例として，実アプリケーションにおける有効性を示し，さらに 2011 年 3 月 11 日の震災を例として，得られたホットトピックを可視化する．

本論文ではさらに，単一文書に対する非時系列分析技術として，ユーザの発話を単一の文書とし，発話からユーザの意図を高精度に理解する手法を提案する．単一文書に対しては，トピックモデルを含め，教師無しの学習により意図理解を行うことは困難である．そこで，本論文では，別途用意した文書集合から，単一文書の意図理解に適した学習モデルを生成し，これに従って文書を生成したユーザの意図理解を行う．この際，単一文書からのユーザ意図理解に有効と思われる特徴を導入することで，モデルの精度向上を図る．本論文では，3008 人の被験者から収集した発話例を元に学習モデルを構築し，提案方式による意図理解の精度を評価し，高い精度でユーザの発話意図を理解可能であることを示す．さらに，実際に提案方式を組み込んだ対話型カーナビゲーションシステムをサービスとして提供し，実際の対話システムにおいても高い精度で発話意図を理解可能であることを示す．

最後に結論として上記の研究を総括し，課題と今後の展望を示す．

目 次

第 1 章　序章 **1**

1.1　研究背景 1

1.2　研究内容 2

1.3　本論文の構成 7

第 2 章　Web 閲覧行動のモデル化 **9**

2.1　まえがき 9

2.2　LDA による定式化 12

2.3　提案方式 14

2.3.1　URL セッションからの単語セットの抽出 14

2.3.2　Cross-Hierarchical Directory Matching 16

2.4　性能評価 20

2.4.1　データセット 20

2.4.2　評価指標 20

2.4.3　評価結果 23

2.5　ユーザプロファイルの可視化 28

2.6　むすび 32

第 3 章　時系列文書のトピック解析 **35**

3.1　まえがき 35

3.2　時系列モデルの最適な平滑化方式 39

3.2.1　Dynamic Topic Model 39

3.2.2　モデル平滑化のための確率モデル 39

3.2.3　最適な平滑化パラメータの導出アルゴリズム 42

3.3 動的語彙集合に対応した時系列モデル 44
3.3.1 動的語彙集合への対応 44
3.3.2 語彙集合の生成 45
3.3.3 事前分布の更新 47
3.3.4 無限の語彙集合拡大に対応した実装方式 49
3.4 性能評価 50
3.4.1 データセット 50
3.4.2 精度評価 50
3.4.3 オーバヘッド評価 60
3.5 むすび 61

第 4 章 対話システムによる発話意図理解 63
4.1 まえがき 63
4.2 システム構成 66
4.3 タスク判定 69
4.3.1 事前処理 71
4.3.2 タスク判定 72
4.4 実験環境における性能評価 76
4.4.1 実験条件 76
4.4.2 実験結果 78
4.5 対話システムにおける性能評価 81
4.6 むすび 85

第 5 章 結論 87
5.1 本論文のまとめ 87
5.2 今後の研究課題 89

謝辞 91

付 録 A Collapsed Gibbs Sampler によるトピック推定 105

付 録 B　発話文からのクエリ抽出 **107**

付 録 C　ドコモドライブネット **109**

図目次

1.1 取り組み全体像 7

2.1 LDA のグラフィカルモデル 13
2.2 セッションと単語セットの関係 15
2.3 Cross-Hierarchical Directory Matching の例 19
2.4 タイムアウト毎のパープレキシティの変化 23
2.5 各方式におけるパープレキシティの比較 25
2.6 セッションヒット率毎のパープレキシティ 26
2.7 学習データ毎のパープレキシティ 27
2.8 トピック数毎の重要度付ヒット率の比較 28
2.9 各方式における重要度付ヒット率の比較 29
2.10 トピックの 2 次元射影結果 32

3.1 2011 年 11 月から 2012 年 5 月までの Twitter 文書における累積単語数 37
3.2 DTM のグラフィカルモデル 41
3.3 提案方式のグラフィカルモデル 41
3.4 パーティクルフィルタによる最適平滑化パラメータの推定 43
3.5 提案システムの概要 47
3.6 事前分布の更新プロセスの概要 48
3.7 データセット 1 におけるパープレキシティ評価 52
3.8 データセット 1 における $\rho^{(t)}$ の変化 53
3.9 データセット 2 におけるパープレキシティ評価 55
3.10 データセット 2 における $\rho^{(t)}$ の変化 56
3.11 $\kappa = 0.1$ における語彙集合のサイズの変化 57

3.12 全トピック平均の KL ダイバージェンスの時間変化 58
3.13 各トピックにおける KL ダイバージェンスの時間変化 59
3.14 計算時間と語彙集合のサイズの関係 60

4.1 タスク判定機能を備える対話システムの構成 67
4.2 タスク判定の処理概要 70
4.3 素性の生成 73
4.4 コーパス収集の Web ページ 77
4.5 未対応タスクの分布 85

C.1 ドコモドライブネットのユーザインタフェース 110

表目次

1.1 本論文で提案する分析技術の分類 4

2.1 LDA モデルで使用する記法 14
2.2 プロキシログの定義において使用する記法 15
2.3 CHDM の提案において使用する記法 17
2.4 ハイパーパラメータによるパープレキシティの変化 24
2.5 24 のトピックと代表的な単語 30

3.1 3.2 節で使用する記法 40
3.2 3.3 節で使用する記法 46
3.3 LDA のハイパーパラメータによるパープレキシティの変化 54
3.4 パーティクルフィルタのパラメータによるパープレキシティの変化 54
3.5 震災前後の代表的なホットトピックと単語 62

4.1 本論文で扱う対話システムのタスク一覧 1 68
4.2 本論文で扱う対話システムのタスク一覧 2 69
4.3 対話システムが利用する主な辞書 71
4.4 bi-gram で抽出される特徴ベクトル 74
4.5 被験者の属性分布 78
4.6 各特徴ベクトルにおけるタスク判定精度 79
4.7 タスク毎の評価結果 80
4.8 タスク毎の評価結果（コーパス追加後） 82
4.9 サービスログの分類 83
4.10 実システムにおけるタスク毎の評価結果 84

B.1　CRF による学習時の特徴ベクトル 108

第1章　序章

1.1　研究背景

インターネットを取り巻くインフラ環境は，ここ数年で急激に変化した．大きな変化の一つは，LTE や公衆無線 LAN などの高速な通信環境に常時接続された，高性能なスマートフォンの普及である．スマートフォンを介してアクセスすることで，ユーザは，インターネットへ常時接続することが可能となった．もう一つ特筆すべき変化は，ストレージコストの劇的な低減である．大容量かつ安価なストレージサービスやクラウドサービスが提供されることで，もはや蓄積するデータの取捨選択に大きな意味はなく，必要なデータを必要なだけインターネットに蓄積可能となった．

これと同時に，多くのユーザにとってのインターネットの使い方も大きく変化を遂げた．ユーザは，いつでもどこでも必要な情報に手軽にアクセスし，WWW という巨大な情報源からあらゆる情報を入手する．さらには，情報の入手だけではなく，情報を自らインターネットに発信することも一般的となってきた．例えば，ブログや Twitter(https://dtwitter.com/) を通じてユーザは自らのライフログをインターネットに蓄積し，Wikipedia(http://www.wikipedia.org/) やその他多くの口コミ・QA サイトで自らの知識を共有し，facebook(https://www.facebook.com/)，mixi(http://mixi.jp/) などの SNS サービスで交流を深め，LINE(http://line.naver.jp/) でコミュニケーションをとる．これらユーザが自ら情報を発信するサービスを総じてソーシャルメディアと呼ぶ．

ソーシャルメディアはまさに，人が持つ自己表現の欲求を満たすサービスであり，上述したインターネットのインフラ環境の変化に伴い，爆発的に普及した．例

えば，Twitter のユーザ数は，2012 年 12 月現在で 2 億人[1]を突破し，facebook の月間アクティブユーザ数は全世界で 11 億人以上[2]であり，LINE のユーザ数は日本だけで 4700 万人[3]と言われている．

上述したインフラの発展とソーシャルメディアの普及によって，インターネット上に蓄積されるデータも変貌をとげつつある．これまでのインターネットは，限られたコンテンツプロバイダによって，大多数のユーザの関心を惹くために生成されたデータのみが存在した．しかし，ソーシャルメディアでは，インターネットを利用する全てのユーザが情報の発信者となり，ユーザの知識，考え，趣味や嗜好に至るまで，彼らの発信したあらゆるデータがインターネットに蓄積される．このようなデータは万人受けはしないであろうが，確実にユーザと共感する考えや背景を持った新たなユーザを惹きつけ，新たなユーザはこれに呼応して，自らデータを発信するようになる．こうして，様々なユーザが自ら発信した多様なデータが，ますますインターネット上に蓄積されていくこととなる．

1.2　研究内容

1.1 節で述べた通り，ユーザは自らの意志を反映したデータをインターネット上に蓄積する．ここでデータには，文書，画像，動画など様々なものが考えられるが，本論文では特に文書データに着目する．ソーシャルメディアを介して蓄積されるデータの中で，情報として文書データが占める割合は大きい．例えば，Twitter は 140 文字という制限があるものの，ほぼ文書データのみで構成される[4]．その他，ブログ，facebook，Wikipedia などもユーザが蓄積する重要な情報は，文書によって記述される．なお，文書データ以外のユーザデータを対象とした分析として，例えば SNS でのユーザ間のリンク構造を利用したコミュニティ分析 [80] [89] なども行われているが，本論文では文書データ以外の分析は議論しない．

[1]Twitter 公式アカウントのツイートより

[2]http://investor.fb.com/releasedetail.cfm?ReleaseID=802760

[3]2013 年 8 月 21 日の“ Hello, Friends in Tokyo 2013 ”にて発表

[4]3 章で示したデータセットにより計測したところ，99.99%以上のツイートは文書データのみで構成されていた．

インターネットに蓄積された文書データを分析することにより，文書データから新たな知識を獲得する取り組みがさかんに行われている．例えば，奥村のサーベイ [61] によると，ホットトレンド分析，評判分析，コミュニティ分析，ユーザの属性推定，情報信頼性評価など多岐にわたる．これらは単に文書の表層を分析するだけでなく，文書を生成したユーザの意見，目的，趣味・嗜好，プロファイル（性格や属性等）などを文書データから抽出することで，文書データから文書の表層には記述されない新たな知識を獲得している．本論文では，これら文書を生成する際にその背景となったユーザの情報を総じて，ユーザが文書を生成する際の「意図」と呼び，文書データからユーザの意図を獲得することを，意図を理解すると表現する．

文書データからユーザの意図を理解することにより，様々なアプリケーションが考えられる．例えば，ホットトレンド分析 [44] や評判分析 [13] は，ユーザ間での嗜好の片寄りを抽出することにより実現しており，コミュニティ分析 [64] やユーザの属性推定 [69] は，文書を生成したユーザのプロファイルを抽出することにより実現している．このようにインターネット上にユーザが生成した文書データが蓄積されるにつれ，これらを分析することで，文書データを生成したユーザの意図を理解し，文書データから新たな知識を獲得する文書分析技術への注目が高まっている．

文書データの分析技術は，総じて自然言語処理技術と呼ばれる．以下では，自然言語処理技術を，文書データを機械が処理可能な特徴へと変換する汎用的な技術と，分析の目的に応じて適用される応用的な技術に分類する．前者には形態素解析 [32, 56, 73]，構文解析 [9, 10, 69]，照応解析 [3] などが当てはまり，後者には，トピックモデル [7, 27]，文書分類手法 [4, 39, 60, 71]，固有表現抽出 [18, 47] などが当てはまる．前者は 1990 年代までに主に確立され，後者は 2000 年代前半までに確立された．これにより，近年の研究の傾向は，これら自然言語処理技術自体の確立から，インターネット上に蓄積された様々な文書データに対して，いかにこれらの技術を適用するかが中心となってきている．

文書データからの知識の抽出は，その目的に応じて必要とされる分析技術が異なってくる．例えば，あるユーザがある時刻に生成した文書（これを本論文では

表 1.1: 本論文で提案する分析技術の分類

		時系列	
		非時系列分析	時系列分析
分析の粒度	文書集合	分類 1	分類 2
	単一文書	分類 3	分類 4

単一文書と呼ぶ）に対する分析を目的とした場合，構文解析を重点的に行うことで，限られた文書の表層から，少しでも多くの情報を抽出することが求められる．一方で，ある一定の期間に複数のユーザが生成した複数の文書（これを文書集合と呼ぶ）に対する分析を目的とした場合，各単一文書の特徴を捉えるより，トピックモデルなどを利用して文書集合全体の特徴を捉えることが求められる．そこで，本論文では，文書を生成したユーザの意図理解を目的とするという制約を設けた上で，ユーザの意図理解を行うために確立すべき分析技術を分類し，それぞれによってどのような知識を獲得可能かについて議論する．

本論文では，文書の分析技術を表 1.1 で示した二つの軸で分類する．第一の軸は，文書分析の粒度である．文書集合を対象とした分析か，単一文書を対象とした分析かによって，分析結果の適用先が大きく異なるため，軸の一つとして重要であると位置付けた．例えば，前者は，文書集合全体の特徴を捉えることで，文書を生成したユーザ，あるいはユーザ集合（コミュニティ）のモデル化や文書中の話題の抽出など，汎化された知識を生かしたアプリケーションが考えられる．その一方で後者は，生成された文書毎に各ユーザの意図を理解し，即時対応するようなアプリケーションが考えられる．なお，ここで本論文における文書のモデル化とは，雑多な文書集合から機械が処理可能な特徴，あるいは特徴の集合を導出することを言う．例えば，トピックモデルの出力である潜在トピック（以下，単にトピックと呼ぶ）の集合や，文書分類手法で導出される学習モデルは一種のモデルであり，モデルを導出する行為はモデル化となる．

第二の軸は，時系列を考慮した分析を行うか否かである．ある単一文書，あるいは文書集合のスナップショットから，上で述べたようなユーザの意図理解を行

うことを目的とした場合には，非時系列分析で良い．この場合，時系列に蓄積された文書データであったとしても，特に時系列を意識した分析は必要とされない．一方で，時系列を考慮して文書を分析した場合，静的な分析では得られないリアルタイムのホットトピックの獲得や異常の検知が可能である．ここでホットトピックは，文書内で盛り上がりを見せる話題のことを指し，文書を生成したユーザの意図であると位置付ける[5]．特に，多様なユーザが時々刻々と文書データを更新していくソーシャルメディアの分析では，このような時系列分析により，文書を生成したユーザの意図の時系列変化を知ることができるため，分類の軸として重要であると考えた．

本論文では，二つの軸に従って文類された文書データの分析技術を，表に示した通り分類 1 から分類 4 と呼ぶ．本論文の目的は，これら 4 つ分類それぞれについて，ユーザの意図理解を目的とした高精度な知識獲得手法を確立することである．これにより，様々な文書データ分析の目的に応じて，各文書データに最適な手法を適用することが可能となる．

以下，本論文では各分類のデータ分析技術について議論する．分類 1 は，文書集合に対する非時系列分析技術である．特に事前情報が与えられていない場合，ユーザの意図理解を目的とした文書集合の分析には，トピックモデルが有効である．トピックモデルは教師無し機械学習の一種であり，文書に対するいかなる教師データも必要とせず[6]，雑多な文書集合からトピックの集合であるモデルを生成する．また，トピック抽出と同時に，文書を生成したユーザを，トピックの多項分布で表現することが可能であるため，文書を生成したユーザの意図理解を行う技術として優れている．例えば，[51, 63, 74, 83] では，トピックモデルを利用したユーザプロファイリングを行っている．本論文では，まず文書データに対してトピックモデルを適用し，より良い知識獲得を行う手法を提案する．提案方式は，文書に登場する単語間の概念辞書により一連の文書データを抽象化することで，モデルの精度向上を実現する．また，その応用先として，文書集合を生成したユーザのプロファイリングやコミュニティ分析への適用例を示す．

[5] 本論文では，トピックモデルの出力である潜在トピックをトピックと呼び，ホットトピックとは区別する．

[6] トピックモデルを教師データ有りの文書に拡張した研究 [68] も存在する．

本論文では次に，分類 2 について議論する．分類 2 は，文書集合に対する時系列分析技術である．分類 2 は文類 1 と同じく文書集合を対象とした分析技術であるため，分類 1 と同様に，トピックモデルの適用が有効であると考えられる．しかし，既存のトピックモデルは，時系列文書の分析を行うことが出来ない．これに対して [6] では，トピックモデルを拡張し，時系列文書から時間変化にロバストなモデルを生成する技術を提案している．本論文ではこれをさらに拡張し，より精度の良いモデル化手法を提案する．提案方式は，既存手法では考慮されてこなかった，時間的に連続するモデルの平滑化と，新規文書集合に登場する新語をモデルへ動的に組み込むことで，モデルの精度向上を実現する．また，その応用先として，各時刻の文書を生成したユーザ間で盛り上がったホットトピック・ホットワード抽出への適用例を示す.

本論文ではさらに，分類 3 について議論する．分類 3 は，単一文書に対する非時系列分析技術である．単一文書に対しては，トピックモデルを含め，教師無しの学習により意図理解を行うことは困難である．そこで，本論文では，あらかじめ教師モデルを生成するためのデータが存在するという制約を設ける．具体的には，別途用意した文書集合から，単一文書の意図理解に適した教師モデルを生成し，これに従って文書を生成したユーザの意図理解を行う．提案方式は，特に短い文書に特化した特徴量を学習素性として用いることで，精度向上を実現する．また，その応用先として，ユーザの発話を逐次理解する対話システムへの適用例を示す.

分類 4 は，単一文書に対する時系列分析という，分類 2 と分類 3 を組み合わせた問題となる．このような問題に対しては，オンラインの教師有り機械学習 [43] を適用し，入力された単一文書を次々と教師データへ組み込んでいく手法が考えられる．応用先としては，ユーザの発話を逐次理解することで連続した対話を行う対話システムや，ユーザのリアルタイムのコンテンツアクセスに応じて，コンテンツの推薦を行うレコメンドシステムなどが考えられる．ただし，分類 4 については本論文では議論せず，今後の課題として位置付ける．

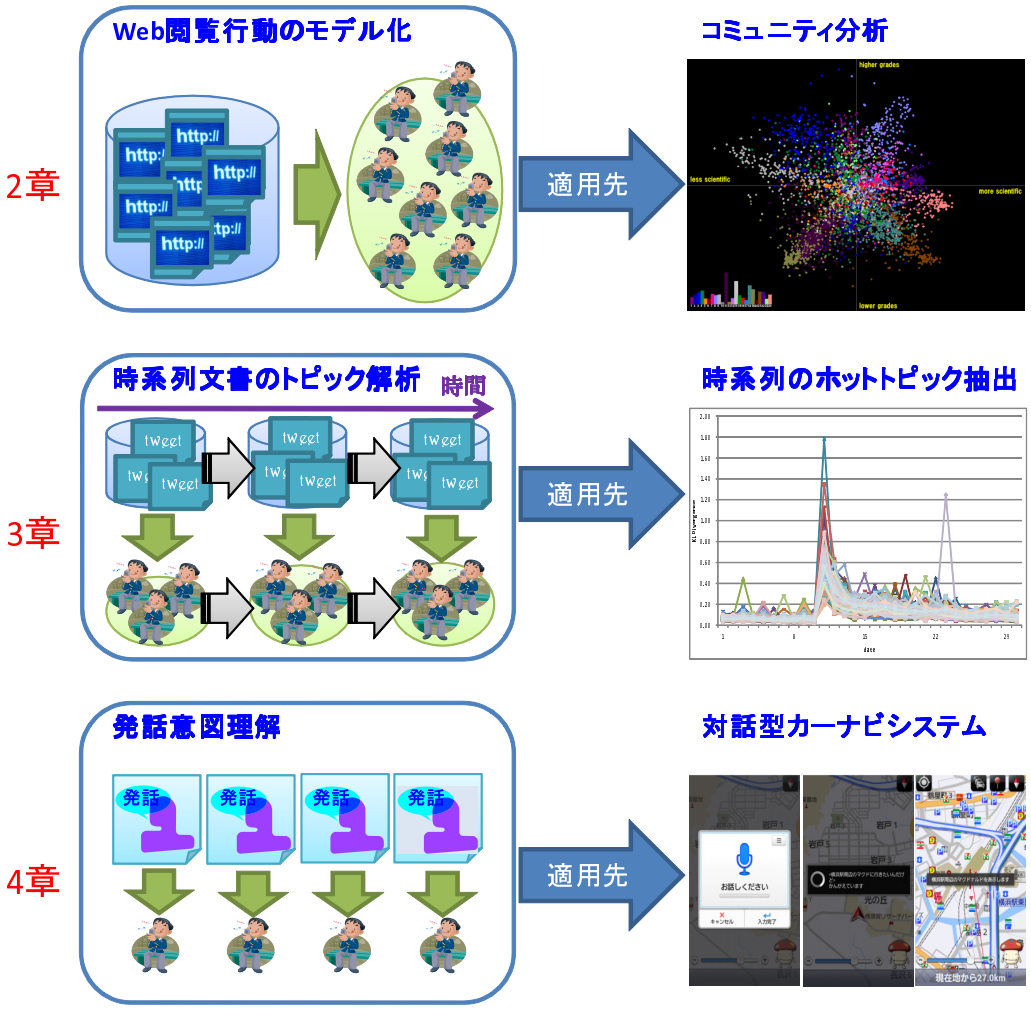

図 1.1: 取り組み全体像

1.3　本論文の構成

本論文は，5 章から構成される．取組の全体像を図 1.1 に示す．本章以降の内容は次の通りである．

第 2 章では，1.2 節で述べた分類 1 の分析技術である，文書集合に対する非時系列分析技術を確立する．具体的には，ユーザの Web 閲覧履歴を文書集合と見なし，トピックモデルによる文書データのモデル化を行う．提案方式は，Web ページ間の概念的な上下関係を表す Web ディレクトリ辞書を構築し，トピックモデルへの

入力データとして，ユーザの意図を最も反映する Web 閲覧行動を抽出する．第 2 章ではさらに，提案方式を 7537 人，4ヶ月間の Web 閲覧履歴に適用し，生成したモデルの最適性を示すと共に，コンテンツ推薦の問題に適用し，実際のアプリケーションにおける有効性を示す．さらに，コミュニティ分析結果の可視化を行う．

第 3 章では，1.2 節で述べた分類 2 の分析技術である，文書集合に対する時系列分析技術を確立する．具体的には，文書データとして，Twitter のように時系列に様々なユーザが蓄積していく文書データを対象とし，トピックモデルを時系列分析に拡張することで，時系列文書のモデル化を行う．提案方式は，モデルの時系列平滑化と動的な語彙集合の導入により，モデルの精度向上を図る．時系列平滑化は，パーティクルフィルタを適用することでモデルの時間変化に最適化する．動的語彙集合は，文書データを key-value 形式で扱う実装により，語彙集合の増加によるシステムの性能劣化を防ぐ．第 3 章ではさらに，提案方式を 9 か月間，1 日 100 万の Twitter データに適用し，生成したモデルの最適性を示すと共に，ホットトピック抽出の問題に適用し，実際のアプリケーションにおける有効性を示す．

第 4 章では，1.2 節で述べた分類 3 の分析技術である，単一文書に対する非時系列分析技術を確立する．具体的には，ユーザが対話システムに逐次行う発話を単一文書と見なし，ユーザの発話を高精度に理解する手法を提案する．提案方式は，事前に収集したユーザの発話例を元に，ユーザの発話を判別する判別モデルを生成し，これによりユーザの発話が何を意図したものであったか，高精度に判別する．この際，モデル化の特徴量として，特に単一文の特徴判別に有効と思われる，品詞選択 N-gram，単語カテゴリ，重み係数の 3 つを導入し，実験によりこれらがいずれも有効であったことを示す．さらに，対話型カーナビゲーションシステムを例として，提案方式を含むシステムを構築し，実際のシステム上でも高い精度でユーザの意図を理解可能であったことを示す．

第 5 章では，本論文のまとめを行い，課題と今後の展望を示す．

なお第 2 章は文献 [19, 20, 23, 24] により公表した結果に基づいて論述し，第 3 章は文献 [21] により公表した結果に基づいて論述し，第 4 章は文献 [22] により公表した結果に基づいて論述する．

第2章　Web閲覧行動のモデル化

2.1　まえがき

本章では，1章で示した分類1の分析技術である，文書集合に対する非時系列分析技術を確立する．具体的には，ユーザのWeb閲覧ログをユーザの意図が反映された文書集合と見なし，Web閲覧を行ったユーザの意図を理解する手法を確立することを目的とする．具体的には，ユーザのWeb閲覧行動のモデル化を行い，さらにユーザプロファイリングやコミュニティ分析への応用を示す．なお，一般的にWeb閲覧ログは時系列に蓄積されていくが，1.2節で述べた通り，ある一定期間に蓄積されたWeb閲覧ログのスナップショットを文書集合として扱い，特に時系列は考慮しないものとする．Guandongらのサーベイ [84] では，Web閲覧行動から得られたユーザプロファイリングやユーザ集合のコミュニティ分析結果の応用として，コンテンツ推薦 [54,59]，Webサイトの最適化 [65]，あるいはe-コマースへの応用 [8] などのアプリケーション例が挙げられている．

本研究では，Web閲覧行動のモデル化にトピックモデルを利用する．トピックモデルは，文書分類分野で広く利用される技術であり，文書をトピックの多項分布で表わす一方で，各トピックを単語の多項分布で表わすモデルである．これまでにWeb閲覧ログの解析にトピックモデルを利用した研究がいくつか報告されている [12,37,49,80,81,83,85]．これらの研究では，Web閲覧ログを文書とし，文書のトピックを各ユーザがWeb閲覧時に持つ潜在的なトピックとし，さらに各ユーザのWeb閲覧行動の単位を単語とした定式化を行うことで，トピックモデルの適用を可能としている．このような定式化においては，ユーザはトピックの多項分布で表現することが可能であるため，トピックによるユーザのモデル化が可能であり，文書を生成したユーザの意図理解を行う技術として優れている．

上記に挙げた先行研究は，Guandong らの研究 [83] を除けば，特定の Web サイトやサービスに適用先を絞り，当該サイトに特化した整形や意味付けが行われたログを利用することで，精度の良いモデルを生成する．例えば，[12,37,81,85] は特定サイトのアクセスログを対象とし，ユーザの Web 閲覧行動をモデル化することで，コンテンツ推薦やターゲット広告へ適用している．また，[49] は検索エンジンのログを対象とし，検索結果のパーソナライズ化へ適用している．さらに，[80] は Twitter のアクセスログを対象とし，Twitter ユーザのプロファイリングを実現している．なお，モデルの生成に際し，2000 年代前半に行われた先行研究 [12,37,49,81,85] では，pLSA（probabilistic Latent Semantic Analysis）[29] を適用しているが，最近の研究 [80,83] では，文書の発生に事前確率分布を導入した，過学習に強いモデルである LDA（Latent Dirichlet Allocation）[7,27] を適用している．

特定のサイトに特化したモデル化は，当該サイトについての行動しかモデル化することが出来ない．例えば，ある EC サイトの Web 閲覧行動をモデル化したとしても，当該サイト外でのユーザの行動をモデル化することは出来ない．一方で本研究では，インターネットを介したあらゆるユーザの Web 閲覧行動を捉えるため，あらゆるサイトやコンテンツへのアクセスが記録されるプロキシログを対象として，Web 閲覧行動のモデル化を行う．モデルの最適性は，Web 閲覧行動予測の精度，つまり単語の発生予測精度（パープレキシティ [7]）により評価する．先行研究では特定のサイトに対する合目的性で評価されてきたが，本論文ではこれについては議論しない．

プロキシログに記録された URL を単語として抽出する場合を考える．トピックモデルは，ユーザの Web 閲覧行動におけるトピックをモデル化するものであるから，閲覧行動の単位となる単語は，ユーザの意図をよりよく表現したものでなければならない．例えば，アプリケーションが自動的にアクセスする URL，ユーザがアクセスしたページに埋め込まれた広告コンテンツやインラインオブジェクトなどのノイズ，ユーザの意図によりアクセスしたページからさらに遷移した詳細ページなど，ユーザの本来の意図とはずれた URL は，単語として抽出するべきではない．すなわち，モデルがよりよい精度を得るためには，ユーザの意図をより良く表現した URL 系列だけを抽出することが大きな課題となる．

文書分類分野では，問題に対して，単語の属性抽出による抽象化を行う辞書構成が有効であると言われてきた [5, 16]．プロキシログの解析においても同様のアプローチが必要であると考えられるが，これまでにこれを考慮した研究は行われていない．ユーザ毎の Web 閲覧ログを文書，行動単位を URL へのアクセスとして文書解析との類似で説明すると，精度の良いモデルを得るために，観測される Web 閲覧ログからどのように単語セットを生成するかが，本章の主題となる．

本研究では，与えられた Web 閲覧ログの URL セッション毎に，最適なモデルを得るための単語セットを生成する手法である，Cross-Hierarchical Directory Matching（CHDM）を提案する．CHDM は，階層型 URL 辞書を利用し，Web 閲覧ログから抽象度の高い URL のみを抽出することで，これを実現する．階層型 URL 辞書は，広範な Web 空間の URL をカバーし，辞書に登録された全ての URL に対して意味的な階層関係を与えるものである．また，抽象度の高い URL とは，辞書においてより上位階層に登録された URL を意味する．例えば，階層型 URL 辞書に，プロ野球ニュースの URL と，その下位概念として，試合結果に関するニュースに関する URL が登録されている場合を考える．この場合，CHDM は，セッション中でアクセスされた URL のうち，より上位に当たるプロ野球ニュースに関する URL を単語として抽出し，下位概念にあたる試合結果に関するニュースに関する URL や，その他の辞書に登録されていない URL は抽出しない．このような辞書は，例えば Yahoo! Japan Directory などの，ディレクトリ型検索エンジンより生成可能である．これにより，単純に Web 閲覧ログに対して LDA を適用する場合と比較して，高精度のモデルを生成可能となる．

以下，本章における主題をまとめる．

1. ユーザの Web 閲覧行動の高精度なモデル化手法である CHDM を提案する．CHDM は，URL の概念的な階層関係を与える階層型 URL 辞書を利用することで，URL セッションに登場する単語を抽象化し，モデル化に最適な単語セットを抽出する技術である．

2. 学生 7537 人の 4ヶ月間に渡る Web 閲覧ログを対象とした，大規模な実験結果を示す．実験では，提案方式によるモデル化の精度をパープレキシティに

より評価する．さらに，ユーザ分類に基づくコンテンツ推薦の問題に適用し，提案方式の実アプリケーションに対する有効性を示す.

3. 上記と同様のデータを利用して，提案方式により得られる学生のプロファイリング結果を可視化し，主観的に評価を行うことで，提案したモデルの有効性を示す.

特定のアプリケーションにチューニングした先行研究 [12, 37, 49, 80, 81, 85] は，特定目的で辞書を構成することにより単語セットを抽出しており，ユーザの広範な Web 閲覧行動をモデル化することは出来ない．また，プロキシログに対してトピックモデルの適用を行っている先行研究としては，Guandong らの研究 [83] があるが，彼らのスコープは，トピックモデルの違いによる性能比較であり，単語生成の手法は対象としていない．これらに対して本研究では，プロキシログから，精度の良いモデルを得るため，いかにして単語セットを生成するかという URL セッション解析手法に着目している点で，あらゆる先行研究と異なっている．

以下では，2.2 節で LDA によるモデルの定式化を行い，2.3 節で URL セッションからの単語の生成手法を提案する．また，2.4 節では，得られたモデルを大阪大学学生 7537 人のプロキシログに適用し，パープレキシティによりモデルの最適性を定量的に評価し，さらに，コンテンツ推薦の問題に適用することで，具体的なアプリケーションにおける提案方式の有効性を示す．また，2.5 節で，モデルから得られる学生のプロファイリング結果を可視化し，主観的に有効性を確かめる．最後に 2.6 節でまとめと課題を述べる．

2.2　LDA による定式化

ユーザの Web 閲覧ログから，ユーザの Web 閲覧行動をモデル化するため，本研究では LDA を利用する．LDA は，文書分類分野において，文書のトピックを抽出するために提案された技術である．LDA は，文書に対して bag-of-words を仮定する．bag-of-words は，文書を登場する単語の並びを無視し，単語の登場頻度によってのみ表現する．この場合，文書は，出現した単語の頻度ベクトルにより表現され

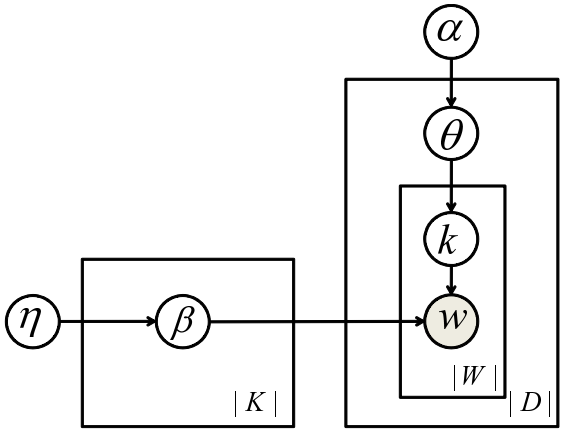

図 2.1: LDA のグラフィカルモデル

る．LDA は，この頻度ベクトルを入力として，各文書をトピックの混合分布で表し，各トピックを単語の出現確率分布で表す．より正確には，トピック集合 K の各要素を k，語彙集合 W の各要素を w，文書集合 $\boldsymbol{D}$ の各要素を $\boldsymbol{d}$ とし，$\boldsymbol{\alpha}$，$\boldsymbol{\eta}$ をディリクレ分布のハイパーパラメータとした場合，図 2.1 に示したようなグラフィカルモデルで表すことが出来る [27]．ただし，$\boldsymbol{\theta}$ は文書ごとのトピック分布であり，$\boldsymbol{\beta}$ はトピック毎の単語分布である．ここでグラフィカルモデルは，各確率変数の依存関係をグラフ構造で記述する確率モデルであり，図では観測変数の背景を色付けしている．すなわち，$\boldsymbol{\alpha}$ によって $\boldsymbol{\theta}$ が決定され，$\boldsymbol{\theta}$ によって $\boldsymbol{k}$ が決定され，$\boldsymbol{\eta}$ によって $\boldsymbol{\beta}$ が決定され，$\boldsymbol{k}$，$\boldsymbol{\beta}$ によって $\boldsymbol{w}$ が決定される．LDA は，上記のような過程で文書が生成されると仮定し，$\boldsymbol{\theta}$，$\boldsymbol{\beta}$ を推定する．例えば，[27] では，Collapsed Gibbs Sampler [50] を利用した推定手法を提案しており，本研究でもこの手法を採用している．詳細は，付録 A で述べる．

Web 閲覧ログの解析に LDA をあてはめる場合，ユーザの Web 閲覧行動にトピックが存在すると仮定する．例えば，ユーザは‘プログラミング’というトピックの元で C 言語の学習サイトへアクセスする．このような仮定を置いた場合，文書をユーザ，単語を URL と置き換えることで，LDA を適用可能となる．すなわち，ユーザの各 URL の閲覧頻度を LDA へ入力することで，各ユーザをトピックの混合分布で表すことが可能となる．この際，各トピックは URL の確率分布で表現され

表 2.1: LDA モデルで使用する記法

記号	定義
$\boldsymbol{W}$	語彙集合 :$\boldsymbol{W} = \{\boldsymbol{w} \mid 1 \leq \boldsymbol{w} \leq \mid \boldsymbol{W} \mid\}$
$\boldsymbol{K}$	トピック集合 :$\boldsymbol{K} = \{\boldsymbol{k} \mid 1 \leq \boldsymbol{k} \leq \mid \boldsymbol{K} \mid$
$\boldsymbol{n_{dw}}$	ユーザの各サイト閲覧回数の集合 $\boldsymbol{D}$ の要素
$\boldsymbol{\theta_{dk}}$	ユーザ毎のトピック分布 $\boldsymbol{\theta}$ の要素
$\boldsymbol{\beta_{kw}}$	トピック毎の単語分布 $\boldsymbol{\beta}$ の要素

る．具体的には，表 2.1 で与えられるパラメータの元で，ユーザが各 URL を閲覧した回数 $\boldsymbol{D}$ を入力として与え，ユーザ毎のトピックの分布 $\boldsymbol{\theta}$，トピック毎の URL の分布 $\boldsymbol{\beta}$ をモデルとして導出する．本研究の目標は，ユーザの Web 閲覧行動を最適にモデル化した $\boldsymbol{\theta}$，$\boldsymbol{\beta}$ を導出することであり，そのための入力 $\boldsymbol{D}$ を生成する．

2.3　提案方式

2.3.1　URL セッションからの単語セットの抽出

提案方式を述べるに先立ち，表 2.2 に従いプロキシログを定義する．図 2.2 に示す通り，プロキシログの各レコードは，ユーザ毎にセッションに分割され，各セッション中でアクセスされた URL は，アクセス順にソートされる．例えば，ユーザ $\boldsymbol{d}$ の $\boldsymbol{i}$ 番目のセッション $\boldsymbol{S_i^{(d)}}$ の $\boldsymbol{j}$ 番目のレコードでは，URL$\boldsymbol{l_{ij}^{(d)}}$ へのアクセスが記録される．

ここでセッションとは，ユーザの一連の Web 閲覧行動をグループ化したものであり，いくつかの既存のアプローチと同様にタイムアウトによって生成される [37, 82, 83, 84, 85]．すなわち，セッション中はユーザは特定の目的によって，Web の閲覧を行い，目的を達成した場合には，次のセッションが開始されるまでに，一定の空き時間（タイムアウト $\boldsymbol{\delta}$）が発生するものとする．

以下では，セッション毎に，LDA への入力となる単語セットを抽出する手法を述べる．すなわち，セッション $\boldsymbol{S_i^{(d)}}$ から，単語セット $\boldsymbol{W_i^{(d)}}$ を生成する．ここで，

表 2.2: プロキシログの定義において使用する記法

記号	定義
$S_i^{(d)}$	ユーザ d の i 番目のセッション
$L_i^{(d)}$	$S_i^{(d)}$ にアクセスされた URL 集合：$L_i^{(d)} = \{l_{ij}^{(d)} \mid 1 \leq j \leq \lvert L_i^{(d)} \rvert\}$. 各要素 $l_{ij}^{(d)}$ は，セッション i の j 番目にアクセスされた URL.
$W_i^{(d)}$	$S_i^{(d)}$ から抽出される単語セット.

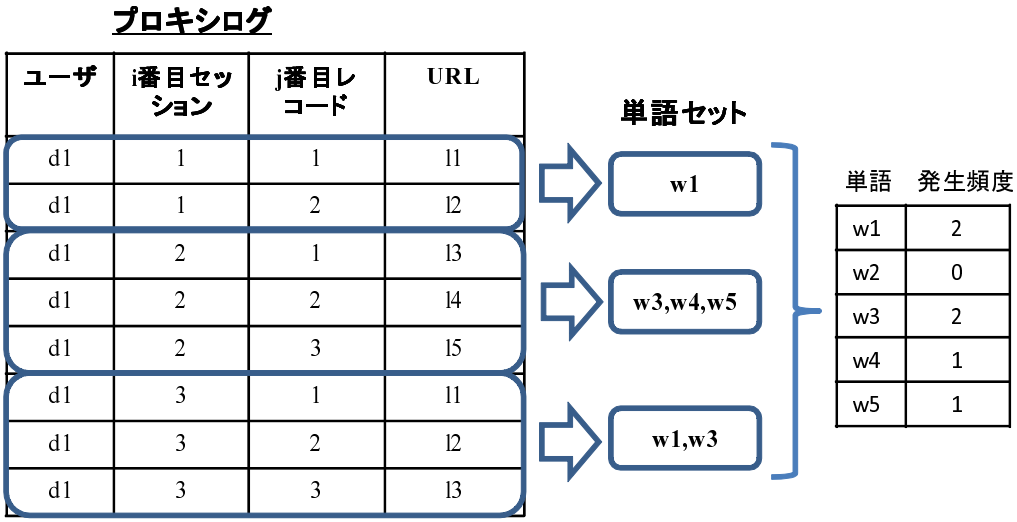

図 2.2: セッションと単語セットの関係

単語は URL ではあるが，各セッション中でアクセスされた URL そのものではなく，アクセスされた URL より生成される新たな URL である．いかにして，単語を生成するかについては，2.3.2 項にて述べる．

図 2.2 に，セッションと生成される単語の関係の例を示す．図は，3 つのセッション 1,2,3 から構成されるあるユーザ d1 のログと，そこから得られる単語セットを示している．例えば，セッション 1 では，l_1 と l_2 の二つの URL へのアクセスが記録され，そこから単語セット w_1 が生成される．また，セッション 2 では，w_3,w_4,w_5 が生成され，セッション 3 では，w_1，w_3 が生成される．これら 3 つの単語セットを合わせることで，最終的にユーザ毎の各単語の発生頻度を導出する．

全てのユーザについて，全てのセッションから単語を抽出したあとは，各ユーザについて，各 URL の閲覧頻度，すなわち入力行列 $\boldsymbol{D}$ の要素 $\boldsymbol{n_{dw}}$ を導出可能である（式 2.1）．ここで，同一セッション内における複数回の同一 URL の閲覧は 1 回と見なし，各 URL が登場したセッション数を以て，閲覧頻度とした．これは，同一セッション内における同一 URL へのアクセスは，Web ページの再読み込みや Web ページに含まれるインラインオブジェクトの取得などが支配的であり，ユーザが意図した閲覧とは見なせないためである．

$$\boldsymbol{n_{dw}} = |\{\boldsymbol{S_i^{(d)}} | \exists \boldsymbol{i} : \boldsymbol{w} \in \boldsymbol{W_i^{(d)}}\}| \tag{2.1}$$

2.3.2 Cross-Hierarchical Directory Matching

高精度なモデルを得るため，各セッションにおいてアクセスされた URL から，辞書を利用した単語の生成を行う．文書分類における従来研究に従えば，単語の生成には，単語の属性抽出による抽象化が有効である [5, 16]．本研究では，セッションからの単語生成においても，同様の効果が得られると仮定し，セッションの中で最も意味的に抽象度の高い，上位概念にあたる URL の集合を抽出し，これを当該セッションから生成される単語のセットとする．セッションから，これら上位概念に抽象化した URL の集合を生成するため，本研究では，Yahoo! Japan Directory [91] を利用し，階層型 URL 辞書を生成する．辞書は，木構造によって階層化されたカテゴリにより構成され，上位の階層ほど，抽象度の高い概念を有し，また各カテゴリには複数の URL が登録されている．例えば，スポーツニュースカテゴリの下には，ワールドカップカテゴリが存在し，各カテゴリには対応する URL が登録される．そのため，登録された URL 間の意味的な階層関係を知ることが可能である．本論文では，上述したような辞書を利用したセッションの抽象化処理を，Cross Hierarchical Directory Matching（CHDM）と呼ぶ．

CHDM の動作を述べる前に，表 2.3 を元に辞書を定義する．辞書は，URL のカテゴリ $\boldsymbol{c_h}$ をノードに持つ木構造であらわされる．カテゴリとは，URL の意味的な概念を表すもので，例えばスポーツニュース，SNS サイトなどがある．また，各

表 2.3: CHDM の提案において使用する記法

記号	定義
$\boldsymbol{C}$	辞書に登録されたカテゴリ集合: $\boldsymbol{C}$=$\{\boldsymbol{c_h}\|1 \leq \boldsymbol{h} \leq \|\boldsymbol{C}\|\}$. 各要素 $\boldsymbol{c_h}$ はカテゴリを表す.
$\boldsymbol{L}^{(c_h)}$	カテゴリ $\boldsymbol{c_h}$ に関連づいた URL 集合: $\boldsymbol{L}^{(c_h)}$=$\{\boldsymbol{l}_{\boldsymbol{n}}^{(c_h)}\|1 \leq \boldsymbol{n} \leq \|\boldsymbol{L}^{(c_h)}\|\}$.
$\boldsymbol{L}_{\boldsymbol{i}}^{\prime(d)}$	ユーザ $\boldsymbol{d}$ の $\boldsymbol{i}$ 番目のセッションからマッチングステップにより抽出される，URL の集合.
$\boldsymbol{P}_{\boldsymbol{i}}^{(d)}$	ユーザ $\boldsymbol{d}$ の $\boldsymbol{i}$ 番目のセッションからマッチングステップにより抽出される，カテゴリと URL のペアの集合.

カテゴリには，一つ以上の URL($\boldsymbol{L}^{(c_h)}$) が登録されているものとする．例えば，スポーツニュースには，Yahoo! スポーツが登録され，SNS サイトには mixi が登録される．もし二つの URL がそれぞれ違うカテゴリに登録され，カテゴリ間に階層関係があった場合，それら二つの URL にもまた，階層関係があるものとする．

CHDM は，辞書に登録された URL を全語彙集合と見なす．そのため，セッション中でアクセスされた URL のうち，辞書に登録された URL のみが単語の候補となりうる．さらに，辞書とのマッチングにより，セッション中から複数の単語の候補となる URL が抽出された場合で，それらに意味的な階層関係があった場合には，より上位概念の単語に統一する．

CHDM は，二つのステップ（マッチングステップ，抽象化ステップ）から構成される．まずマッチングステップでは，セッションに含まれる URL から，辞書に登録されている URL と対応するカテゴリの集合 $\boldsymbol{P}_{\boldsymbol{i}}^{(d)}$ を抽出する．これは，ログに記録された URL と辞書に登録された URL との文字列マッチングにより実現する．各ユーザについて，マッチングステップのアルゴリズムを下記に示す．

アリゴリズム 1：マッチングステップ

1 $\boldsymbol{P}_{\boldsymbol{i}}^{(d)} := \phi$

2 ***for*** *j* ***from*** *1 to* $|\boldsymbol{L}_i^{(d)}|$
3　***for*** *h* ***from*** $|\boldsymbol{C}|$ *to 1*
4　　***for*** *n* ***from*** *1 to* $|\boldsymbol{L}^{(c_h)}|$
5　　***if*** $\boldsymbol{l}_n^{(c_h)}$ *is substring of* $\boldsymbol{l}_{ij}^{(d)}$ ***then***
6　　$\boldsymbol{P}_i^{(d)} := \boldsymbol{P}_i^{(d)} \cup (\boldsymbol{c}_h, \boldsymbol{l}_n^{(c_h)})$
7 ***return*** $\boldsymbol{P}_i^{(d)}$

マッチングステップは，セッション中の全ての URL($\boldsymbol{l}_{ij}^{(d)}$) について，辞書に登録された URL($\boldsymbol{l}_n^{(c_h)}$) とのマッチングを行い，マッチした URL($\boldsymbol{l}_n^{(c_h)}$) と対応するカテゴリ $\boldsymbol{c}_h$ を抽出する．ここで URL のマッチングは，両者の文字列マッチングにより行うが，$\boldsymbol{l}_n^{(c_h)}$ が $\boldsymbol{l}_{ij}^{(d)}$ の部分文字列であれば，マッチしたものとする．これは，辞書に登録された URL は，サイトの代表的な URL が登録されるが，アクセスログ中には，そのサイトを構成する Web ページの URL やページに埋め込まれたインラインオブジェクトの URL も含まれ，それらも辞書とマッチさせるためである．例えば，$\boldsymbol{l}_n^{(c_h)}$ が‘ http://xxx/yyy/ ’であり，$\boldsymbol{l}_{ij}^{(d)}$ が‘ http://xxx/yyy/zzz.html ’であった場合にはマッチしたものと見なす．

次に，抽象化ステップは，各セッションにおいて，マッチングステップにより抽出された URL の集合から上位概念の URL の集合を抽出する．これは，辞書を利用することで，意味的な階層関係がある URL の集合を発見し，各集合について，最上位概念の URL を抽出していくことで実現する．各セッションについて，抽象化ステップのアルゴリズムを下記に示す．

アルゴリズム 2: 抽象化ステップ

1 $\boldsymbol{W}_i^{(d)} := \boldsymbol{L}_i'^{(d)}$
2 ***for*** $\hat{j}$ ***from*** *1 to* $|\boldsymbol{P}_i^{(d)}| - 1$
3　***for*** $\check{j}$ ***from*** $\hat{j} + 1$ *to* $|\boldsymbol{P}_i^{(d)}|$
4　　***if*** $\boldsymbol{c}_{i\check{j}}^{(d)}$ *is found by BFS (breadth-first search)*
5　　*from the node* $\boldsymbol{c}_{i\hat{j}}^{(d)}$ *to the bottom of the tree*
6　　***then*** $\boldsymbol{W}_i^{(d)} := \boldsymbol{W}_i^{(d)} - \boldsymbol{l}_{i\check{j}}^{(d)}$

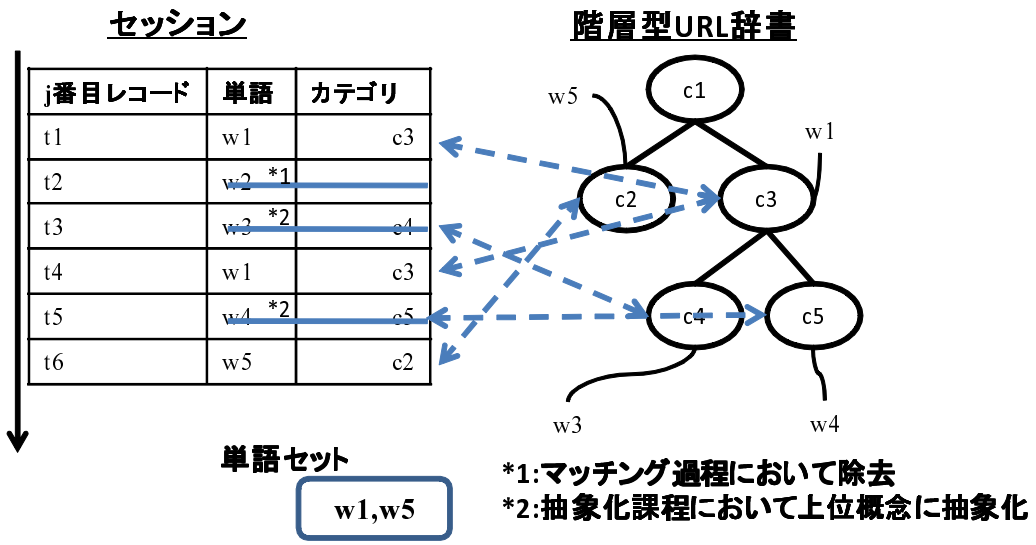

図 2.3: Cross-Hierarchical Directory Matching の例

7 ***return*** $W_i^{(d)}$

抽象化ステップでは，$P_i^{(d)}$ に登録された各カテゴリ $c_{i\hat{j}}^{(d)}$ を根として，辞書を子孫ノード方向に探索していく．もし，$P_i^{(d)}$ に登録された他のカテゴリを発見した場合には，対応する URL($l_{i\hat{j}}^{(d)}$) は，上位概念の URL($l_{i\hat{j}}^{(d)}$) に抽象化される．こうして，得られた抽象化された URL の集合が，当該セッションにおける単語セットとなる．なお，子孫ノードのみを対象に抽象化を行うため，同一カテゴリに属する二つの URL は抽象化されず，それぞれ別々の単語として抽出されることとなる．

CHDM の動作例を，図 2.3 を元に説明する．図は，プロキシログに記録されたあるユーザのセッションを左に示し，辞書を右に示している．セッションには 6 つのアクセスが含まれ，辞書には 5 つのカテゴリと 4 つの URL が登録されている．

まずマッチングステップにおいて，順にアクセスされた 6 個の URL から辞書にマッチした URL を抽出する．2 番目にアクセスされた URL は，除去される（図中の*1）．

次に，抽象化ステップでは，抽出された URL から最上位概念にあたる URL の集合を抽出する．まず，抽出された各 URL に対応するカテゴリを辞書より抽出す

る．得られたカテゴリについて，c3 と c4，c3 と c5 は，それぞれ意味的な階層関係にある．そこで，これらに対応する URL については，最上位概念にあたる c3 に対応する URL のみを抽出する（図中の*2）．従って，当該セッションから最終的に得られる単語セットは，カテゴリ c2，カテゴリ c3 に対応する URL（w1，w5）となる（図中単語セット）．

2.4　性能評価

2.4.1　データセット

CHDM によって得られたモデルの精度を評価するため，大阪大学の学生 7537 人の Web 閲覧を記録したプロキシログを，2010 年 4 月から 7 月の 4 か月に渡って収集した．ログのサイズは，40 GB，レコード数は約 1 億 3 千万レコードである．ログをセッションに分割するにあたり，タイムアウト δ を決定する必要がある．δ の値によって生成されるセッションが異なるため，最終的に得られるモデルも異なる．そこで，最適な δ を実験的に探索し，1800 [sec] と設定した．詳細は 2.4.3 項で述べる．これにより，合計で 175831 のセッションを得た．また，2010 年 7 月に Yahoo! JAPAN Directory をクロールすることで，約 57 万の URL が登録された辞書を生成した．この辞書により，セッションの 86%から一つ以上の単語が生成された．なお，以下では，全体セッションに対する一つ以上の単語が生成されたセッション数の割合 (この場合 0.86) をセッションヒット率と呼ぶ.

2.4.2　評価指標

パープレキシティ

LDA によって得られたモデルの精度を評価するため，本研究ではパープレキシティ [7] を用いた評価を示す．評価指標に用いるパープレキシティは，モデルのテスト文書に対する，各トピック中での単語の発生予測精度を示す．具体的には以

下の式で表され，値が小さいほどモデルがテスト文書に対して当てはまることを示す．

$$\boldsymbol{perplexity} = \exp(-\frac{\Sigma_{d \in \tilde{D}} \log p(d)}{\Sigma_{d \in \tilde{D}} N_d}) \tag{2.2}$$

$$\log p(d) = \Sigma_{w \in W} n_{dw} \log p(w|d) \tag{2.3}$$

ここで，$\tilde{D}$ はテスト文書の集合であり，N_d は各文書において観測された単語数である．パープレキシティは，各文書 d の確率分布 $p(w|d)$ による対数尤度の平均値で表される．従って，d をテスト文書とし，$p(w|d)$ を評価対象モデルから生成すれば，得られるパープレキシティは，当該モデルの文書に対する精度を示すこととなる．ここで $p(w|d)$ は，以下のとおりモデルから得られる θ_{dk}，β_{kw} より導出可能である．

$$p(w|d) = \Sigma_{k \in K} p(w|k) p(k|d) \tag{2.4}$$

$$p(w|k) = \frac{\theta_{dk} + \alpha}{\Sigma_{k=1}^{|K|} (\theta_{dk} + \alpha)} \tag{2.5}$$

$$p(k|d) = \frac{\beta_{kw} + \eta}{\Sigma_{w=1}^{|W|} (\beta_{kw} + \eta)} \tag{2.6}$$

以下では特に断らない限りは，前 3ヶ月のログを学習期間としてモデルを生成し，残りの 1ヶ月のログをテスト文書として利用する．

重要度付ヒット率

パープレキシティによる評価は，モデルの最適性を示すのに有効であるが，実アプリケーションにおけるモデルの性能を評価することは出来ない．そこで本研究では，生成したモデルを利用したユーザプロファイリングを行い，これに基づきコンテンツ推薦の問題に適用し，実アプリケーションに対する性能評価を行った．

本評価ではまず，モデルを利用したユーザプロファイリングを行う．具体的には，生成したモデルによって，ユーザを特定のトピックに分類する．これは，前

3ヵ月のログから得られた $\boldsymbol{\theta}$ を利用し，各ユーザに最も高い確率が与えられているトピックへ，当該ユーザを割り当てることで実現する．

各ユーザをモデルを利用して各トピックに分類後，残りの 1ヶ月のログからランダムに 1000 人のテストセットユーザと 6537 人の学習セットユーザを選択し，学習セットユーザの Web 閲覧ログを利用して，テストセットユーザの Web 閲覧行動を予測し，推薦すべき URL の集合を決定する．具体的には，各セッションにおいて，各テストセットユーザと同一トピックに属する学習セットユーザがアクセスしたトップ N の URL を，推薦する URL の集合とする．実際にテストセットユーザがアクセスした URL と比較し，推薦した URL のヒット率が高いほど，当該モデルによる予測精度が良いこととなる．

さらに本評価では，予測精度の指標にセレンディピティ [55] を導入する．セレンディピティの考え方によれば，コンテンツ推薦に成功した場合，そのコンテンツの人気が低いほど，当該推薦の価値が高くなる．これに従い，本研究では，人気の低い URL に高い重要度を付与する．具体的には，文書分類分野で広く利用される重み付けの指標である IDF（Inverse Document Frequency）[38] を利用し，以下の通り各 URL$\boldsymbol{l}$ に重要度（$\boldsymbol{weight}$）を付与する．

$$\boldsymbol{weight(l)} = \log \frac{\textbf{全ユーザの全セッション数}}{\boldsymbol{URLl}\textbf{がアクセスされたセッション数}} \tag{2.7}$$

式によれば，アクセスされたセッション数が少ないほど，当該 URL に高い重要度が付与されることとなる．

上記定義した重要度に基づき，各テストユーザの各セッションにおける，推薦した URL の重要度付ヒット率を下記の通り定義する．

$$\textbf{重要度付ヒット率} = \frac{\Sigma_j \boldsymbol{h_j weight(l_j)}}{\Sigma_j \boldsymbol{weight(l_j)}} \tag{2.8}$$

ここで $\boldsymbol{l_j}$ は，当該セッションにおいて $\boldsymbol{j}$ 番目にアクセスされた URL であり，$\boldsymbol{h_j}$ は，$\boldsymbol{l_j}$ が推薦された URL に含まれていれば 1，含まれていなければ 0 となる．こうして得られた重要度付ヒット率は，人気のないコンテンツの推薦がヒットするほど高い値となる重みが付与された評価指標となる．また，重要度付ヒット率は，

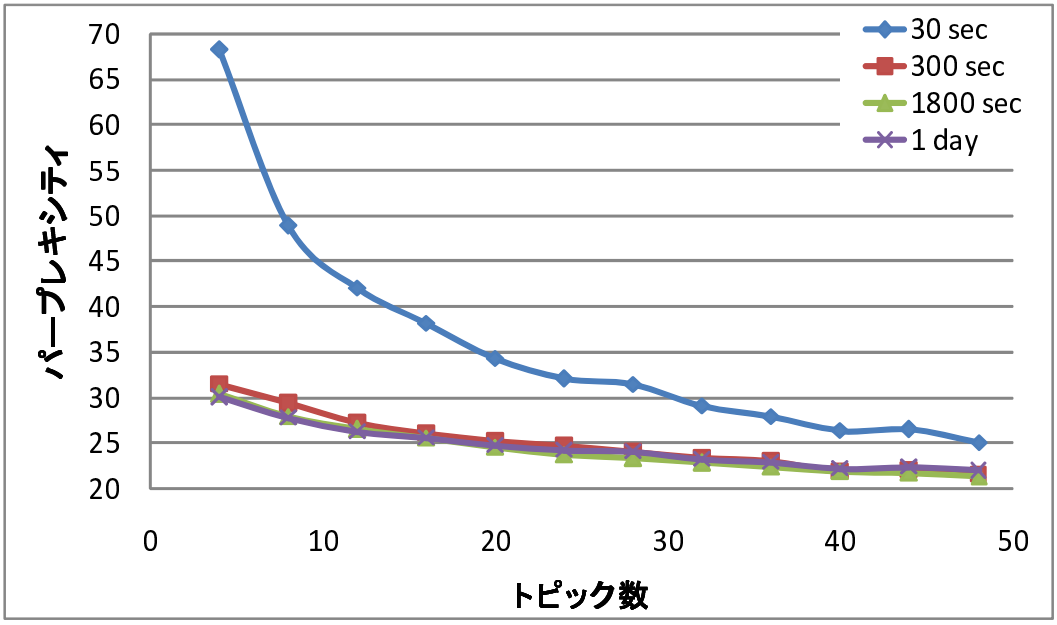

図 2.4: タイムアウト毎のパープレキシティの変化

推薦する URL の数が多くなるほど高くなるが，多くの場合，推薦システムやアプリケーションの制約によって，数個程度に制限される．

2.4.3 評価結果

まず，タイムアウト δ を決定する．各タイムアウトにおいて，LDA のハイパーパラメータを $\alpha = |K|/50$，$\eta = 0.01$，トピック数を 24 とした場合の，提案方式におけるパープレキシティを図 2.4 に示す．図は，横軸が LDA のトピック数，縦軸がパープレキシティを表す．図に示す通り 1800[sec] で最も良い性能を示すことが分かる．そこで，以下では，δ を 1800[sec] と設定した結果を示す．

次に，提案方式の評価に利用する LDA のハイパーパラメータを決定する．いくつかの α，及び η を与えた場合の，トピック数 24 におけるパープレキシティの平均値を表 2.4 に示す．表に示す通り，パープレキシティはハイパーパラメータによっては，ほぼ変化しない．そのため，これらの値については，これまでに提案された既存の LDA システム（[15,30,78]）と同様に，Griffiths らが提案した $\alpha = |K|/50$,

表 2.4: ハイパーパラメータによるパープレキシティの変化

$\boldsymbol{\alpha}$	$\mathbf{10}/\lvert\boldsymbol{K}\rvert$	$\mathbf{30}/\lvert\boldsymbol{K}\rvert$	$\mathbf{50}/\lvert\boldsymbol{K}\rvert$	$\mathbf{70}/\lvert\boldsymbol{K}\rvert$	$\mathbf{90}/\lvert\boldsymbol{K}\rvert$
$\boldsymbol{\eta}$	0.01	0.01	0.01	0.01	0.01
パープレキシティ	23.8706	23.8786	23.7952	23.8132	23.9345
$\boldsymbol{\alpha}$	$\mathbf{50}/\lvert\boldsymbol{K}\rvert$	$\mathbf{50}/\lvert\boldsymbol{K}\rvert$	$\mathbf{50}/\lvert\boldsymbol{K}\rvert$	$\mathbf{50}/\lvert\boldsymbol{K}\rvert$	$\mathbf{50}/\lvert\boldsymbol{K}\rvert$
$\boldsymbol{\eta}$	0.01	0.03	0.05	0.07	0.09
パープレキシティ	23.7952	23.7664	23.8561	23.8873	23.9735

$\boldsymbol{\eta} = \mathbf{0.01}$ を適用する [27]．なお，ハイパーパラメータの最適値の推定方法については，Gabriel ら [15] が提案する方法がある．

以上設定したパラメータの元で，CHDM で得られたモデルの精度を，以下 2 つの方式で得られた精度と比較する．まず，directory-matching で，これは抽象化を一切行わないマッチングステップのみを適用した方式である．次に，辞書を用いずに URL のディレクトリ構造により抽象化を行った domain-abstraction で，これは，同一セッション中において，マッチングステップにより抽出された URL 集合同士を文字列マッチングすることにより，最も URL の階層が上位の URL のみを抽出する方式である．例えば，同一セッション中に，‘ http://xxx/ ’と‘ http://xxx/yyy/ ’があった場合には，‘ http://xxx/ ’のみを単語として抽出する．CHDM を含む以上 3 方式は，同一辞書を利用するため，語彙集合は全て同じであり，パープレキシティによる比較が可能である．また，CHDM に対して，domain-abstraction と directory-matching を比較することにより，辞書による抽象化の有効性を示すことが可能となる．

パープレキシティ

まず，パープレキシティによる評価結果を，図 2.5 に示す．図は，各方式について，トピック数毎のパープレキシティの変化を示している．CHDM は，他の二方式と比較して，良い精度を示している．特に，directory-matching から $\mathbf{10\%}$ 程度向上しており，辞書による抽象化が効果が大きいことがわかる．一方で，domain-abstraction は，directory-matching と比較して逆に精度が悪化しており，辞書に従わない抽象

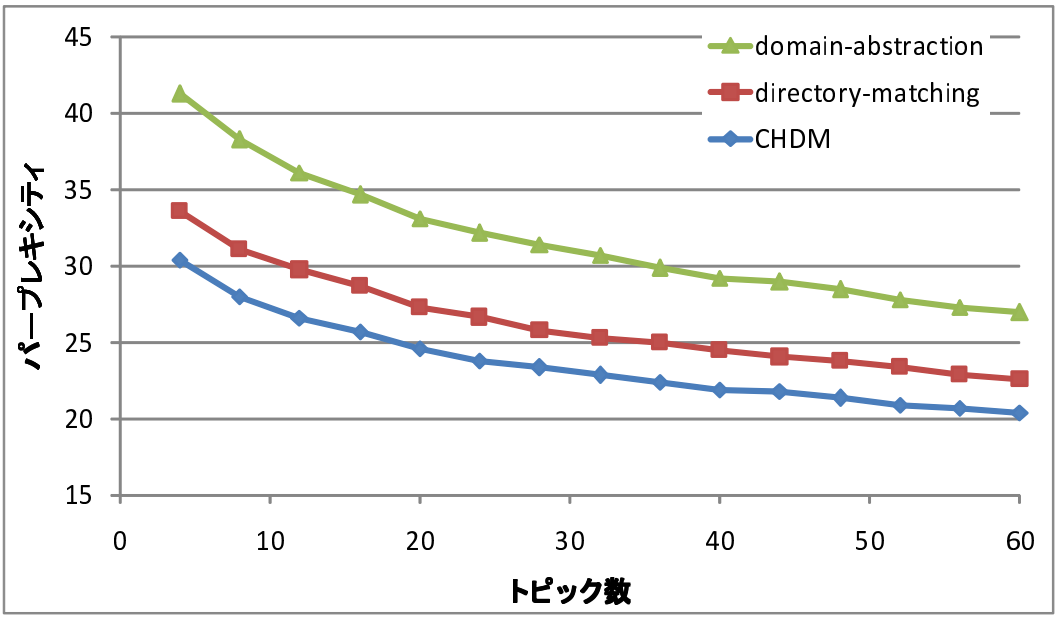

図 2.5: 各方式におけるパープレキシティの比較

化を行った場合は，逆効果となってしまうことが分かる．

次に，辞書のセッションヒット率による，モデルの精度を評価する．学習データにおいて，単語が生成されたセッションのうち，ランダムに 25%ずつプロキシログからセッションを除いた上で，LDA のモデルを生成し，パープレキシティを測定した．この場合，100%から 25%とセッションを減らしていくにつれて，セッションヒット率は，0.86，0.64，0.43，0.21 と変化した．

結果を図 2.6 に示す．図は，各学習データにおけるパープレキシティの変化を示している．評価結果より，セッションヒット率が減少するにつれ，急激にパープレキシティが悪化することが分かる．特に 0.21 の場合は，パープレキシティが元の 10 倍程度に増加する．これにより，セッションヒット率がモデルの精度に及ぼす影響は大きく，辞書の生成は非常に重要であることが分かる．

最後に，学習データの選択によるモデルの精度を評価する．以上示した実験では，4 月から 6 月のデータを全て学習データとしてモデルを生成したが，実際には時系列によるユーザの Web アクセスのパターンが変化しており，古い学習データがモデルの精度に悪影響を及ぼしている可能性がある．例えば，より直近の 6 月

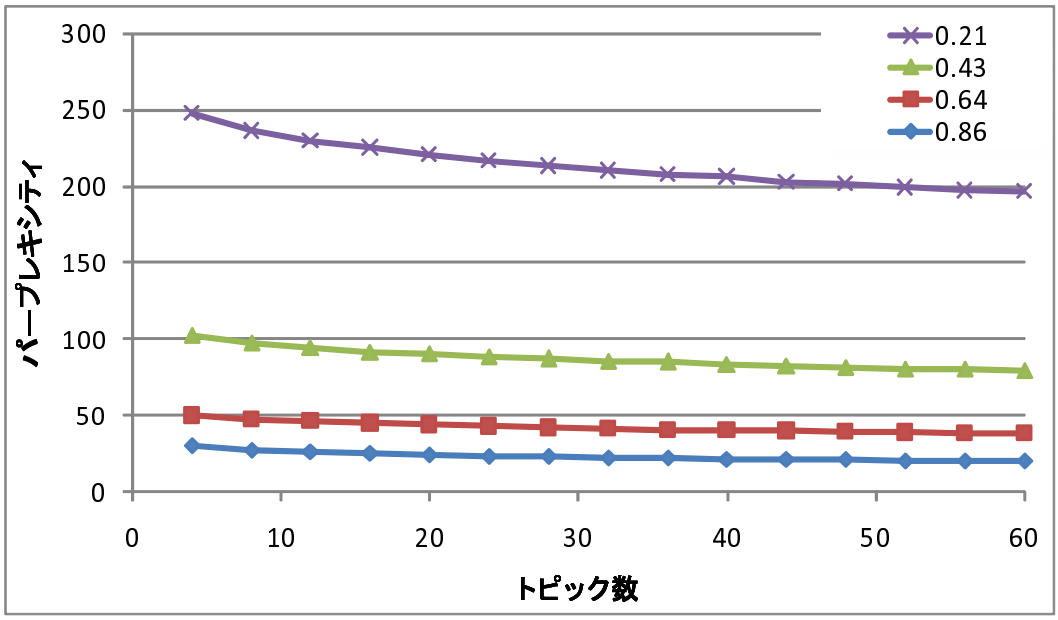

図 2.6: セッションヒット率毎のパープレキシティ

のデータから生成されたモデルが，4 月や 5 月のデータを含んだモデルよりも，精度が良いことも考えられる．このような，時系列の学習データによる影響を確かめるため，様々な学習データを利用してパープレキシティを測定した．

結果を図 2.7 に示す．図は，各学習データにおけるパープレキシティの変化を示している．4-6 月データとは，全ての学習データを利用した場合の結果であり，5-6 月は，5 月と 6 月のみを利用した場合であり，4 月データ，5 月データ，6 月データは，それぞれの月の学習データを利用した結果となる．図より，全てのデータを利用して生成したモデルが最も精度が良いことが分かる．ただし，5-6 月や 6 月データを利用した場合では，それほど精度が落ちないのに対して，4 月や 5 月のデータのみを利用した場合には，急激に精度が悪化する．そのため，直近の学習データに重みを置いたモデルの生成が有効であると予測される．

このような，時系列毎の学習データの重みを考慮した，LDA によるユーザプロファイリング方式として，Iwata ら [35] が提案した動的なトピックモデルによるユーザプロファイリング方式がある．Iwata らの方式は，過去の時刻のモデルを事前分布とし，現在時刻のモデルを生成する．その際に，現在に近いモデルに重み

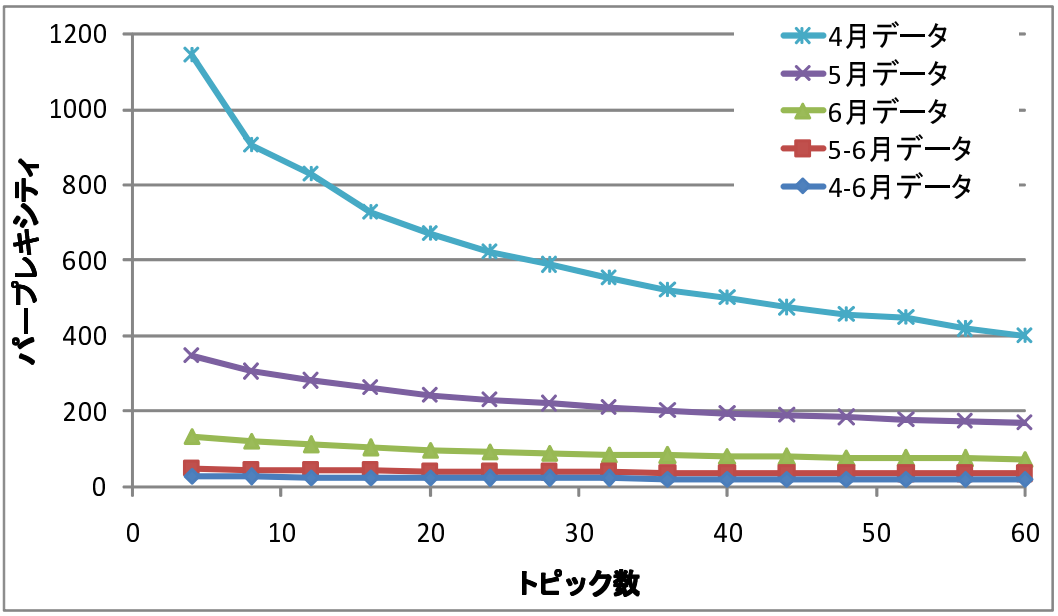

図 2.7: 学習データ毎のパープレキシティ

を置いた事前分布を生成することで，時系列によるモデルの変化の影響を考慮したモデルの生成が可能となる．今後，Iwata らの方式と提案方式とを組み合わせることで，より良いモデルを生成可能な方式について検討していく．

重要度付ヒット率

次に，重要度付ヒット率による評価結果を示す．まず，評価に利用する LDA のトピック数を決定する．ここでは，モデルの生成に CHDM を利用し，各トピック数における全テストユーザの全セッションにおいて導出した重要度付ヒット率の平均値を評価した．結果を図 2.8 に示す．図は，推薦した URL 数が 5 と 10 の場合の結果であり，横軸はトピック数，縦軸は重要度付ヒット率を表す．図に示す通り，いずれの場合も，トピック数 24 で最も良い結果となった．そこで，以下ではこの値をトピック数として設定した．

次に，推薦 Web ページ数毎の重要度付ヒット率を図 2.9 に示す．図は，CHDM と directory-matching を利用したユーザ分類に基づき，全テストユーザの全セッショ

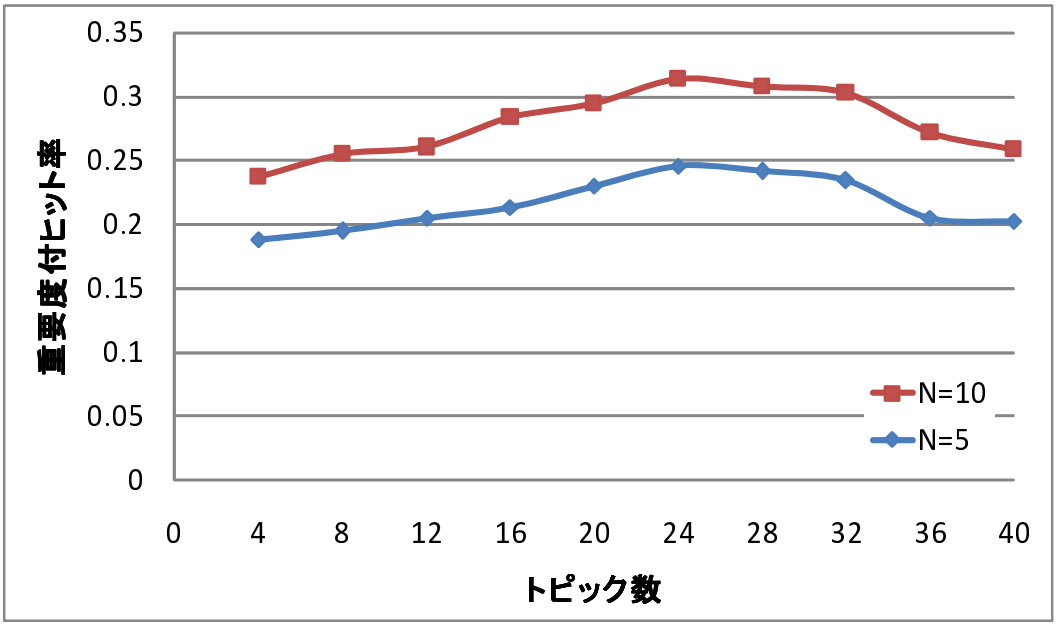

図 2.8: トピック数毎の重要度付ヒット率の比較

ンにおいて導出した重要度付ヒット率の平均値を表したものである．さらに，比較のために，アクセスされたセッション数が多い順（人気順）に推薦した場合のヒット率を baseline として示した．横軸は，推薦した URL 数，縦軸は重要度付ヒット率を表す．図より，directory-matching と baseline の精度がほぼ精度が等しい一方で，CHDM はこれら二つの手法と比較して，最大で 10%程度精度が高かった．これにより，CHDM による辞書を利用した抽象化手法が，コンテンツ推薦のアプリケーションにおいて，有効であることを確認できた．また，ユーザの Web 閲覧行動を高精度にモデル化したことにより，全体のユーザにとって人気のあるコンテンツではなく，当該ユーザにとって価値のあるコンテンツを提供可能であることが確認できた．

2.5　ユーザプロファイルの可視化

本節では，CHDM により得られたモデルを利用したユーザプロファイリング結果を示し，モデルの有効性を主観的に評価する．データは，2.4.3 項で利用したも

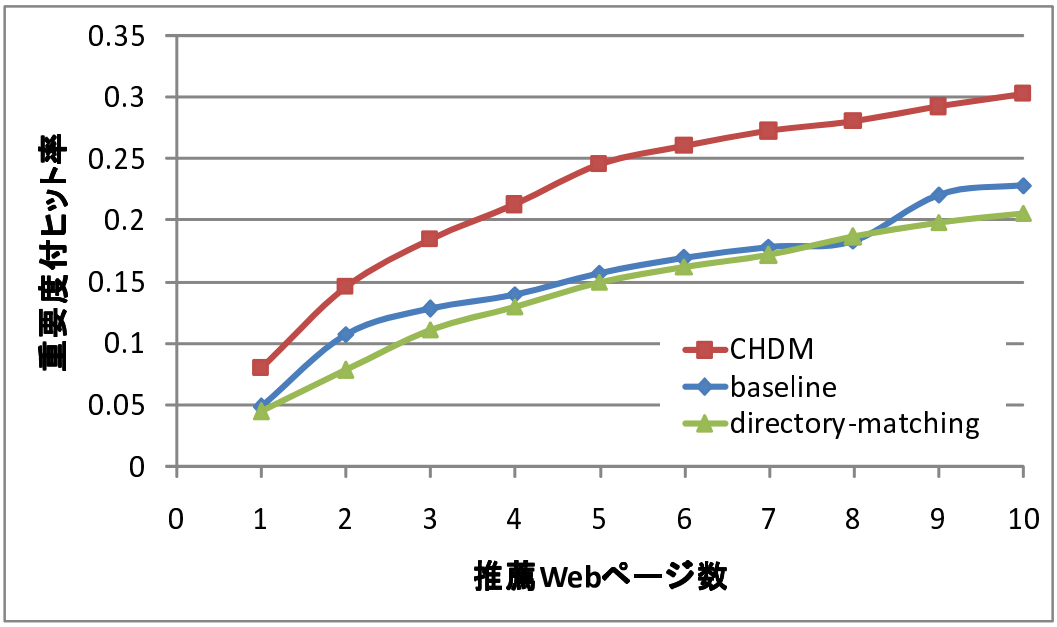

図 2.9: 各方式における重要度付ヒット率の比較

のと同じデータを用い，CHDM により得られたモデルを用いる．また，トピック数には，重要度付ヒット率の評価において最も精度の良かった 24 を設定した．

得られた 24 のトピックについて，代表的な単語と，それらより名づけたトピック名を表 2.5 に示す．図より，学生の勉学，趣味，就職など様々なトピックが抽出でき，学生の Web 閲覧行動がモデル化出来ていることが分かる．

また，得られたモデルの有効性をさらに裏付ける結果として，得られたトピックが，学生が持つ属性（専攻，学年）と強い相関を持つことが分かった．以下では，24 のトピックを,「専攻（理系/文系）」と「学年（高学年/低学年）」の二つの軸に射影することで，これを可視化する．具体的には，各ユーザ $\boldsymbol{d}$ が持つ各トピックの分布を元に，理系度 $\boldsymbol{x_d}$ と高学年度 $\boldsymbol{y_d}$ によって得られる属性値 $\boldsymbol{a_d}$ を導出し，これを二次元上に配置することとする．なお，$\boldsymbol{x_d}$ は，トピックが理系的であるほど大きく，文系的であるほど小さい．また，$\boldsymbol{y_d}$ は，トピックが高学年傾向であるほど大きく，低学年傾向であるほど小さい．そして，ユーザのトピック分布 $\boldsymbol{\theta_{dk}}$ が与えられた場合に，そこから属性値 $\boldsymbol{a_d}$ を導出するための回帰式を下記の通りモデル化し，これを導出する．

表 2.5: 24 のトピックと代表的な単語

トピック	代表的な単語	トピック	代表的な単語
#1 MSN サービス	Hotmail SkyDrive	#13 2 ちゃんねる まとめ	痛いニュース 2 ログ
#2 YouTube	Youtube MegaVideo	#14 Yahoo!ニュース	Yahoo!ニュース イザ！
#3 就職活動	リクナビ 2012 facebook	#15 ブログ	goo ブログ Yahoo!ブログ
#4 mixi	mixi Google ニュース	#16 ニコニコ動画動	ニコニコ動画 ニコニコ大百科
#5 外出	Yahoo!天気情報 Google maps	#17 ネット ショッピング	Yahoo!オークション Amazon
#6 新聞ニュース	毎日新聞 Google ニュース	#18 工学専攻	工学部ページ
#7 スポーツ	Yahoo!スポーツ FIFA WC	#19 Wikipedia	Wikipedia
#8 生物遺伝専攻	蛋白質研究所 遺伝情報実験センタ	#20 アルバイト	タウンワーク フロム・エーナビ
#9 図書館利用	大阪大学図書館	#21 理系レポート	Latex コマンド一覧 IT 用語辞典
#10 株	Yahoo!ファイナンス Yahoo!掲示板	#22 Q and A サイト	Yahoo!知恵袋 教えて goo
#11 低利用集団	大阪大学ポータル	#23 Twitter	Twitter, Twitpic
#12 2 ちゃんねる	2 ちゃんねる アメーバブログ	#24 プログラミング	学習 C 言語 C 言語入門

入力:$\{\boldsymbol{\theta}_{dk}, \boldsymbol{a}_k\}_{k=1}^{24}$

出力:$\boldsymbol{a}_d$

ただし $\boldsymbol{a}_k$ はトピック毎の平均的な属性値のペア ($\boldsymbol{x}_k$,$\boldsymbol{y}_k$) であり，$\boldsymbol{a}_d$ は，ユーザ毎の属性値のペア ($\boldsymbol{x}_d$,$\boldsymbol{y}_d$) を表す.

$\boldsymbol{a}_k$ は，各トピックが持つ平均的な属性値であり，以下の通り導出する．まず，各ユーザが持つ実際の属性値（専攻，学年）を $-\mathbf{1}$ から $\mathbf{1}$ に数値化する．専攻がとる値は，理系なら $\mathbf{1}$，文系なら $-\mathbf{1}$ であり，学年は 1 年が $-\mathbf{1}$，2 年が $-\mathbf{1/3}$，3 年が

$\mathbf{1/3}$，4 年が $\mathbf{1}$ である．この実際の属性値を，以下では $\boldsymbol{a_{\tilde{d}}(x_{\tilde{d}}, y_{\tilde{d}})}$ とする．例えば，2 年生の理科系のユーザは，$\boldsymbol{a_{\tilde{d}}(x_{\tilde{d}}, y_{\tilde{d}}) = (-1/3, 1)}$ となる．さらに，LDA の出力 $\boldsymbol{\theta}$ より，各ユーザが持つトピックの分布が分かるため，その中から最も大きな値を持つトピックを抽出する．こうすることで，トピック毎に，当該トピックの成分を最も強く持つユーザと，そのユーザの属性値が分かるため，それらのユーザの属性値の平均をとることで，トピック毎に平均的な属性値 $\boldsymbol{a_k}$ を導出可能である．次に，全体ユーザから学習セットを選択し，回帰式を学習する．学習時には，出力 $\boldsymbol{a_d}$ には，$\boldsymbol{a_{\tilde{d}}}$ を利用する．また，学習には RVM [75] の Regression mode を利用した．

導出した回帰式を利用して，7537 人全てのユーザの $\boldsymbol{a_d}$ を導出し，二次元グラフ上にマップした結果を，図 2.10 に示す．各点は，ユーザを表し，x 軸の正方向に配置されるほど，当該ユーザのトピックが理系的であり，負方向に配置されるほど文系的であり，y 軸の正方向に配置されるほど高学年傾向であり，負方向に配置されるほど低学年傾向となる．また，24 のトピック数それぞれに対応する色を用意し，各ユーザの最も支配的なトピックにより，各ユーザを 24 色のいずれかで色づけしている．さらに，各トピック（#1-#24）に色づけされたユーザの人数を図の左下の棒グラフで示している．

図では,‘ 就職活動 ’(#3),‘ 生物遺伝専攻 ’(#8),‘ Wikipedia ’(#19),‘ 低利用集団 ’(#11),‘ 工学専攻 ’(#18),‘ 理系レポート ’(#21), 及び‘ プログラミング ’(#24) など，属性値に強く関係があると思われる，大学生活関連のトピックについては，同じ色の点が同じ場所に集まる傾向が強く見られる.（‘ Wikipedia ’は，文系低学年の授業で広く使われた.）これにより，生成したモデルから得られたプロファイリング結果は，学生の属性を良く反映しており，定性的に良いモデルが得られていると言える．

一方で,‘ mixi ’(#4) , ‘ ニコニコ動画 ’(#16) , ‘ Youtube ’(#16) ‘ Twitter ’(#21) など，属性値にそれほど関係がないと思われるコミュニティサイトやコンテンツサイトについては，点が集まる傾向は見られない．これらのトピックを可視化するためには，趣味・思考など大学生の属性値とは別の軸により，射影する必要があると考えられる．

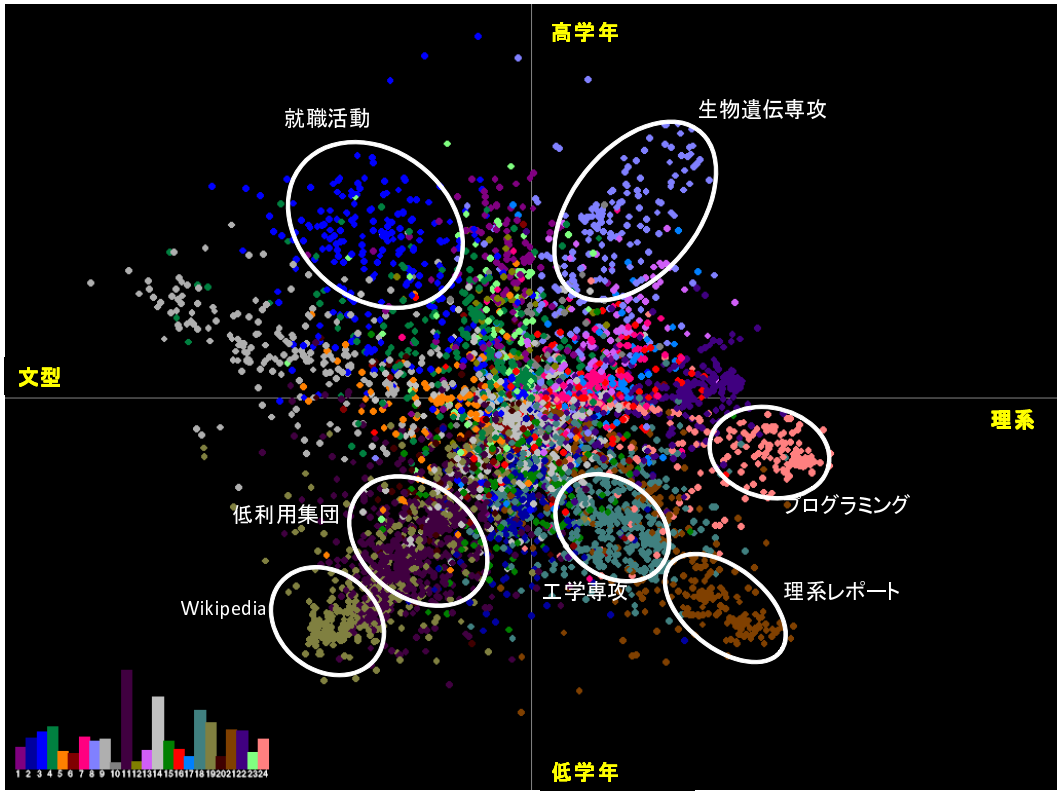

図 2.10: トピックの 2 次元射影結果

2.6　むすび

本章では，ユーザが蓄積した Web 閲覧ログをユーザの意図を反映した文書集合であると見なし，ユーザプロファイルやコミュニティ分析を目的とした，ユーザの Web 閲覧行動のモデル化手法を提案した．提案方式では，階層型 URL 辞書を利用した URL 系列の抽象化を行うことにより，高精度なモデルを得られる単語セットの抽出を可能とした．また，7537 人のユーザのプロキシログに対して提案方式を適用し，単語の予測精度の評価を行うことで，提案方式の有効性を示した．また，ユーザプロファイリングに基づく Web ページ推薦を例として，実アプリケーションにおける提案方式の有効性を示した．さらに，モデルから得られたプロファイリング結果を可視化し，主観的に，モデルの有効性を示した．

本章では，特に，Web 閲覧ログを文書集合の例としたが，ユーザを文書，ユー

ザがアクセスしたWebページを単語と見なすことで，一般的な文書集合と同様に扱った．そのため，提案方式は，単語間の概念辞書を構築することで，Web閲覧ログに限らず，いかなる文書集合に対しても，有効な手法であると言える．

提案方式は，使用する階層型URL辞書により，精度が異なる．そこで，今後は，構成する辞書の違いによる性能比較を行い，より精度の良い辞書の構成手法について検討する．

第3章　時系列文書のトピック解析

3.1　まえがき

インターネット上では，様々なソーシャルメディアを通して，ユーザによって生成された大量の文書が，リアルタイムに生成・蓄積される．これら時系列のユーザの意図を反映した文書集合を分析することにより，時事刻々と移り変わるユーザ間でのホットトピックやトレンドを知ることが可能となるため，注目が高まっている．例えば，[46] や [67] では，Twitter 上で生成される文書を解析することで，話題の盛り上がりを表すホットトピックやホットワードを抽出する手法が提案されている．そこで本章では，1 章で示した分類 2（文書集合に対する時系列分析技術）の技術として，これら様々なユーザが時系列に蓄積することで，時系列に生成される文書集合を高精度にモデル化可能な手法を提案し，ホットトピック抽出への応用を示す.

文書集合を対象とした分析であるため，トピックモデルの LDA [7] を適用することで，時系列に生成される文書集合をモデル化する手法を確立する．これまでに，文書集合の時系列解析に LDA を利用した研究がいくつか報告されている [2, 6, 35, 36, 58, 76, 77, 79]．LDA を拡張した動的なトピック解析手法（以下，動的 LDA と呼ぶ）では，過去からの文書集合の時系列変化を学習してモデルを生成していくことで，生成される文書集合の時系列の変化に対して，ロバストなモデルを生成可能となる．ここで時間変化に対してロバストであるとは，現在までに観測された文書を元に生成されたモデルが，未来の時刻に入力された文書集合を精度良くモデル化可能であることを意味する．本研究におけるモデルの精度は，生成したモデルによる，モデル化対象の文書集合に対する単語の発生予測精度（パープレキシティ [7]）により評価する.

Dynamic Topic Model [6]（DTM）は，動的 LDA として，最初に提案されたアルゴリズムである．DTM は，文書集合とモデルの生成にマルコフ過程を仮定し，現在のモデルの事前分布を，前時刻のモデルから生成することで，時間変化にロバストなモデルの生成を可能としている．これに対して，[36] や [58] では，前の時刻のモデルだけではなく，過去のモデルを様々な時間間隔で合成し，これらを元に事前分布を生成することで，文書の時間変化に対してよりロバストなモデルを生成可能としている．また，[35] や [79] では，動的にモデルを更新していくだけではなく，同時にモデルの混合分布により表わされる各文書に関しても，動的に更新する．この手法は，時間に応じて新規文書が生成されるのではなく，各文書の内容が更新されていくようなデータセットに対して，特に有用である．[66,76,77] では，時間を DTM のような離散変数ではなく，連続変数として扱う手法を提案している．これにより，マルコフ過程を仮定した手法が離散的に生成していたモデルを，時間の関数として連続的に生成可能としている．ただし，これらの手法は，モデル自体は固定であると仮定した上で，各モデルが時間変化によりどのように盛り上がり，どのように衰退していくかを追跡するための手法であり，DTM のように，時間変化に応じて動的にロバストなモデルを生成することはできない．

上述した既存研究は主に，LDA の確率モデルの拡張や文書表現の多様化に着目して，提案されたものである．一方で本研究では，例えば Twitter，ニュース記事，ブログなど，日々様々な話題や新語が生成されつづけるような文書集合を対象とした，動的 LDA の手法を提案する．このような文書集合において，新しく生成される文書に対して，より精度の良いモデルを生成するために，本研究では，既存研究が扱ってこなかった二つの課題を解決する．

第一の課題は，時系列に生成されるモデルの平滑化である．日々変化しつづける文書集合からモデルを生成する際，過去のモデルと現在のモデルとを平滑化することで，モデルの逐次の文書に対する過学習を防ぎ，より時間変化にロバストなモデルを生成可能となる．例えば [28] では，LDA のオンライン学習において，過去のモデルと現在のモデルとを混合して平滑化することにより，次の試行における事前分布を導出し，これにより逐次生成される文書集合に対して，高精度なモデルを生成することに成功している．しかし，彼らの手法は，平滑化時のパラ

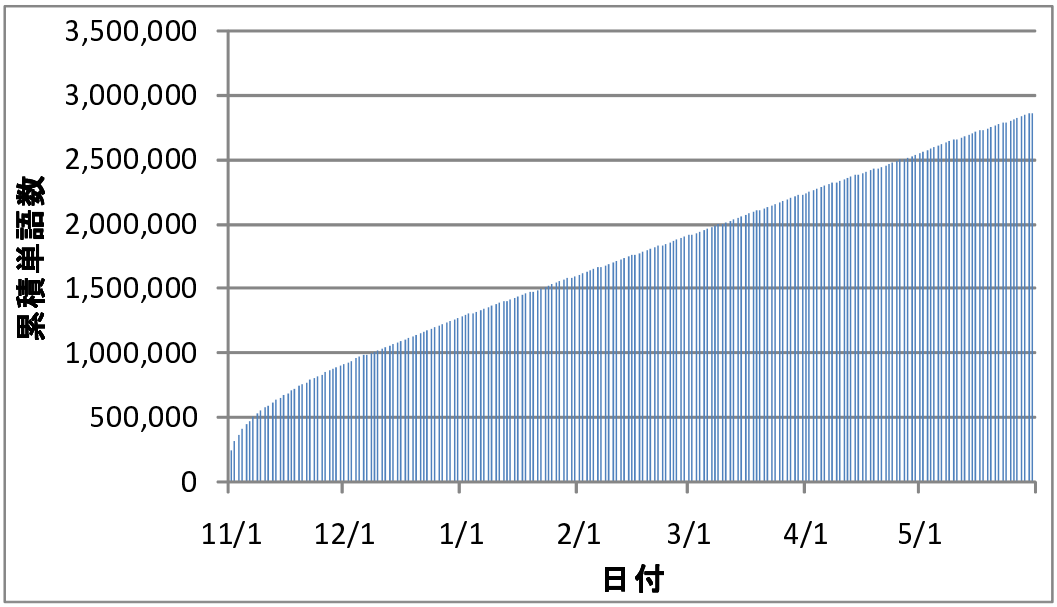

図 3.1: 2011 年 11 月から 2012 年 5 月までの Twitter 文書における累積単語数

メータをあらかじめ固定的に決定しており，モデルが最適に平滑化されているとは限らない．これに対し，本研究では，最適な平滑化パラメータを導出することで，モデルを最適に平滑化する．ここで，平滑化の最適性は，平滑化したモデルによる，未観測文書の生成予測精度により評価されるものとする．

第二の課題は，逐次の文書集合生成時に発生する新語への対応である．図 3.1 は，2011 年 11 月から 2012 年 5 月までの 7 か月間収集した 1 日 100 万のツイートにおいて観測された，累積の名詞の数を表している．図に示す通り，半年以上経過した後であっても累積値が増え続けており，一定数の新語が日々登場し続けることが分かる．既存の動的 LDA は，いずれも，あらかじめ定められた固定的な語彙集合を扱っているため，Twitter のように新語が多く発生する文書集合に対して，新語を含めたモデルを生成することを考えた場合，精度の良いモデルを生成することはできない．これに対し，本研究では，動的に語彙集合を拡大可能なシステムを実現し，新語を追加可能とする．

以上の二つの課題を解決し，逐次生成されつづけるる文書集合から，高精度なモデルを生成することが，本章における主題となる．具体的には，以下の通りで

ある．

1. DTM を拡張し，時系列に生成される LDA のモデルを最適に平滑化するアルゴリズムを提案する．提案方式は，パーティクルフィルタ [42] を利用することで，最適な平滑化パラメータを導出する．

2. DTM をさらに拡張し，動的に語彙集合を拡大可能なシステムを提案する．提案方式は，新規の文書集合の生成に合わせて，システムが扱う語彙集合に新語を追加可能とする．

3. 2 つのデータセット（合計 9 か月間，1 日 100 万ツイート）を対象とした大規模な実験結果を示す．実験では，本研究が提案した二つの拡張により，DTM に対してモデルの精度が大きく上回ることを，パープレキシティ評価により示す．また，実際のホットトピック抽出においても提案方式が有効であることを，KL（Kullback–Leibler）ダイバージェンス [48] に基づく評価により示す．さらに提案方式が，計算時間において，語彙集合の拡大に対してスケーラブルであることを示す．

上述したように，本研究の主題は，最適なモデル平滑化と動的な語彙集合を導入することで，新規に生成される文書に対して，高精度なモデルを生成することである．著者の知る限り，動的 LDA を利用した時系列文書分析において，同様の手法を提案する研究はない．

以下では，3.2 節で最適なモデル平滑化方式を提案し，3.3 節では動的に語彙集合を拡大可能なシステムを提案する．また，3.4 節では，提案方式を大規模なツイートデータに適用し，パープレキシティにより定量的に評価し，最後に 3.5 節でまとめと課題を述べる．

3.2 時系列モデルの最適な平滑化方式

3.2.1 Dynamic Topic Model

Blei らが最初に提案した LDA [7] は，文書集合 $\boldsymbol{D}$ を入力として，二つのパラメータ $\boldsymbol{\theta}$，$\boldsymbol{\beta}$ を出力する．$\boldsymbol{D}$ の各要素は一つの文書であり，あらかじめ用意された語彙集合に含まれる単語と，当該単語の当該文書内での出現回数によってあらわされる．また，$\boldsymbol{\theta}$ は，文書のトピックによる多項分布表現であり，$\boldsymbol{\beta}$ はトピックの単語による多項分布表現である．これらは，ハイパーパラメータである $\boldsymbol{\alpha}$，$\boldsymbol{\eta}$ を元に生成される．

Blei らの LDA に代表される手法は，ある時刻における静的なモデルを抽出する手法（本章では静的 LDA と呼ぶ）であるため，時系列に生成されていく文書集合から，逐次モデルを生成していく手法としては適さない．これに対して，動的 LDA は，静的 LDA を拡張した手法であり，時系列に生成されるモデルに対して，時間変化にロバストなモデルを生成可能とする [2, 6, 35, 36, 58, 76, 77, 79].

動的 LDA として最初に提案されたのが Dynamic Topic Model(DTM) [6] である．以下，表 3.1 の記法を元に，DTM の概要を述べる．DTM は，時系列の文書からのモデル生成を可能とするために，ある定められた時間間隔で文書が入力され，それに伴いモデルが生成されていく，マルコフ過程を仮定する．図 3.2 は，DTM のグラフィカルモデルを表す．各時刻 $\boldsymbol{t}$ において，DTM は生成された文書 $\boldsymbol{D}^{(t)}$ に対して，$\boldsymbol{\theta}^{(t)}$，$\boldsymbol{\beta}^{(t)}$ を生成する．また，次の時刻のモデルに対する事前分布は，現在の時刻のモデルより生成する．なお，初期状態のモデルは，静的 LDA と同じ方法で生成される．前の時刻のモデルを考慮することにより，生成されたモデルは文書集合の時系列の変化にロバストとなり，次の時刻に生成される文書により適合したモデルとなる．

3.2.2 モデル平滑化のための確率モデル

DTM は，次の時刻に対する事前分布を現在のモデルからのみ生成するため，特に，各時刻において，生成される文書の話題が変化し続けるような文書集合に対

表 3.1: 3.2 節で使用する記法

記号	定義
$\boldsymbol{K}$	LDA のトピック集合．各要素は $\boldsymbol{k}$ で表す．
$\boldsymbol{D}^{(t)}$	時刻 $\boldsymbol{t}$ における文書集合．各要素は，文書 $\boldsymbol{d}$ 中に含まれる単語 $\boldsymbol{w}$ の出現頻度 $\boldsymbol{n}_{\boldsymbol{dw}}^{(t)}$ で表す．
$\boldsymbol{\beta}^{(t)}$	時刻 $\boldsymbol{t}$ におけるトピック $\boldsymbol{k}$ の単語 $\boldsymbol{w}$ による多項分布表現（時刻 $\boldsymbol{t}$ におけるモデル）．各要素は，$\boldsymbol{\beta}_{\boldsymbol{kw}}^{(t)}$ で表す．
$\boldsymbol{\theta}^{(t)}$	時刻 $\boldsymbol{t}$ における文書のトピックによる多項分布表現
$\hat{\boldsymbol{\beta}}^{(t)}$	次の時刻のモデルの事前分布
$\boldsymbol{\rho}^{(t)}$	時刻 $\boldsymbol{t}$ における平滑化係数（$\boldsymbol{0 \leq \rho^{(t)} \leq 1}$）
$\boldsymbol{\rho}_{\boldsymbol{i}}^{(t)}$	時刻 $\boldsymbol{t}$ における平滑化係数の各要素（$\boldsymbol{i = \{1, 2, ..., N\}}$，$\boldsymbol{0 \leq \rho_i^{(t)} \leq 1}$）．
$\boldsymbol{w}^{(t)}$	パーティクルフィルタにおける尤度．各要素 $\boldsymbol{w}_{\boldsymbol{i}}^{(t)}$ は，粒子 $\boldsymbol{\rho}_{\boldsymbol{i}}^{(t)}$ に対応する．
$\boldsymbol{\sigma}$	パーティクルフィルタにおけるランダムウォークの距離．

しては，現在のモデルを過学習し，精度が低下する可能性がある．これに対して本研究では，事前分布を，現在のモデルと過去のモデルを平滑化することで，よりロバストなモデルを生成する手法を提案する．また，各時刻における平滑化のパラメータについては，パーティクルフィルタ [42] を利用することで，逐次最適化する．

提案方式のグラフィカルモデルを図 3.3 に示す．提案方式は，DTM に対して，新しいパラメータ $\boldsymbol{\rho}$ を導入する．具体的には，事前分布 $\hat{\boldsymbol{\beta}}^{(t)}$ は，下記の通り生成される．

$$\hat{\boldsymbol{\beta}}^{(t)} = (\boldsymbol{1} - \boldsymbol{\rho}^{(t)})\boldsymbol{\beta}^{(t-1)} + \boldsymbol{\rho}^{(t)}\boldsymbol{\beta}^{(t)} \tag{3.1}$$

ここで，$\boldsymbol{\rho}^{(t)}$ は，過去の状態を元に時系列に生成される．各時刻において，どのように最適な $\boldsymbol{\rho}^{(t)}$ を導出するかについては，3.2.3 項において述べる．

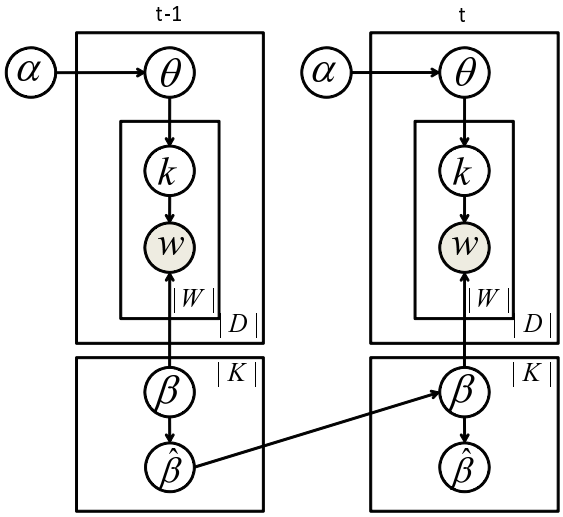

図 3.2: DTM のグラフィカルモデル

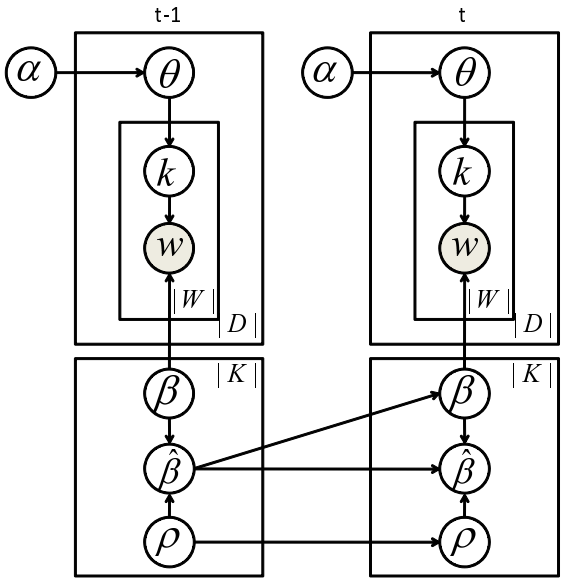

図 3.3: 提案方式のグラフィカルモデル

[28] では，提案方式と同様に，平滑化による事前分布の生成手法が提案されている．彼らは，平滑化のパラメータを，あらかじめ固定的に設定している．一方で，提案方式は，各時刻において，逐次最適なパラメータを導出する点で，彼ら

の手法とは異なる．

3.2.3　最適な平滑化パラメータの導出アルゴリズム

$\boldsymbol{\rho}^{(t)}$ は，過去と現在のモデルを平滑化するためのハイパーパラメータであり，現在のモデルが次の時刻のモデルにどの程度影響を与えるかを決定する．定性的には，モデルの変化が著しい場合には，$\boldsymbol{\rho}^{(t)}$ を大きくし，現在のモデルの影響を強くすれば良い．

提案方式は，最適な平滑化を行うために，パーティクルフィルタ [42] を適用する．ここで，平滑化の最適性は，平滑化したモデルによる，未観測文書の生成予測精度により評価される．パーティクルフィルタを利用することで，未観測文書の生成尤度を最大化する平滑化パラメータを導出することが可能であり，これにより最適な平滑化を実現可能である．

パーティクルフィルタは，各時刻におけるハイパーパラメータの候補として多数の粒子を用意し，各粒子の現在時刻の状態に対する尤度を計算する．次に，これらの粒子を，尤度の重み付きで平均することにより，当該時刻における最適なハイパーパラメータを推定する．

図 3.4 に，パーティクルフィルタによる $\boldsymbol{\rho}^{(t)}$ 導出アルゴリズムの概要を示す．あらかじめ，$\boldsymbol{N}$ 個の粒子 $\{\boldsymbol{\rho}_i^{(t-1)}\}_{i=1}^{N}$ と，対応する尤度 $\{\boldsymbol{w}_i^{(t-1)}\}_{i=1}^{N}$（$\Sigma_{i=1}^{N}\boldsymbol{w}_i^{(t-1)} = \mathbf{1}$）が与えられているものとする．そして，パーティクルフィルタのプロセスに従い，下記の順序で $\boldsymbol{\rho}^{(t)}$ を導出する．

1. 各粒子を，対応する $\boldsymbol{w}_i^{(t-1)}$ を確率として，$\boldsymbol{N}$ 回，重複を許してサンプリングする．

2. サンプリングされた各粒子を一定距離ランダムウォークさせることで，$\boldsymbol{\rho}_i^{(t)}$ を導出する．提案方式は，各粒子の動作距離を，標準偏差 $\boldsymbol{\sigma}$ の正規分布により発生させる．

3. 各粒子の尤度 $\boldsymbol{w}_i^{(t)}$ を導出する．提案方式による尤度の導出については後述する．

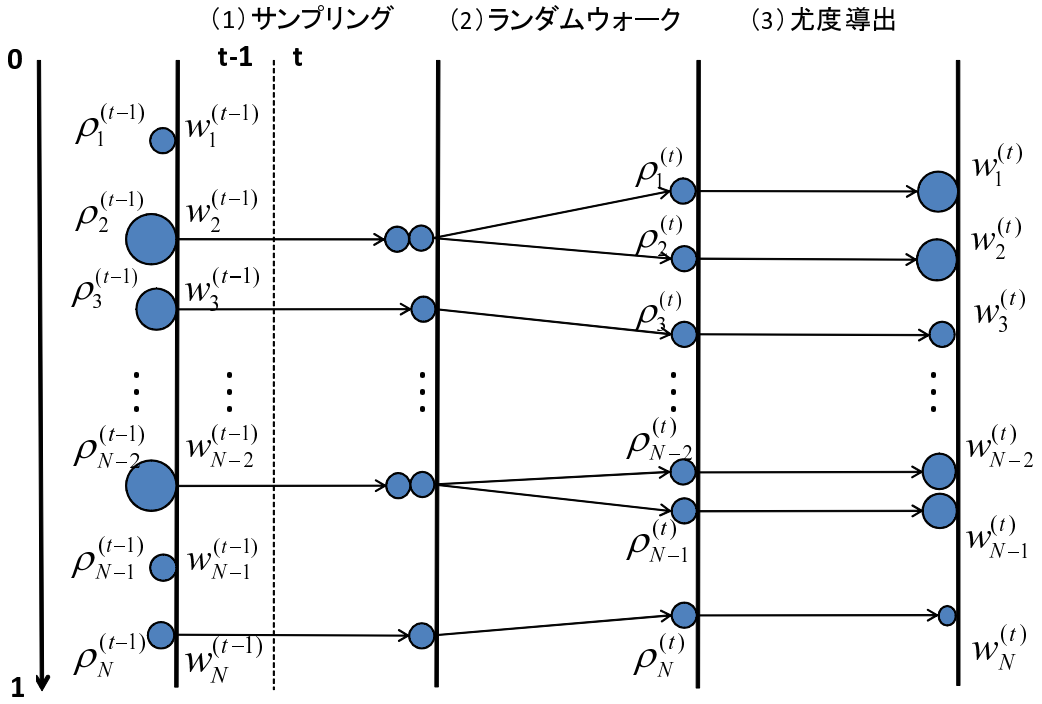

図 3.4: パーティクルフィルタによる最適平滑化パラメータの推定

4. $\boldsymbol{\rho}^{(t)}$ を各粒子の尤度による重み付平均値 $\Sigma_{i=1}^{N}\boldsymbol{\rho}_i^{(t)}\boldsymbol{w}_i^{(t)}$ として導出する．

パーティクルフィルタのハイパーパラメータは，各粒子の現在の状態に対する尤度を元に，それら粒子の尤度を重みとした重み付き平均値として導出される必要がある．これに従い提案方式は，過去のモデルを粒子 $\boldsymbol{\rho}_i^{(t)}$ により平滑化したモデルを生成し，当該モデルによる時刻 $\boldsymbol{t}$ における文書集合の生成尤度 $\boldsymbol{p_i(D^{(t)})}$ を，当該粒子の尤度とする．$\boldsymbol{p_i(D^{(t)})}$ は，具体的には下記の通り与えられる．

$$\boldsymbol{p_i(D^{(t)}) = \Pi_{d^{(t)}\in D^{(t)}}\Pi_{w\in W^{(t)}}p(w|d^{(t)})^{n_{dw}^{(t)}}} \tag{3.2}$$

$$\boldsymbol{p(w|d^{(t)}) = \Sigma_{k\in K}p(w|k)p(k|d^{(t)})} \tag{3.3}$$

ここで提案方式は，短期間ではトピック変化の度合いは変化せず，直前の時刻で最適化された平滑化係数が，現在時刻においても最適であるという仮定をおく．この仮定に基づくと，各トピックに帰属する単語の発生確率分布 $\boldsymbol{p(w|k)}$ は，$\boldsymbol{\rho_i^{(t)}}$

により平滑化したモデルから生成可能である．すなわち，時刻 $t-2$ におけるモデルと，時刻 $t-1$ におけるモデルを，式 (3.1) を利用して，$\rho_i^{(t)}$ により平滑化することで導出される．一方で，$p(k|d^{(t)})$ は，時刻 t における各文書のトピックによる表現であり，確率分布 $p(w|k)$ を文書 $d^{(t)}$ へ適用した結果として，静的 LDA と同様に導出される．

全ての粒子に対して $p_i(D^{(t)})$ が導出された後，尤度 $w^{(t)}$ の各要素は下記の通り，$p_i(D^{(t)})$ を L1 ノルムで正規化したベクトルとして与えられる．

$$w_i(t) = \frac{p_i(D^{(t)})}{\Sigma_{i=1}^{N} p_i(D^{(t)})} \tag{3.4}$$

パーティクルフィルタの性質に従えば，提案方式は，過去のモデルから時刻 t の文書を推定する際の，最適な平滑化パラメータ $\rho^{(t)}$ を推定可能となる．連続した時間においては，時間経過によるモデルの変化の大きさにそれほど差がないと仮定すれば，得られた平滑化パラメータにより，次の時刻のモデルをロバストに生成可能となる．

なお，提案方式は，過去二つの時刻のモデルを利用して，最適な平滑化パラメータを学習する．従って，過去二つのモデルの間に急激な変化があった場合には，最適な平滑化パラメータを学習できない可能性がある．この影響については，3.4 節で述べる．

平滑化係数の導出に際し，尤度関数の計算による実行時間のオーバヘッドが懸念される．パーティクルフィルタは，各パーティクルについて，尤度を独立に導出することが可能である．そこで，本研究では，平滑化係数導出の際，並列分散処理により各尤度を導出することで，オーバヘッドの削減を行う．これについては，3.4.3 項において評価する．

3.3　動的語彙集合に対応した時系列モデル

3.3.1　動的語彙集合への対応

話題変化が著しい文書では，新しく生み出された話題に対応して，それまでの文書集合で観測されなかった新語が出現する．例えば，図 3.1 によれば，2011 年

11 月から 2012 年 5 月までの，1 日 100 万文書のツイートを観測した結果，半年が経過した 2012 年 5 月でも，一定の新語が出現し続けることが分かる．語彙集合に存在しない新語が発生した場合，固定的な語彙集合では，新語に対応したモデルを生成することが出来ず，新語を含む文書の予測精度は悪化してしまう．また，どのような新語が出現するか知ることは不可能であるため，事前に全ての新語を把握し，語彙集合を生成していくことも出来ない．そのため，時系列に出現した新語に対応して，語彙集合も時系列に拡大する必要がある．

本研究で提案するシステムは，語彙集合の動的な拡大を実現する．提案方式は，時系列に出現した新語に応じて動的に語彙集合を拡大する．これにより，新語を含む文書集合についてモデル化が可能となり，固定的な語彙集合しか持たないシステムに対して，より精度の良いモデルを生成可能となる．また，提案方式は，新語の追加だけではなく，使われなくなった単語を語彙集合から削除することで，モデルや語彙集合の無制限の拡大によるシステムの性能劣化を防ぐ．

提案方式における動的な語彙集合の実現について，表 3.2 の記法，及び図 3.5 を元に示す．まず，時刻 $\boldsymbol{t}$ における語彙集合 $\boldsymbol{W}^{(t)}$ を，以下の通り生成する．

$$\boldsymbol{W}^{(t)} = \boldsymbol{W}_{\boldsymbol{u}}^{(t)} \sqcup \boldsymbol{W}_{\boldsymbol{o}}^{(t)} \tag{3.5}$$

ここで，$\boldsymbol{W}_{\boldsymbol{u}}^{(t)}$ は時刻 $\boldsymbol{t}$ において観測された新語の集合であり，$\boldsymbol{W}_{\boldsymbol{o}}^{(t)}$ は $\boldsymbol{W}^{(t-1)}$ に含まれていた既知の単語の集合である．これらの語彙集合をいかに生成するかについては，3.3.2 項で述べる．

次に提案方式は，語彙集合の $\boldsymbol{W}^{(t-1)}$ から $\boldsymbol{W}^{(t)}$ への更新に対応するため，時刻 $\boldsymbol{t-1}$ で生成された事前分布である $\hat{\boldsymbol{\beta}}^{(t-1)}$ を更新し，事前分布 $\tilde{\boldsymbol{\beta}}^{(t)}$ を生成する．

さらに，更新された事前分布により，$\boldsymbol{\beta}^{(t)}$ を生成し，最後に，次の時刻の事前分布 $\hat{\boldsymbol{\beta}}^{(t)}$ を生成する．

3.3.2 語彙集合の生成

$\boldsymbol{W}_{\boldsymbol{u}}^{(t)}$ は新語の集合であり，時刻 $\boldsymbol{t}$ に生成された文書集合に含まれ，それ以前の文書集合には含まれない単語の集合である．一方で，$\boldsymbol{W}_{\boldsymbol{o}}^{(t)}$ は，時刻 $\boldsymbol{t-1}$ の語彙集合 $\boldsymbol{W}^{(t-1)}$ の部分集合として生成される．$\boldsymbol{W}^{(t-1)}$ に含まれる単語は，時刻 $\boldsymbol{t-1}$

表 3.2: 3.3 節で使用する記法

記号	定義
$\boldsymbol{W^{(t)}}$	時刻 $\boldsymbol{t}$ における語彙集合
$\boldsymbol{W_u^{(t)}}$	$\boldsymbol{W^{(t)}}$ の部分集合であり，時刻 $\boldsymbol{t}$ における新語の集合
$\boldsymbol{W_o^{(t)}}$	$\boldsymbol{W^{(t)}}$ の部分集合，かつ $\boldsymbol{W^{(t-1)}}$ の部分集合
$\hat{\boldsymbol{\beta}}_{\boldsymbol{u}}^{(\boldsymbol{t})}$	$\hat{\boldsymbol{\beta}}^{(\boldsymbol{t})}$ のうち，$\boldsymbol{W_u^{(t)}}$ に対応するモデル
$\hat{\boldsymbol{\beta}}_{\boldsymbol{o}}^{(\boldsymbol{t})}$	$\hat{\boldsymbol{\beta}}^{(\boldsymbol{t})}$ のうち，$\boldsymbol{W_o^{(t)}}$ に対応するモデル
$\tilde{\boldsymbol{\beta}}^{(\boldsymbol{t})}$	$\boldsymbol{\beta^{(t)}}$ を元に生成される新たな事前分布
$\tilde{\boldsymbol{\beta}}_{\boldsymbol{u}}^{(\boldsymbol{t})}$	$\tilde{\boldsymbol{\beta}}^{(\boldsymbol{t})}$ のうち，$\boldsymbol{W_u^{(t)}}$ に対応する事前分布
$\tilde{\boldsymbol{\beta}}_{\boldsymbol{o}}^{(\boldsymbol{t})}$	$\tilde{\boldsymbol{\beta}}^{(\boldsymbol{t})}$ のうち，$\boldsymbol{W_o^{(t)}}$ に対応する事前分布

のモデル $\boldsymbol{\beta^{(t-1)}}$ を構成し，時刻 $\boldsymbol{t}$ のモデルの精度に寄与するため，モデルの精度の面からは，全ての単語を $\boldsymbol{W_o^{(t)}}$ へ含めるのが理想的である．しかし，一切の単語を削除せずに新語を追加していった場合，語彙集合やモデルサイズの増大によるメモリ使用量の増加など，システムの性能の劣化が懸念される．そのため，提案方式では，$\boldsymbol{W^{(t-1)}}$ に含まれる単語のうち，モデルの精度への貢献が低い単語を $\boldsymbol{W_o^{(t)}}$ から削除する．

具体的には，提案方式は，$\boldsymbol{W^{(t-1)}}$ に含まれる単語のうち，時刻 $\boldsymbol{t-1}$ で生成されたモデルにおいて，各トピックに登場する回数が一定値に満たないものを $\boldsymbol{W_o^{(t)}}$ から削除する．すなわち，$\boldsymbol{W_o^{(t)}}$ は以下の通り表わされる．

$$\boldsymbol{W_o^{(t)} = \{w | w \in W^{(t-1)} \cap \max(\{\beta_{kw}^{(t-1)}\}_{k=1}^{K}) \geq \kappa\}} \tag{3.6}$$

ここで，$\boldsymbol{\max(\{\beta_{kw}^{(t-1)}\}_{k=1}^{K})}$ は，各単語 $\boldsymbol{w}$ の各トピック $\boldsymbol{k}$ における出現頻度 $\boldsymbol{\beta_{kw}^{(t-1)}}$ の最大値を表し，$\boldsymbol{\kappa}$ はあらかじめ与えられる閾値である．

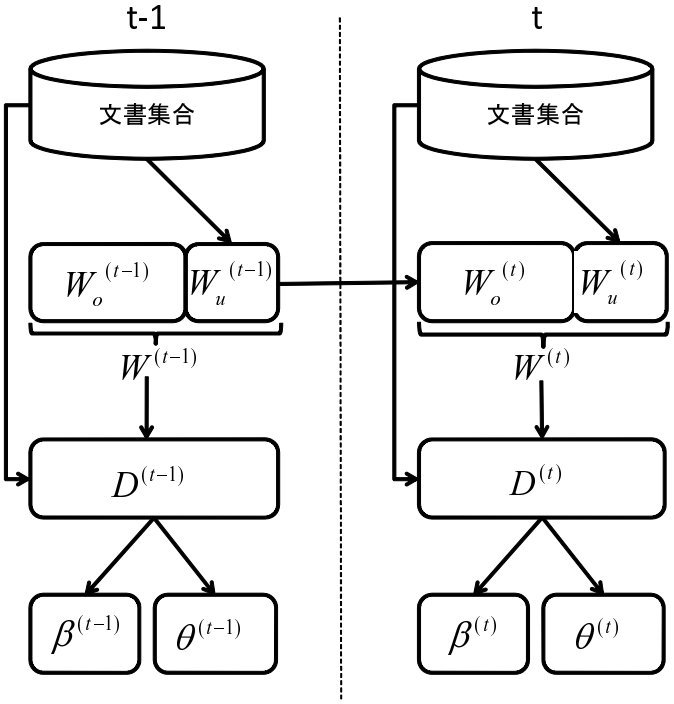

図 3.5: 提案システムの概要

3.3.3 事前分布の更新

提案方式による平滑化を行うためには，$\hat{\boldsymbol{\beta}}^{(t-1)}$ と $\boldsymbol{\beta}^{(t)}$ は，ともに同じ語彙集合により表現されなければならない．しかし語彙集合が動的に変化するため，$\hat{\boldsymbol{\beta}}^{(t-1)}$ の語彙集合 $\boldsymbol{W}^{(t-1)}$ は，$\boldsymbol{W}^{(t)}$ と異なっている．そこで提案方式は，語彙集合の更新に伴い，時刻 $\boldsymbol{t-1}$ の事前分布 $\hat{\boldsymbol{\beta}}^{(t-1)}$ を更新する．

図 3.6 に概要を示す．ここでは新しく，更新された事前分布 $\tilde{\boldsymbol{\beta}}^{(t)}$ を定義する．$\tilde{\boldsymbol{\beta}}^{(t)}$ は，新たに生成される分布 $\tilde{\boldsymbol{\beta}}_{\boldsymbol{u}}^{(t)}$ と $\tilde{\boldsymbol{\beta}}_{\boldsymbol{o}}^{(t)}$ との連結により生成される．前者は，新たに生成される事前分布のうち，新語集合 $\boldsymbol{W}_{\boldsymbol{u}}^{(t)}$ に対応する分布である. 新語は，時刻 $\boldsymbol{t}$ により初めて生成されたため，各トピックに対する分布は不明である．そこで提案方式は，各トピックの新語による分布は，すでに生成されている時刻 $\boldsymbol{t-1}$ の事前分布 $\hat{\boldsymbol{\beta}}^{(t-1)}$ を参照し，既知の単語の分布の平均値による一様分布で与えられるものと仮定する．

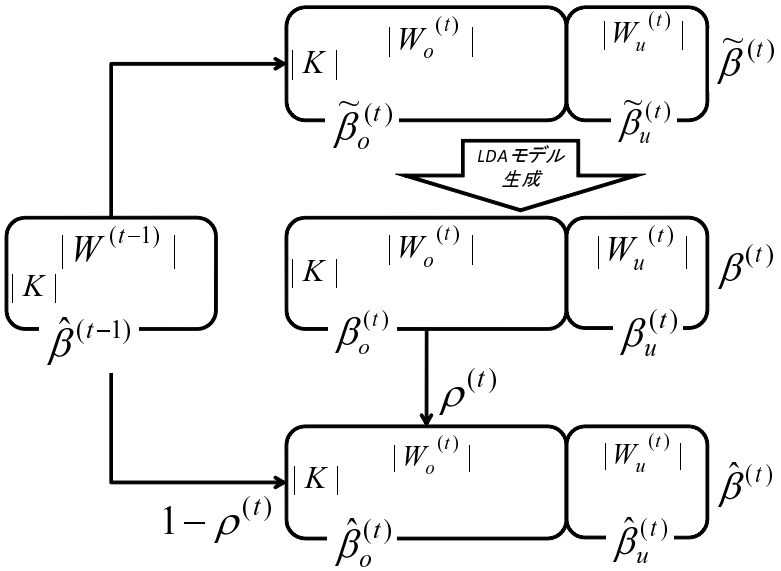

図 3.6: 事前分布の更新プロセスの概要

一方で後者の $\tilde{\boldsymbol{\beta}}_{\boldsymbol{o}}^{(\boldsymbol{t})}$ は，既知の単語集合 $\boldsymbol{W}_{\boldsymbol{o}}^{(\boldsymbol{t})}$ に対応する分布であり，$\boldsymbol{\beta}^{(\boldsymbol{t-1})}$ から，対応する単語の分布を抽出することで生成可能である．従って，時刻 $\boldsymbol{t}$ における語彙集合に含まれる全ての単語 $\boldsymbol{w_u} \in \boldsymbol{W}_{\boldsymbol{u}}^{(\boldsymbol{t})}$，及び $\boldsymbol{w_o} \in \boldsymbol{W}_{\boldsymbol{o}}^{(\boldsymbol{t})}$ について，時刻 $\boldsymbol{t-1}$ の事前分布を下記の通り更新する．

$$\tilde{\boldsymbol{\beta}}_{\boldsymbol{kw_u}}^{(\boldsymbol{t})} = \frac{\sum_{\boldsymbol{w} \in \boldsymbol{W}^{(\boldsymbol{t-1})}} \hat{\boldsymbol{\beta}}_{\boldsymbol{kw}}^{(\boldsymbol{t-1})}}{|\boldsymbol{W}^{(\boldsymbol{t-1})}|},\ \tilde{\boldsymbol{\beta}}_{\boldsymbol{kw_o}}^{(\boldsymbol{t})} = \hat{\boldsymbol{\beta}}_{\boldsymbol{kw_o}}^{(\boldsymbol{t-1})} \tag{3.7}$$

ここで，$\tilde{\boldsymbol{\beta}}_{\boldsymbol{kw}}^{(\boldsymbol{t})}$ は $\tilde{\boldsymbol{\beta}}^{(\boldsymbol{t})}$ の各要素を表す．提案方式は，上記更新された事前分布を元に，時刻 $\boldsymbol{t}$ におけるモデル $\boldsymbol{\beta}^{(\boldsymbol{t})}$ を生成する．事前分布に対応し，$\boldsymbol{\beta}^{(\boldsymbol{t})}$ は，$\boldsymbol{\beta}_{\boldsymbol{u}}^{(\boldsymbol{t})}$ と $\boldsymbol{\beta}_{\boldsymbol{o}}^{(\boldsymbol{t})}$ との連結によって表現される．

最後に提案方式は，平滑化により次の時刻のための事前分布 $\hat{\boldsymbol{\beta}}^{(\boldsymbol{t})}$ を生成する．提案方式は $\tilde{\boldsymbol{\beta}}^{(\boldsymbol{t})}$ 生成時と同様に，$\hat{\boldsymbol{\beta}}^{(\boldsymbol{t})}$ を，語彙集合 $\boldsymbol{W}_{\boldsymbol{u}}^{(\boldsymbol{t})}$ に対応する分布 $\hat{\boldsymbol{\beta}}_{\boldsymbol{u}}^{(\boldsymbol{t})}$ と $\boldsymbol{W}_{\boldsymbol{o}}^{(\boldsymbol{t})}$ に対応する $\hat{\boldsymbol{\beta}}_{\boldsymbol{o}}^{(\boldsymbol{t})}$ との連結により生成する．提案方式は，前者については, 単純に $\boldsymbol{\beta}_{\boldsymbol{u}}^{(\boldsymbol{t})}$ と同じ分布とする. 一方で後者は，同じ語彙集合で表現される $\boldsymbol{\beta}_{\boldsymbol{o}}^{(\boldsymbol{t})}$ と $\hat{\boldsymbol{\beta}}_{\boldsymbol{o}}^{(\boldsymbol{t-1})}$ との平滑化により生成する．すなわち，提案方式は，式 (3.1) を更新し，下記の式

に従って，平滑化を行う．

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{u}}^{(t)} = \boldsymbol{\beta}_{\boldsymbol{u}}^{(t)}, \hat{\boldsymbol{\beta}}_{\boldsymbol{o}}^{(t)} = (\boldsymbol{1} - \boldsymbol{\rho}^{(t)})\hat{\boldsymbol{\beta}}_{\boldsymbol{o}}^{(t-1)} + \boldsymbol{\rho}^{(t)}\boldsymbol{\beta}_{\boldsymbol{o}}^{(t)} \tag{3.8}$$

3.3.4 無限の語彙集合拡大に対応した実装方式

本研究では，提案方式を，[28] で提案された，LDA のオンライン学習アルゴリズムを拡張して実装した．彼らの LDA アルゴリズムは，変分ベイズ法に基づく実装であり，$\boldsymbol{\theta}$ と $\boldsymbol{\beta}$ が収束するまで，以下の過程を繰り返し実行する．

$$\phi_{dwk} \propto \exp\{\boldsymbol{E}_q[\log \boldsymbol{\theta}_{dk}] + \boldsymbol{E}_q[\log \boldsymbol{\beta}_{kw}]\} \tag{3.9}$$

$$\gamma_{dk} = \boldsymbol{\alpha} + \Sigma_w \phi_{dwk} \boldsymbol{n}_{dw} \tag{3.10}$$

ここで，ϕ_{dwk} は，文書毎のトピックの単語分布であり，$\boldsymbol{E}_q[\log \boldsymbol{\theta}_{dk}]$，及び $\boldsymbol{E}_q[\log \boldsymbol{\beta}_{kw}]$ は，以下の式により更新される．

$$\boldsymbol{E}_q[\log \boldsymbol{\theta}_{dk}] = \boldsymbol{\Psi}(\gamma_{dk}) - \boldsymbol{\Psi}(\Sigma_{i=1}^{|\boldsymbol{K}|}\gamma_{di}) \tag{3.11}$$

$$\boldsymbol{E}_q[\log \boldsymbol{\beta}_{kw}] = \boldsymbol{\Psi}(\boldsymbol{\lambda}_{kw}) - \boldsymbol{\Psi}(\Sigma_{i=1}^{|\boldsymbol{W}|}\boldsymbol{\lambda}_{ki}) \tag{3.12}$$

なお，$\boldsymbol{\Psi}$ は，ディガンマ関数である．収束後は，$\boldsymbol{\gamma}$ を $\boldsymbol{\theta}$ とし，$\boldsymbol{\lambda}$ を $\boldsymbol{\beta}$ とする．

上記の式は，単純な行列ベースの実装では，語彙サイズ $|\boldsymbol{W}|$ に比例して，計算回数が増加していくため，語彙集合の無制限の拡大を許容する提案方式では，計算時間を抑制するための手段が必要となる．

そこで本研究では，提案方式を，各文書を出現単語とその頻度の key-value 形式で表現することで実装する．これにより，上記の二つの過程は，各文書の単語の登場頻度 $|\boldsymbol{W_d}|$ に比例した計算回数で実行されることとなる．$|\boldsymbol{W_d}|$ は特に，Twitter のように文書の長さが一定長に保たれるような文書集合では，$|\boldsymbol{W}|$ よりはるかに小さな値となり，語彙集合のサイズが無限に増大していく場合でも，計算時間をほぼ一定に保つことが可能となる．

3.4　性能評価

本節では，Twitter の文書を蓄積した二つのデータセットを利用し，提案方式の評価を行う．評価では，時系列に生成される新規文書を，精度よくモデル化可能であることを示す．ある時点でのモデルの精度は，次の時刻に生成される文書をテストデータとして，パープレキシティにより評価する．また，実際のホットトピック抽出においても提案方式が有効であることを，KL ダイバージェンス [48] に基づく評価により示す．さらに提案方式が，計算時間において，語彙集合の拡大に対してスケーラブルであることを示す．

3.4.1　データセット

本研究では，提案方式の評価に際し，Twitter をクローリングすることで，二つのデータセットを用意した．データセット 1 は，2012 年 11 月から 2012 年 5 月まで，1 日 100 万の日本語ツイートであり，データセット 2 は，2011 年 2 月から 2011 年 3 月まで，1 日 100 万の日本語ツイートである．いずれのデータセットにおいても，1 ツイートを 1 文書とした．また，各文書は形態素解析を行い，形態素を単語とした bag-of-words で表現した[1]．なお，Twitter では形態素解析器の辞書に登場しない崩れた表現の単語が登場する可能性もあるため，形態素解析において名詞だけではなく，品詞が不明であった未知語についても，bag-of-words 表現に利用する単語の対象とした．また，時刻の単位は 1 日とした．従って，各時刻において，100 万文書が生成されることとなる．

3.4.2　精度評価

パープレキシティ

モデルの最適性の評価指標に用いるパープレキシティは，2.4.2 項で示した通り，以下の式で表され，各トピック中での単語の発生予測精度を示す．

[1]形態素解析器には NTT の JTAG [32] を利用している．

$$\boldsymbol{perplexity} = \exp(-\frac{\Sigma_{d \in \tilde{D}} \log \boldsymbol{p(d)}}{\Sigma_{d \in \tilde{D}} \boldsymbol{N_d}}) \tag{3.13}$$

$$\log \boldsymbol{p(d)} = \Sigma_{w \in \boldsymbol{W}} \boldsymbol{n_{dw}} \log \boldsymbol{p(w|d)} \tag{3.14}$$

ここで，$\boldsymbol{N_d}$ は各文書において観測された単語数である．$\boldsymbol{d}$ をテスト文書とし，$\boldsymbol{p(w|k)}$ を評価対象モデルから生成することで，当該モデルの文書に対する精度を示す．

以下，4 つの手法について，パープレキシティによる精度評価を示す．

- SDV（smoothing-dynamic-vocabulary）：提案方式の最適平滑化と動的語彙集合に対応した方式
- NSDV（non-smoothing-dynamic-vocabulary）：提案方式の動的語彙集合のみに対応した方式
- NSSV（non-smoothing-static-vocabulary）：平滑化，動的語彙集合に対応しない，DTM と同等の方式．新語に対応出来ないため，新語に対応する $\boldsymbol{p(w|k)}$ をランダムに与えるものとする．
- SLDA（static-LDA）：時系列変化を考慮しない静的 LDA と同等の方式

図 3.7 に，データセット 1 における 2012 年 5 月 1 日から 5 月 30 日までの，パープレキシティ評価の結果を示す．縦軸はパープレキシティを表し，横軸は 5 月 1 日を $\boldsymbol{t}$=1，2 日を $\boldsymbol{t}$=2,... 5 月 30 日を $\boldsymbol{t} = \boldsymbol{30}$ とした．各時刻におけるパープレキシティは，時刻 $\boldsymbol{t}$ において生成したモデルの，時刻 $\boldsymbol{t+1}$ に入力された文書に対する値である．ここで，LDA のハイパーパラメータは，$\boldsymbol{\alpha}$=0.01, $\boldsymbol{\eta}$=0.01 とした．また，SDV のパーティクルフィルタにおけるパラメータは，$\boldsymbol{N}$=100, $\boldsymbol{\sigma}$=0.1 とし，平滑化係数 $\boldsymbol{\rho}$ は，初期値を 1 とした．

ここで，4 手法を比較するためには，モデルが参照する語彙集合は同じものとする必要がある．そのため，NSDV と SDV において $\boldsymbol{\kappa} = \boldsymbol{0}$ と設定し，単語の削除は行わないものとした．さらに，NSSV は，新規文書の生成に対して語彙集合を拡大することが出来ないため，データセット 1 の 2011 年 11 月から 2012 年 4 月までに

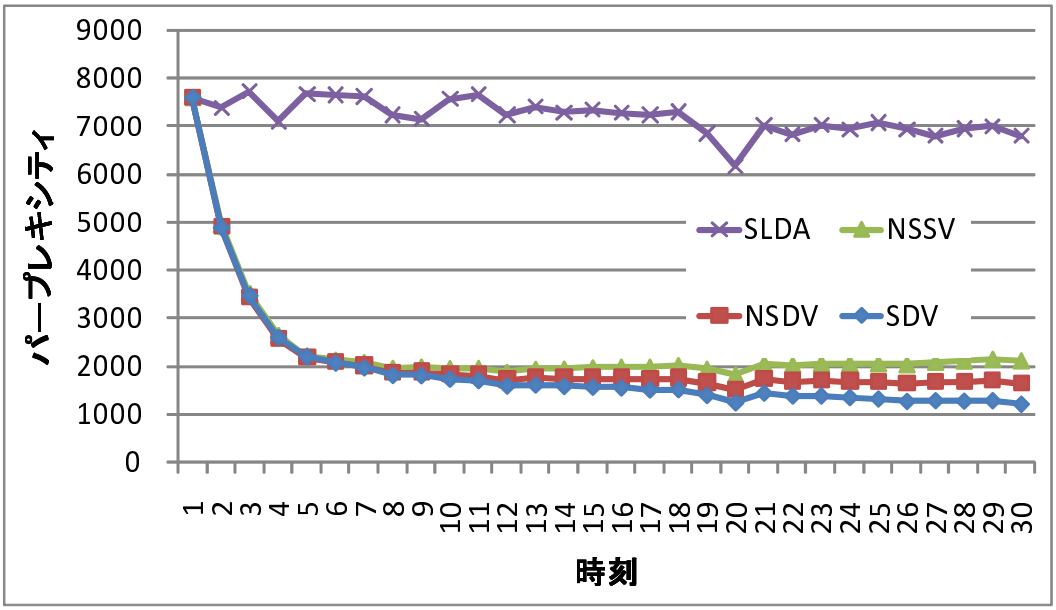

図 3.7: データセット 1 におけるパープレキシティ評価

登場した全ての単語を語彙集合に登録し，実験中は固定した．なお，2011 年 11 月から 2012 年 5 月までの登場累積単語数は，図 3.1 に示した通りである．NSSV は，2011 年 4 月までに登場した単語に対するモデルを生成する一方で，その他の手法では，各時刻で登場した単語に対するモデルを生成していく．

図に示す通り，SDV，NSDV，NSSV，SLDA の順に精度の良いモデルが生成されており，各モデル間の精度の差は，時間がたつにつれ広がっていくことが分かる．NSDV と NSSV との精度の差は，NSDV による動的語彙集合への対応に基づく．図 3.1 に示したように，日々大量の新語が発生する．これらのうちの一部は新語として定着し，その後の文書集合に継続的に登場する．そのため NSDV では，これらの新語を語彙集合に追加していくことで，時間が経過するにつれて，NSSV に対して良い精度でモデルを生成可能となる．また，SDV と NSDV では，SDV が平滑化を行うことによって，さらに時間変化にロバストなモデルを生成するために，精度の差が広がっていく．これにより，本研究で提案した平滑化アルゴリズム，動的語彙集合のシステムは，いずれも，時系列文書に対する高精度なモデルの生成に有効であることが分かる．

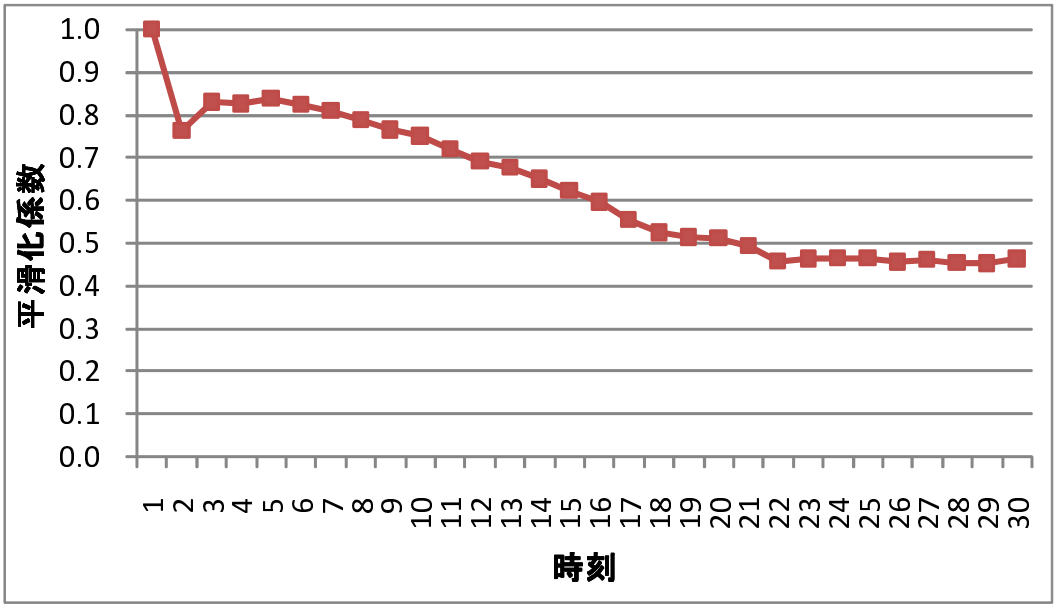

図 3.8: データセット 1 における $\boldsymbol{\rho}^{(t)}$ の変化

図 3.8 は，同条件の評価における，$\boldsymbol{\rho}^{(t)}$ の変化を示したものである．$\boldsymbol{\rho}^{(t)}$ は 1 からスタートし，減少していく．これは，時間経過により過去のモデルが蓄積されて，徐々にロバストなモデルが生成されることで，平滑化時に，過去のモデルの比重を大きくしたほうが，より精度の良いモデルを生成可能となっていくためである．

各パラメータを変化させた場合の，提案方式の性能について評価する．表 3.3 に，$\boldsymbol{\alpha}, \boldsymbol{\eta}$ を変化させた場合の，データセット 1 における 2011 年 5 月 1 日から 5 月 30 日までの，平均のパープレキシティを示す．表に示す通り，どのハイパーパラメータの組合せの場合であっても，SDV 及び NSDV は，NSSV や SLDA の性能を上回る．また，SDV の性能は，$\boldsymbol{\alpha}$=0.01, $\boldsymbol{\eta}$=0.01 の場合が最も良いため，以降の評価では，これらの値を利用した．

次に，パーティクルフィルタのパラメータであるパーティクル数 $\boldsymbol{N}$ と，ランダムウォークの距離 $\boldsymbol{\sigma}$ を変化させた場合の，平均のパープレキシティを表 3.4 に示す．表に示す通り，$\boldsymbol{N}$ を 100 より増加させても，パープレキシティはほぼ変化しない．また，$\boldsymbol{\sigma} = \mathbf{0.1}$ で最も良い性能を示すことが分かる．そこで，以降の評価で

表 3.3: LDA のハイパーパラメータによるパープレキシティの変化

$\boldsymbol{\alpha}$	0.01	0.01	0.01	0.01	0.001	0.1	1
$\boldsymbol{\eta}$	0.01	0.001	0.1	1	0.01	0.01	0.01
SDV	1943	1948	1946	1982	1952	1945	2002
NSDV	2144	2150	2148	2188	2155	2147	2210
NSSV	2371	2377	2375	2419	2383	2374	2444
SLDA	7188	7201	7296	7332	7220	7195	7405

表 3.4: パーティクルフィルタのパラメータによるパープレキシティの変化

$\boldsymbol{N}$	100	100	100	100	150	200
$\boldsymbol{\sigma}$	0	0.05	0.1	0.2	0.1	0.1
SDV	1963	1952	1943	1974	1942	1940

は，パーティクルフィルタのパラメータとして，$\boldsymbol{N = 100}$，$\boldsymbol{\sigma = 0.1}$ を利用した．

3.2.3 項で述べた通り，提案方式は，過去のモデルを利用して，時系列のモデルの変化を平滑化するパラメータを学習するため，突然，文書内の話題が切り替わった場合には，過去のモデルから学習した平滑化係数が最適とならずに，精度が下がってしまう可能性がある．そこで，このような場合の提案方式の性能を評価するため，例として，2011 年 3 月 11 日の震災前後に注目する．震災発生以降は，Twitter で，震災関連の情報交換が広く行われたため，3 月 10 日以前の Twitter とは，全く異なる多くの話題が発生したと考えられる．そこで以下では，平滑化を行った SDV と，行わない NSDV との間で，パープレキシティの変化を比較する．

図 3.9 に，データセット 2 における 2011 年 3 月 1 日から 3 月 30 日までの，SDV，及び NSDV のパープレキシティを示す．縦軸，横軸は，図 3.7 と同様であり，3 月 1 日を t=1，2 日を t=2,… 3 月 30 日を $\boldsymbol{t = 30}$ としている．図に示す通り，2011 年 3 月 11 日に発生した震災により，Twitter 上での話題が大きく変化したため，2011 年 3 月 10 日に生成されたモデルの精度が大きく低下する．ここで，平滑化を行わない NSDV は，$\boldsymbol{t = 11}$（3 月 11 日）には，当日のモデルを事前分布とすることで，

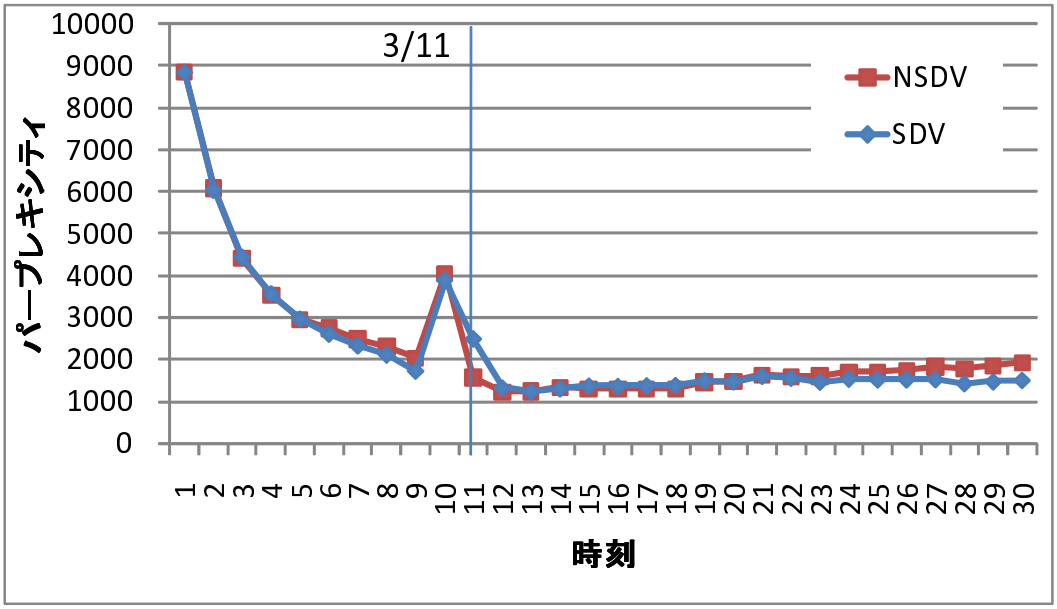

図 3.9: データセット 2 におけるパープレキシティ評価

$t = 12$ の文書集合に対するモデルの精度が向上している．

SDV の精度について考察する．同条件における $\rho^{(t)}$ の変化を，図 3.10 に示す．図に示す通り，$t = 11$ における $\rho^{(t)}$ の値が，過去のモデルの変化を反映して，低いままである．しかし実際には，$t = 11$ のモデルから $t = 12$ の文書を予測するためには，急激な話題の変化に対応して，$\rho^{(t)}$ を高くしなければならない．そのため，図 3.9 にも示す通り，SDV では，$t = 11$ におけるモデルの精度が悪化することとなる．しかし，精度の悪化は一時的であり，$t = 12$ では，地震関連で話題が安定した状態を学習可能であるため，その後は精度の良いモデルを生成可能となる．また，$t = 13$ 以降では，SDV による平滑化の効果により，時間変化にロバストなモデルが生成されていくため，時間がたつにつれて SDV の精度が NSDV 精度を上回っていく．

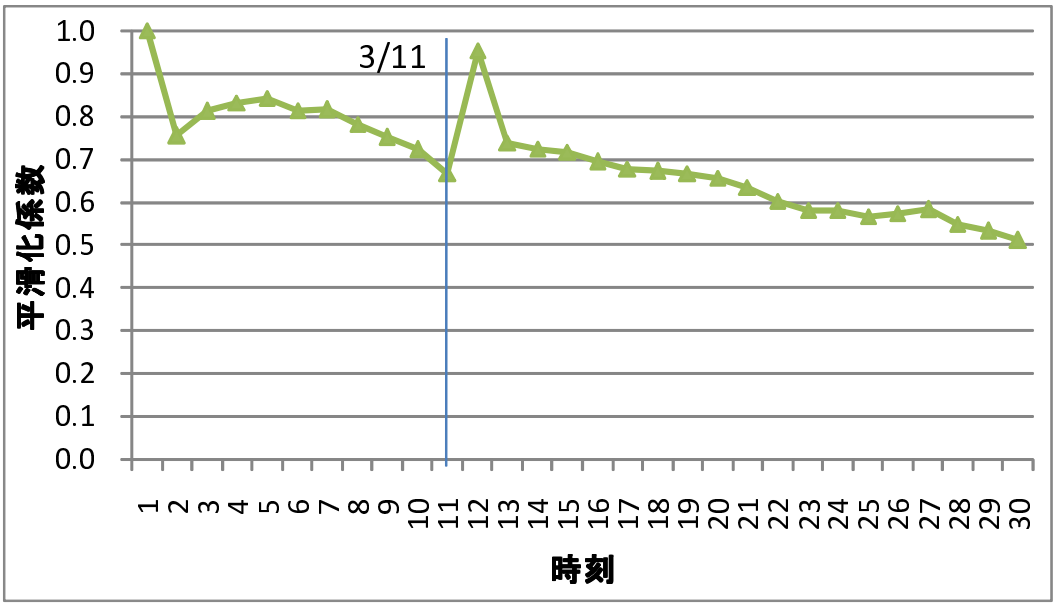

図 3.10: データセット 2 における $\boldsymbol{\rho}^{(t)}$ の変化

KL ダイバージェンス

2.4.2 項でも述べた通り，パープレキシティの評価は，モデルの最適性を示すのに有効であるが，実アプリケーションにおけるモデルの性能を評価することは出来ない．そこで本研究では，時系列文書のホットトピック抽出においても提案方式が有効であることを，KL ダイバージェンスを用いた評価により示す．

KL ダイバージェンスは，二つの確率分布の距離を表す指標で，以下の通り導出される．

$$\boldsymbol{D_{KL}(P||Q) = \Sigma_{k \in K} \log \frac{P}{Q}} \tag{3.15}$$

ここで $\boldsymbol{P}$ を $\boldsymbol{Q}$ をそれぞれ，時刻 $\boldsymbol{t}$，$\boldsymbol{t-1}$ における各トピックの単語発生分布（提案方式では $\hat{\boldsymbol{\beta}}^{(t)}$，$\hat{\boldsymbol{\beta}}^{(t-1)}$）とすると，隣り合う時刻におけるトピックの変化を定量的に示す指標となる．すなわち，この値が大きい場合は，隣り合う時刻でのトピック変化が大きいことを示す．そのため，大きなトピック変化が発生した時系列文書集合において，KL ダイバージェンスを導出した際，より大きな値が得ら

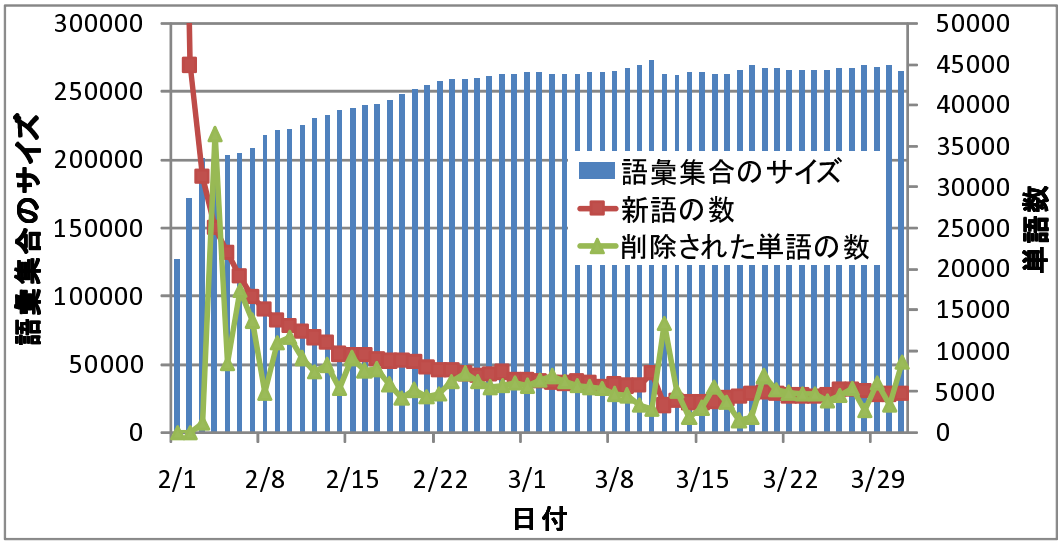

図 3.11: $\boldsymbol{\kappa = 0.1}$ における語彙集合のサイズの変化

れたモデルはよりトピックの変化に過敏に反応したことを示し，ホットトピック抽出に適したモデルであると言える．

以下では，震災により大きなトピック変化が発生したデータセット 2 において，KL ダイバージェンスによる評価結果を示す．評価対象は，SDV（単語削除無し．$\boldsymbol{\kappa = 0}$），SDV（単語削除有り．$\boldsymbol{\kappa = 0.1}$），NSDV，NSSV の 4 手法である．ここで，単語削除有りの場合の $\boldsymbol{\kappa}$ は，データセット 2 において，2011 年 2 月 1 日から 3 月 31 日までの各時刻において追加される新語と削除される単語がほぼ同じとなるように調整して設定した．$\boldsymbol{\kappa = 0.1}$ における各時刻の新語，削除された単語，及び語彙集合のサイズを図 3.11 に示す．図に示す通り，$\boldsymbol{\kappa = 0.1}$ と設定することで，新語と削除された単語の数がほぼ同一となり，語彙集合のサイズが一定となっていることが分かる．

図 3.12 は，データセット 2 における全トピックの平均 KL ダイバージェンスの時間変化を手法毎に示したものである．縦軸は KL ダイバージェンス，横軸は 3 月 2 日を t=1，3 日を t=2,… 3 月 31 日を $\boldsymbol{t = 30}$ としている．二つの SDV はほぼ同じ値となり，NSDV もわずかに値が低いもののほとんど値は変わらない．その一方で，NSSV は他 3 手法と比較すると値がかなり低くなっている．本評価における

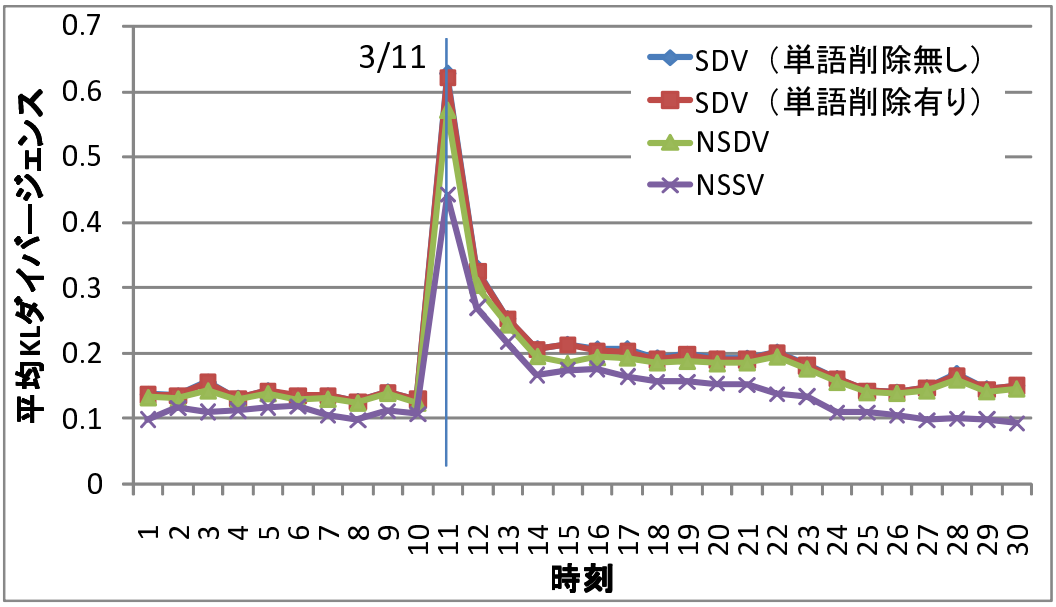

図 3.12: 全トピック平均の KL ダイバージェンスの時間変化

KL ダイバージェンスはトピックの単語発生分布の分布間距離を表すため，特にトピックの生成に重要な影響を及ぼす新語の有無に大きく影響を受ける．そのため，動的語彙集合に対応しない NSSV では，他手法と比較すると値が低くなる．これは，ホットトピック抽出のアプリケーションにおいて，提案方式の抽出精度の方がより優れていることを意味する．

一方で，単語削除無しと有りの二つの SDV は，KL ダイバージェンスがほぼ同じである．これは，各トピックにとって重要でない語を削除したとしても，ホットトピック抽出には影響を与えないことを示している．これにより，提案方式による単語の削除により，語彙集合のサイズを一定に保ちつつ，全単語を語彙集合に追加した場合と同等の性能でホットトピック抽出が可能であることが分かる．また，SDV は NSDV よりわずかながら KL ダイバージェンスが大きく，SDV の平滑化がホットトピック抽出においても有効であることが分かる．これは，SDV が，時間変化によりロバストなモデルを生成するため，トピックの急激な変化に対しては，予測モデルと実測の分布に差が生じるためである．この結果は，3.4.2 項で示した評価結果とも一致している．

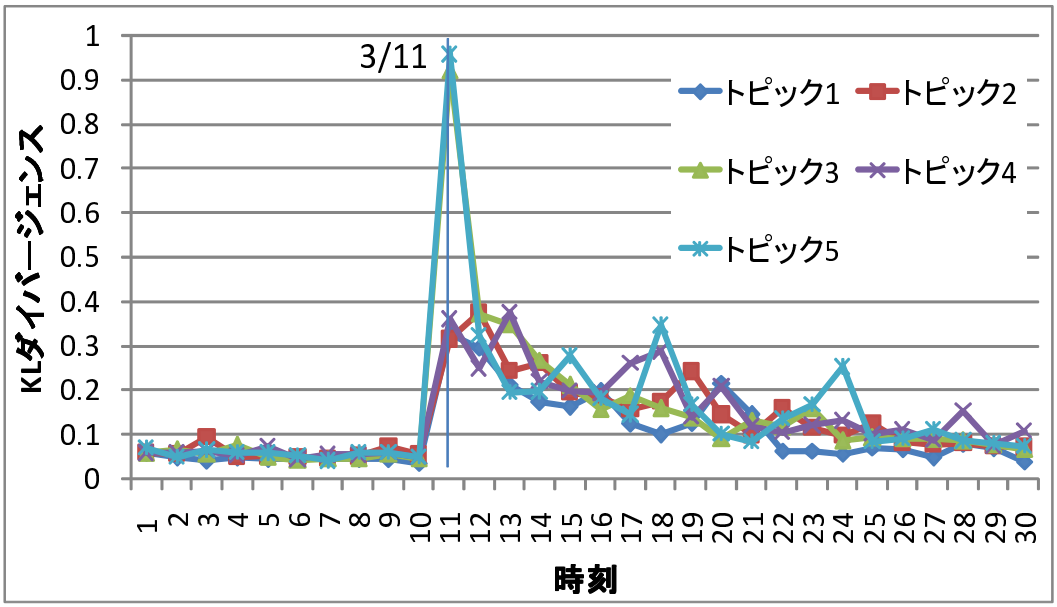

図 3.13: 各トピックにおける KL ダイバージェンスの時間変化

なお，実際にホットトピック抽出に適用するには，各トピックの KL ダイバージェンスの変化に着目すれば良い．例えば，SDV（$\kappa = 0$）において，3 月 11 日から 12 日にかけて特に KL ダイバージェンスの高かった 5 つのトピックの，KL ダイバージェンスの時間変化は図 3.13 に示した通りである．KL ダイバージェンスが高い値を示すトピックに着目することで，ホットトピックやホットワードの抽出が可能となる．

これら 5 トピックについて表 3.5 に当該トピックを構成する代表的な単語を示す．トピック 1 は，Twitter の一般的な単語が並ぶトピックであったが，震災以降は様々な情報をリツイートで展開するトピックへと変化し，大きく盛り上がっている．トピック 2 は，各地域の地震情報[2]に関するトピックであったが，震災後は大震災の話題へと変化している．トピック 3 は,「朝」,「今日」など時間に関する小さなトピックであったが，大震災後は特に電車情報に関するトピックへと変化している．トピック 4 は，人間，国民，日本人といった単語が並ぶトピックであったが，大震災後は人間の道徳的な行動を示すトピックへと変化している．トピック 5

[2]大震災数日前から小規模な地震は発生していた．

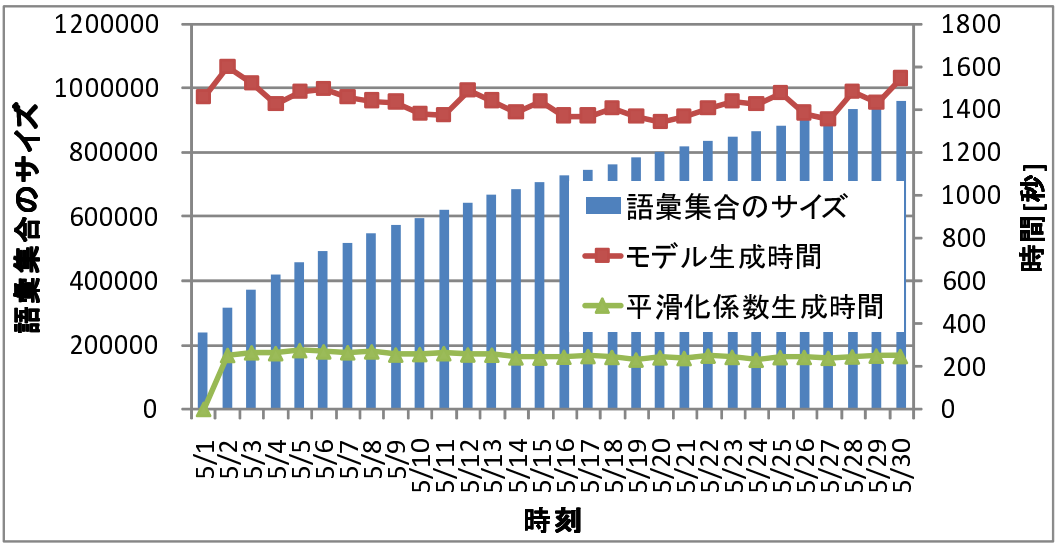

図 3.14: 計算時間と語彙集合のサイズの関係

は，一般的なインフラに関するトピックであったが，大震災後は原発事故に関するトピックへと変化し，度々盛り上がっている．これらのトピックに共通して言えるのは，大震災後に急激に変化し盛り上がっていると同時に，単語から類推されるトピックもより鮮明となり，ホットトピック，ホットワードとして抽出すべき情報であると言える点である．

3.4.3　オーバヘッド評価

提案方式の実行時間について，図 3.14 に示す．提案方式の実行時間は，主に LDA によるモデルの生成時間と，パーティクルフィルタによる平滑化係数の生成時間が大半を占める．図は，データセット 1 について，SDV の各時刻におけるモデルの計算時間，平滑化係数の生成時間，及び各時刻における語彙集合のサイズ（単語数）を示したものである．図より，語彙サイズは最終的に 90 万に達するが，その一方で，提案方式の key-value 型の実装により，モデルの生成時間はほぼ一定に保たれていることが分かる．

一方，パーティクルフィルタの精度は粒子数が多いほど良く，計算時間は粒子

数に比例する．3.4.2 項で示した通り，本評価では，$N = 100$ より粒子数を増加させても精度はほぼ変化しないため，$N = 100$ における平滑化係数の生成時間を評価した．ここで，3.2.3 項で述べた通り，平滑化係数生成時の各粒子に対する尤度は，並列分散処理により導出が可能である．本実験では，粒子数 $N = 100$ に対して，分散数を 10 とすることで平滑化係数を導出した．その結果，図に示す通り，平滑化係数の生成時間を 200 秒程度となった．すなわち，SDV は，他の手法と比較して精度の良いモデルを生成可能となる一方で，計算時間については，10%程度のオーバヘッドが発生することとなる．

3.5 むすび

本章では，トピックモデルを拡張することで，ユーザの意図が時事刻々と反映される時系列文書を高精度にモデル化する手法を提案した．提案方式は，パーティクルフィルタを適用し，各時刻における LDA のパラメータを最適に平滑化し，時間変化にロバストなモデルの生成を可能とした．また，動的な語彙集合を実現し，key-value 型の実装により，モデル生成時のスケーラビリティを確保した．さらに，合計 9ヶ月間 1 日 100 万ツイートのデータセットを対象とした大規模な実験を行い，提案方式が精度良く時系列文書をモデル化することを，パープレキシティ評価により示し，ホットトピック抽出への有効性も示した．また，提案方式の語彙集合拡大に対するスケーラビリティを示した．

本章では，特に Twitter データを文書集合の対象としたが，Twitter に特有の特徴は一切利用していない．また，Twitter に限らず，時系列に文書が蓄積されていく過程で，新語が登場するのは自然である．そのため，本章で提案した手法は，あらゆる文書集合に対する時系列分析において有効であると言える．

今後の課題として，提案方式によるホットトピックの抽出を応用した，実アプリケーションへの適用が考えられる．また，2 章の手法を組み合わせた時系列のコミュニティ分析を行うことが考えられる．

表 3.5: 震災前後の代表的なホットトピックと単語

	3 月 9 日	3 月 10 日	3 月 11 日	3 月 12 日	3 月 13 日
トピック 1	お願い	お願い	お願い	お願い	お願い
	フォロー	フォロー	拡散希望	拡散希望	拡散希望
	Twitter	Twitter	RT	RT	RT
	ありがとう	ありがとう	情報	情報	情報
	フォロワー	フォロワー	フォロー	協力	協力
トピック 2	地震	東京	地震	地震	地震
	東京	地震	避難	避難	情報
	大阪	大阪	東京	破壊	避難
	渋谷	渋谷	津波	東京	被災者
	北海道	北海道	避難所	津波	破壊
トピック 3	今日	朝	朝	無事	無事
	朝	今日	無事	全線	元気
	元気	一日	今日	朝	朝
	勉強	勉強	全線	元気	全線
	一日	一日	一日	#jisin	#jisin
トピック 4	必要	必要	必要	必要	必要
	人間	人間	非難	医療	医療
	人々	日本人	人間	日本人	行動
	主張	人々	行動	行動	日本人
	日本人	人類	重要	非難	人々
トピック 5	夜	夜	電気	原発	原発
	電気	電気	夜	電気	電気
	パソコン	PC	ガス	夜	ガス
	ガス	ガス	安全	安全	夜
	便利	便利	原発	ガス	安全

第4章　対話システムによる発話意図理解

4.1　まえがき

スマートフォンの普及に伴い，Siri [90] やしゃべってコンシェル [86] など，ユーザが音声で操作可能な対話システムへの注目が高まっている．対話システムは，ユーザの自然な音声発話をエージェントが理解し，その結果に基づきユーザの望む処理を行う．これは，ユーザの発話を単一文書と見なした場合に，単一文書からユーザの発話意図を理解する問題に帰着する．本章では，分類 3（単一文書に対する非時系列分析技術）の技術として，発話を行ったユーザの意図を高精度に理解する手法を提案し，実システム上での有効性について示す．

対話システムの例として，対話で操作可能なカーナビゲーションシステム（以下，カーナビ）を考える．自動車の運転には両手を使用することから，ハンズフリーでカーナビを操作するための対話システムに対する需要は大きい．実際に，最近のほとんどのカーナビは，音声認識機能を備えることで，ユーザの発話に応じて，さまざまな処理を行うことが可能である [40] [62]．しかし，最近のカーナビに搭載された対話システムは，そのほとんどが，あらかじめ定められた定型の発話文に対して特定の処理を行う，音声コマンド方式を採用している [45]．音声コマンド方式では，ユーザが各処理を実行するための定型の発話文を覚えなければならない．例えば，渋滞情報を表示する処理に対して，ユーザは「渋滞情報検索」といったあらかじめ定められた定型文を覚え，発話しなければならない．しかし，多くのユーザにとっては，これらの定型文を覚えることは困難であり，カーナビの音声認識機能が広く利用されない要因の一つとなっている [45]．発話の誤認識を防ぐために，定型文に不要となる語を発話から除去することにより精度を向上

するシステム [14] [40] もあるが，定型文を覚えなければならないことには変わりはなく，根本的な問題の解決には至っていない．

これを受けて，本研究では，特にユーザの自然で多様な発話を対象として，ユーザの発話から対話システムが実行すべき処理を高精度に特定可能とする手法を提案する．以下では，対話システムが実行すべき処理をタスクと呼び，発話から適切なタスクを特定する処理をタスク判定と呼ぶ．本研究の目的は，タスク判定を高精度に行う手法の確立である．ここで精度は，入力された発話からいかに適切なタスクを実行したかによって評価されるものとする．

提案方式を適用することで，対話システムはユーザの自然な発話の意図を理解し，これまでの対話システムでは処理が困難であった自然な発話に対して，適切な処理を実行することが可能である．例えば，上で示した渋滞情報を表示する処理において，定型の「渋滞情報検索」だけではなく,「渋滞あるかな」や「道路混んでる」といった話し言葉に近い自然な発話に対しても，その意図を正しく理解し，処理を実行可能である．

タスク判定は，発話として入力された文を，あらかじめ定められたいずれかのタスクに振り分ける．上記の渋滞情報の例では，渋滞情報を検索するタスクとして判定する．あるいは,「東京スカイツリーに行きたい」という発話文は，目的地設定のタスクとして判定する[1]．これは，ユーザの発話を入力文書と見なして，文書をいずれかのタスクに分類する文書分類問題と見なすことができる．近年では特に，機械学習による文書分類手法が多く提案されている [4] [39] [60] [71]．提案方式は，多くの分類問題において優れた汎化性能を持つ SVM（Support Vector Machine）[39] を利用して，タスク判定を実現する．なお，機械学習を利用した文書分類手法として，最大エントロピー法 [4]，ナイーブベイズ [60] などを利用した手法も提案されているが，本章では学習器の選択に関しては議論しない．

一般的に，機械学習による文書分類では，各文書からどのように機械学習の特徴を生成するかによって，精度に大きな違いが発生する [34] [71]．以下では，文書からあるルールを以って抽出された要素を素性と呼び，各素性に重みを与えることで生成されるベクトルを特徴ベクトルと呼ぶ．例えば，単純な特徴ベクトルは,

[1]対話型カーナビシステムにおけるタスクの一例は，表 4.1，4.2 に示す通りである．

素性を文書中に含まれる単語とし，各単語の出現回数を重みとして生成される．これに対して，単語の N-gram や上位概念といった素性を組み合わせることで，文書分類の精度が向上することが知られている [9] [26] [70]．そこで本研究では，特に短文となる発話文の特徴に着目した上で，タスク判定を実現するにあたり様々な素性や重みを組み合わせた特徴ベクトルを比較し，最も良い精度でタスク判定が可能な特徴ベクトルを実験的に探索した．

以上で述べた通り，入力されたユーザの発話を高精度に理解するタスク判定手法を提案することが本章における主題となる．具体的には以下の通りである．

1. 自然な発話を高精度にタスク判定可能な手法を提案する．提案方式は，特に短い文となる発話の特徴に着目した素性を利用することで，高い精度でタスク判定可能なモデルを生成する．

2. 3008 人の被験者から収集した 13861 発話を対象とした大規模な実験を行った結果，実験環境において，92%を超える精度でタスク判定可能であったことを示す．

3. 提案方式を適用した対話システムを実際にサービスとして提供し，実システム上で評価を行った結果，97%と実験環境を超える精度でタスク判定可能であったことを示す．また，実サービスを展開することで得られた知見，及び課題について示す．

日本語を対象として，自然な発話からその意図を理解することを目的とした研究がいくつか報告されている [33] [45] [72]．[33] では，あらかじめ収集した発話文例にタスクを付与しておき，新たに入力された発話が，どの発話文例に近いかで，当該発話のタスクを判定する．一方で [72] では，収集した発話文例を用いて，本研究と同様に機械学習によるタスク判定モデルを構築し，入力された発話のタスクを判定する．これらはいずれも，本研究で述べるタスク判定を実現するための技術を提案しているが，実システムの構築や評価には至っていない．一方で，本研究では，既存技術の機械学習によるアプローチを踏襲しつつ，大規模な実験データを利用して発話に最適な特徴量の組合せを実験的に探索する．また，[45] では，

カーナビ上でのシステム実現を前提として，本研究と同様に機械学習のアプローチによるタスク判定モデルを構築し，音声コマンド方式に対する優位性を実験的に示している．これに対して，本研究では機械学習によるタスク判定モデルの構築を前提として，いかにして高精度なタスク判定モデルを構築するかについて述べるとともに，実験環境だけではなく，実際の対話システムにおける評価や知見について示す．

以下では，4.2 節でタスク判定を行うタスク判定機能を含むシステム構成について述べ，4.3 節ではタスク判定の実現方式について述べる．また，4.4 節では，提案方式を実験環境において定量的に評価し，4.5 節では，実際の対話システムにおける評価と知見について示す．最後に，4.6 節でまとめと課題を述べる．

4.2　システム構成

本節では，[22] で提案されている対話システムを例として，タスク判定機能を備えることで実現される対話システムについて述べ，タスク判定に求められる要件についてまとめる．対話システムの構成を図 4.1 に示す．図に示すとおり，対話システムは大きく，ユーザインタフェース，音声認識機能，タスク判定機能，クエリ抽出機能，及び対話システムが呼び出す各タスクによって構成される．ユーザインタフェースを介して入力されたユーザの発話は，(1) 音声認識機能によって音声認識され，(2) タスク判定機能によってどのタスクを実行すべきか決定され，(3) クエリ抽出機能によって，タスク判定に必要な情報を発話文から抽出し，(4) タスク判定結果に応じて適切なタスクを呼び出し実行する．例えば,「横浜周辺のマクドを探して」といった発話が入力された場合には，まず音声認識が行われ，次にタスク判定がおこなわれる．この場合は，施設検索のタスクを呼び出すのが適切であると考えられる．さらに，クエリ抽出により,「横浜」「マクド」といったタスクの実行に必要となる情報が，発話文から抽出され，最後に施設検索タスクが実行される．

以下では，タスク判定について述べる．本論文において対話システムが扱うタ

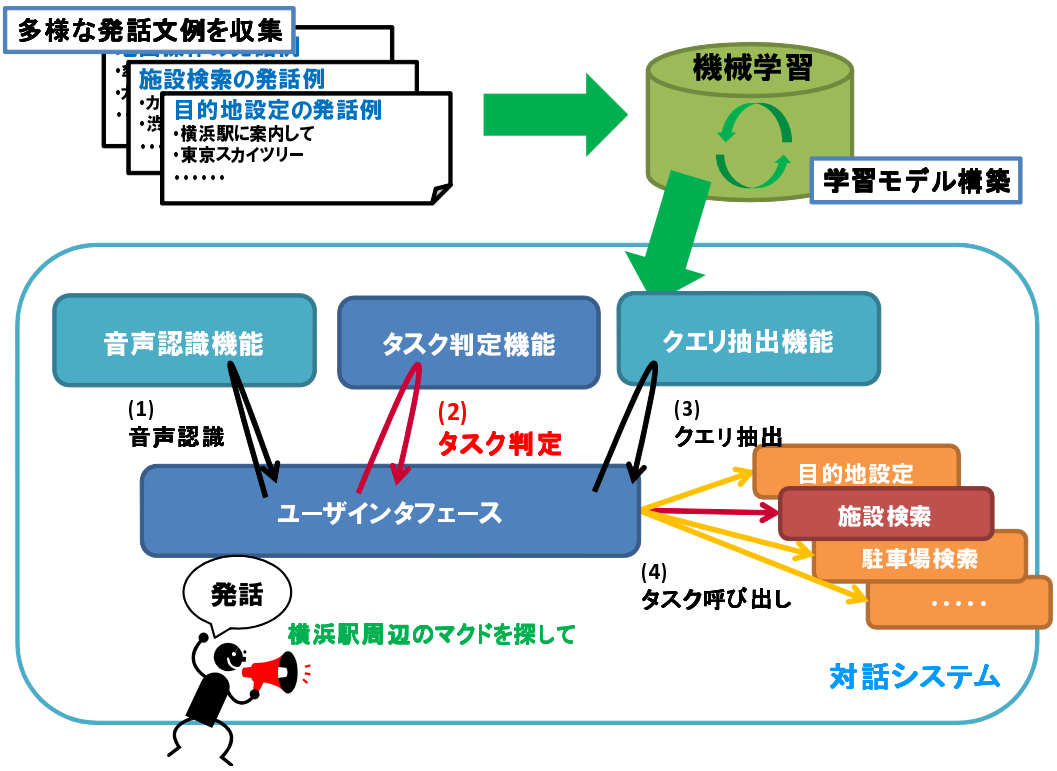

図 4.1: タスク判定機能を備える対話システムの構成

スクの一例を，表 4.1, 4.2 に示す[2]．表に示した発話例の通り，対話システムはタスク判定機能を備えることで，ユーザの自然な発話から適切なタスクを実行可能である．タスク判定は，図 4.1 に示す通り，タスク毎に当該タスクを実行すべき発話文例を収集し機械学習を行うことで判定モデルを構築し，このモデルを利用することで，入力された発話文をいずれかのタスクに分類する．タスク判定の精度は，ユーザの発話からユーザが意図した正解のタスクを呼び出す精度により決定する．タスク判定については 4.3 節で議論し，その精度を 4.4 節，及び 4.5 節において示す．なお，本章で議論するタスク判定の精度は，与えられたタスクの集合の中から，正解のタスクを判定する精度であるが，実サービス環境においては，システムが想定していないタスクの呼び出しを目的とした発話も存在すると考えられ，ユーザの意図理解という文脈では無視することはできない．この問題につい

[2]対話型のカーナビゲーションシステムを想定している．詳細は，C 節を参照されたい．

表 4.1: 本論文で扱う対話システムのタスク一覧 1

タスク名	説明	主な発話例
目的地設定	地名目的地を設定し，案内を開始するタスク	東京に行きたい
施設検索	周辺の施設を地図上に表示するタスク	近くのローソン探して
駐車場検索	周辺の駐車場を地図上に表示するタスク	車停めたい
GS 検索	周辺のガソリンスタンドを地図上に表示するタスク	給油したい
レストラン検索	周辺のレストラン地図上に表示するタスク	お腹が減った
喫茶店検索	周辺の喫茶店を地図上に表示するタスク	喉が渇いた
渋滞情報	渋滞情報を表示するタスク	渋滞情報教えて
自宅設定	自宅を目的地に設定し，案内を開始するタスク	うちに帰る
案内中止	ルート案内を中止するタスク	案内やめて
地図拡大	地図を拡大するタスク	地図表示を大きく
地図縮小	地図を縮小するタスク	地図を広域に

ては，4.5 節において，実際のサービスログを元に議論する．

次にクエリ抽出について述べる．対話システムが実際にタスクを実行するには，発話文からタスクの実行に必要となる情報の抽出が必要となる．この情報を以下ではクエリと呼ぶ．例えば，施設検索タスクである「横浜駅周辺のマクドを探して」という発話からは,「横浜」「マクド」をクエリとして抽出する．また，目的地設定タスクである「東京スカイツリーに行きたい」という発話文から「東京スカイツリー」をクエリとして抽出する．クエリ抽出機能は，システムを実現する上で必須であるが，本論文の主題である高精度なタスク判定手法の確立からは外れるため，付録 B において議論する．

以上，タスク判定とクエリ抽出を行うことで，対話システムはユーザの意図を

表 4.2: 本論文で扱う対話システムのタスク一覧 2

タスク名	説明	主な発話例
地図スケール指定	地図の縮尺を直接指定するタスク	地図の縮尺を 10 メートルに設定
音量上げ	音量を上げるタスク	聞こえない
音量下げ	音量を下げるタスク	うるさい
ミュート	音量を消音するタスク	音声消して
ノースアップ	地図を北向きに固定するタスク	北を上にして
ヘディングアップ	地図を進行方向に固定するタスク	進行方向を上にして
スカイビュー	地図を俯瞰視点で固定するタスク	上から見下ろしたい
休憩	付近の休憩可能な施設を検索するタスク	疲れた
マニュアル	マニュアルを呼び出すタスク	ナビの使い方教えて
ハイウェイモード	地図表示をハイウェイモードにするタスク	高速表示にして
ノーマルビュー	地図表示をノーマルビューにするタスク	一般道表示にして

理解した適切なタスクを実行可能となる．

4.3 タスク判定

本研究におけるタスク判定は，機械学習に基づく文書分類問題に帰着する．具体的には，あらかじめ，タスク毎に収集した発話文例をコーパスとして，機械学習のモデルを生成する．そして，このモデルに基づき，入力された発話文を，いずれかのタスクに分類する．ここで，機械学習における文書分類の精度は，文書か

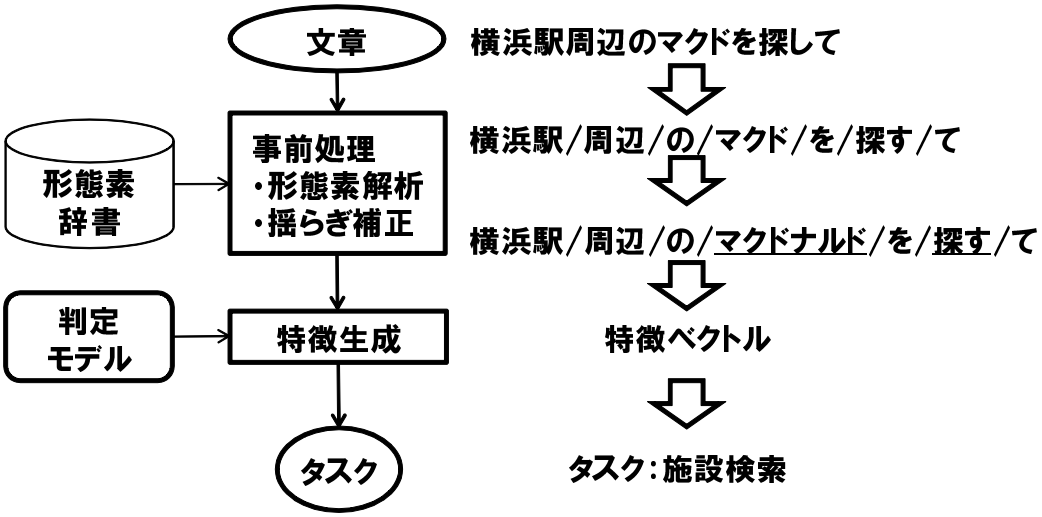

図 4.2: タスク判定の処理概要

らいかに分類に有効な特徴を生成するかに大きく依存する．以下では，タスク判定の処理概要について述べた後，本研究で提案する特徴生成手法について述べる．

タスク判定の処理概要を図 4.2 に示す．タスク判定では，入力された発話文に対して，事前処理を行った後，特徴生成を行い，あらかじめ用意された判定モデルに基づき当該発話文が意図するタスクを判定する．

事前処理では，発話として認識された発話文を形態素に分解し，さらに形態素の表現の揺らぎを統一する．特徴生成では，発話文から生成した素性を組み合わせることで特徴ベクトルを生成する．そして，SVM を用いて事前に学習した判定モデルにより，当該特徴ベクトルに対応するタスクを判定する．機械学習の文書分類の精度は，文書からどのような特徴ベクトルを生成するかに依存する．本研究では，様々な素性を組み合わせて実験的に探索することで，対話システムに入力される発話文を高精度に分類可能な特徴ベクトルを導出し，これを元に判定モデルを生成した．

以下では，事前処理，及び特徴生成について述べる．なお，実際にどの特徴ベクトルで生成した判定モデルが最も良い精度でタスク判定可能であったかについては，4.4 節で述べる．

表 4.3: 対話システムが利用する主な辞書

単語カテゴリ	辞書のサイズ	登録データ例
地名辞書	138,300	横浜，横浜駅，東京スカイツリー，富士山
施設名辞書	21,988	コンビニ，マクドナルド，駅
住所辞書	729,644	東京都，横須賀市光の丘
路線辞書	973	山手線，京急本線
飲食類辞書	973	山手線，京急本線
路線辞書	1730	ハンバーグ，コーラ
揺らぎ辞書	40,891	(マクド, マクドナルド)，(吉牛, 吉野家)

4.3.1 事前処理

図 4.2 に示す通り，事前処理では，まず発話文を形態素に分割する[3]．この際，対話システムへの発話に頻出する単語を形態素単位で抽出可能とするために，形態素解析器の辞書に，独自の辞書を追加する．本論文において対話システムが扱う辞書の一例を表 4.3 に示す．ここで，揺らぎ辞書は，特に地名や施設名において，様々な揺らぎ表記が存在する単語と，その単語の原型のペアが登録されたもので，形態素解析の辞書には揺らぎ表記と原型の両方を登録する．なお，単語カテゴリについては，4.3.2 項にて述べる．

事前処理ではさらに，形態素の揺らぎを補正する．揺らぎ補正を行うことにより，同一の意味を持つ形態素を同一の素性として抽出可能となる．揺らぎ補正では，辞書を利用した補正と，活用形の原型変換が行われる．辞書を利用した補正は，揺らぎ辞書を利用し，揺らぎ表記を原型へと変換する．例えば図 4.2 の例では，「マクド」が「マクドナルド」に変換されている．活用形の原型変換は，活用形を持つ動詞や形容詞が原型に変換される．例えば図 4.2 の例では,「行き」が「行く」に変換されている．

以上の形態素解析処理により，入力された発話文から原型の形態素が抽出され，辞書に登録されている単語はすべて，揺らぎが補正された独立の形態素として抽

[3]形態素解析器には NTT の JTAG [32] を利用している．

出されることとなる．

4.3.2　タスク判定

提案するタスク判定処理は，各発話文を文書と見なしたマルチクラスの文書分類である．機械学習を利用した文書分類の学習，及び判定を行う際，文書を機械学習が扱う特徴ベクトルに変換する必要がある．特徴ベクトルを構成する代表的な素性は，文書を形態素などの単語に分割した uni-gram [71] である．これに対して，様々な素性を追加することで，さらに分類精度を向上することが可能である．しかし，どのような素性の組合せが良い精度を示すかは，各分類問題に依存する [34].

そこで本研究では，様々な素性を組み合わせて生成した特徴ベクトル毎に判定精度を評価し，実験的に発話文のタスク判定に最適な特徴ベクトルの探索を行った．以下では，素性選択を含めた特徴ベクトルの生成について述べる．

N-gram

文書分類問題において多く使われる素性として，N-gram [9] がある．N-gram は，隣り合う N 個の形態素を組み合わせたものである．すなわち，2-gram（bi-gram）では uni-gram に加えて，隣り合う二つの形態素を連結し，3-gram（tri-gram）では uni-gram，bi-gram に加えて，隣り合う三つの形態素を連結する．例えば，図 4.3（N-gram）において，素性として bi-gram を選択した場合，特徴ベクトルの要素と値は，それぞれ表 4.4 に示す通りとなる．ここで，bi-gram により生成された要素は，‘ - ’により，隣り合う形態素を連結して表現している．また，文頭，文末を表す S と E をそれぞれ独立した形態素として元の文に追加することで，先頭，末尾の形態素に隣り合う仮想的な形態素を生成している．N が 2 以上の場合，N=1（uni-gram）では考慮できなかった単語の登場順序を考慮することで，分類精度が向上する場合がある．

しかし，N-gram は，N が大きくなるほど学習コーパス中に極めて低頻度の素性が発生する．その結果，得られた学習モデルにおいて，これら低頻度素性の出現

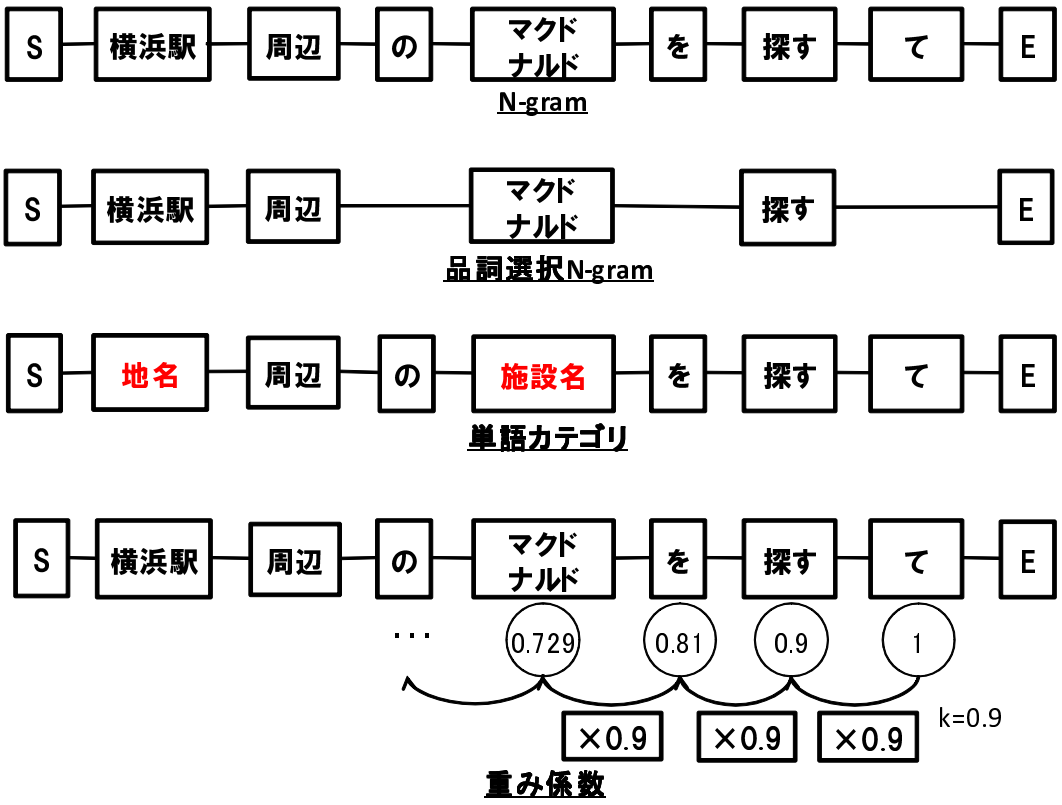

図 4.3: 素性の生成

確率がほぼ 0 となってしまう，所謂ゼロ頻度問題が発生し，これがモデルの精度低下を招く [41].

特にユーザの自然な発話を入力とする本システムでは，各ユーザがそれぞれ多様な表現で発話することにより，上述した問題の発生が顕著となることが考えられる．これに対して，[1] では，特定の品詞の N-gram を要素として利用することで，無駄な次元数の増加を抑えつつ重要な単語の前後関係を抽出する手法を提案している．本研究では，これらの重要な単語がユーザの意図を特に強く反映すると仮定し，タスク判定に有効な特徴であると考えた．

例えば，動詞と名詞を重要な品詞と見なした場合，図 4.3（品詞選択 N-gram）で示した形態素の N-gram を要素とする素性が抽出される．この場合，表 4.4 で示した bi-gram の要素のうち，‘ S-横浜駅 ’,‘ 横浜駅-周辺 ’,‘ 周辺-マクドナルド ’,‘ マ

表 4.4: bi-gram で抽出される特徴ベクトル

要素	値
S-横浜駅	1
横浜駅-周辺	1
周辺-の	1
の-マクドナルド	1
マクドナルド-を	1
を-探す	1
探す-て	1
て-E	1

クドナルド-探す ’,‘ 探す-E ’が特徴ベクトルの要素となる．本研究では，単なる N-gram だけではなく，様々な組合せの品詞[4]を対象とした N-gram から，最適な素性を探索した．

単語カテゴリへの抽象化

一方で，単語を意味的に抽象化した素性（以下，単語カテゴリと呼ぶ）とすることで，精度が向上することが知られている [26] [70]．辞書を利用した単語の抽象化による精度向上効果は 2 章でも示した通りである．特に，ユーザの多様な発話により，発話文には様々な単語や表現が含まれるため，これらをいかに抽象化して単一の単語カテゴリとして扱うかが重要となる．例えば,「横浜駅周辺のマクドを探して」の「横浜駅」が「東京スカイツリー」であったとしても，タスク判定の観点からは一つの素性として扱うべきである．

提案手法が単語の抽象化に利用する辞書の一例を表 4.3 に示す．表に示す通り，辞書には単語カテゴリと各単語カテゴリに抽象化すべき単語が登録される．この場合，図 4.3（単語カテゴリへの抽象化）に示す通り,「横浜駅」「マクドナルド」

[4]JTAG が付与する品詞は，益岡，窪田文法 [52] を元とする．

（以下，単語カテゴリに対して元の形態素の表記を表層と呼ぶ）をそれぞれ，地名，施設名といった単語カテゴリに置換する．

重み係数

特徴ベクトルを生成するためには，素性の選択に加えて，各素性の重み（特徴ベクトルの値）を決定する必要がある．一般的には，素性の位置は考慮せずに，素性が文書に登場した回数を値とする [71]．しかし，対話システムへの実際の発話例[5]を分析したところ，文の後ろに登場する形態素が，タスク判定により重要な意味を持つ場合が多く発見された．例えば,「お腹が空いた．レストラン探して.」「うるさいなあ．音下げてよ」という二つの文例を考える．前者は，付近のレストランを検索する施設検索タスクであるが，前半の「お腹が空いた」の部分だけでは，自宅設定といったタスクであることも考えられるため，ユーザの意図が曖昧であり，後半の「レストラン探して」により重みを置いた学習を行った方が，タスク判定の精度が良いことが推測される．後者は，音量下げのタスクであるが，前半部分だけではミュートや案内中止といったタスクも考えられ，同様にユーザの意図が曖昧である．

そこで本研究では，素性の登場回数による重みだけではなく，各素性の文中での位置に応じた重みを付与して生成した特徴ベクトルについても，評価を行った．具体的には，図 4.3（重み係数）で示した通り，重み係数 $k(0 < k \leq 1)$ を定義し，i 番目の形態素（$1 \leq i \leq n$,n は発話文の形態素数）には $w_i = k^{n-i}$ の重みを付与する（$k = 1$ であれば，登場回数を重みとした場合と同じ）．

以上，品詞選択 N-gram，単語カテゴリ，重み係数といった様々な特徴量を組み合わせた特徴ベクトルの生成を行い，どの組合せで生成したモデルが最も精度良くタスク判定可能か，実験的に探索を行った．また，得られたモデルの，実験環境における有効性の検証を行った．実験の具体的な設定と実験結果については，4.4 節で示す．

[5]収集文については 4.4 節で述べる

4.4 実験環境における性能評価

4.4.1 実験条件

提案した特徴量に基づくタスク判定の有効性を，各特徴ベクトルを元に生成した判定モデルによるタスク判定の精度により評価する．具体的には，対話システムの発話例として収集した学習コーパスにあらかじめ正解のタスクを付与し，5分割交差検定により精度を測定した．分類するタスクは，表4.1，4.2で示した通りである．

学習コーパスは，被験者3008人に対するWebベースのアンケートにより収集を行った．収集に利用したWebページの例を図4.4に，被験者の属性分布を表4.5に示す．アンケートでは，タスク毎に被験者が置かれる状況を設定し，その上で被験者がカーナビの各タスクに対してどのような発話をするか問い合わせ，発話の記述を促した．例えば，目的地設定であれば，図で示したような被験者の状況を示し，さらに被験者への問いかけを示すことで，被験者に対話システムへの発話を記述させた．このような状況は，目的地設定タスクに対して8，施設検索に対して4，残りは各タスク1つずつ，合計32用意した．さらに，収集した発話文から重複を全て取り除き，作業者3名によってクレンジングを行った結果，合計13861文の学習コーパスを得た．タスク毎の発話数は，表4.6の発話数で示した通りである．また，得られた学習コーパスは，4.3.1項で述べた事前処理の方法に従い，形態素解析，及び揺らぎ補正を行った．事前処理を行った後の学習コーパスについて，平均形態素数は5.99であり，5.1%の形態素が地名辞書に，1.9%が施設名辞書に，2.3%が住所辞書に，0.1%が路線辞書に登録された形態素であった．

評価対象とする特徴量は以下の通り選択した．N-gramは，各Nの値について，それぞれ品詞選択有り，無しの場合を評価した．ここで，品詞は，各Nにおいて最適であった組合せを採用した．これらには，たとえば名詞，動詞，形容詞，各活用系の接尾辞などが含まれる．以下の評価結果では最も精度が良かったN=2（bi-gram）の場合について示す．

単語カテゴリについては，カーナビに対する発話文に頻出する10の単語カテゴリを用意した．抽象化に利用した辞書の一部は，表4.3で示した通りである．以下

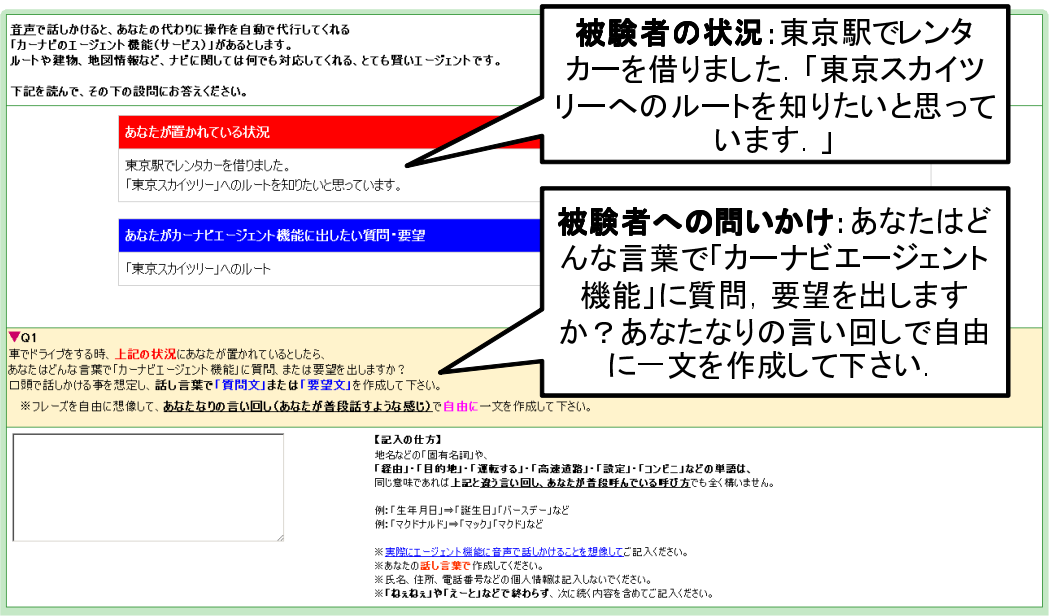

図 4.4: コーパス収集の Web ページ

の評価結果では，これらの単語カテゴリを素性とした場合，及び形態素の表層のみを素性とした場合について示す．

重み係数については，$k = 0$ から $k = 1.0$ まで 0.1 きざみで評価した．以下の評価結果では特に，$k = 1.0$，$k = 0.9$，$k = 0.8$ の 3 通りについて示す．

生成した特徴ベクトルは，F 値 [11] を利用し，特徴ベクトル毎に最適な特徴次元数に調整した．また，学習器の実装には，線形 SVM による多クラス分類が可能な liblinear [17] を利用した．ここで，liblinear は，過学習を抑える正則化項 [31] の与え方として，L2 正則化，L1 正則化のいずれかを選択可能である．また，学習データ中の誤分類を許容するソフトマージン SVM であり，誤分類データに対するペナルティを調整可能である．これら SVM のパラメータについては，特徴ベクトル毎に最適値に調整した．

以下では，学習コーパスの発話文を利用したタスク判定の精度について評価結果を示す．なお，実際の対話システム考えた場合には，音声を認識する際の音声認識誤りや，用意した 22 タスク以外の発話がなされることによる誤りも想定しなければならないが，本実験では考慮していない．これらについては，4.5 節におい

表 4.5: 被験者の属性分布

属性（性-年代）	人数）	割合 [%]
男性-10 代	20	0.7
男性-20 代	103	3.4
男性-30 代	337	11.2
男性-40 代	536	17.8
男性-50 代	422	14.0
男性-60 代以上	305	10.1
女性-10 代	34	1.1
女性-20 代	248	8.2
女性-30 代	416	13.8
女性-40 代	327	10.9
女性-50 代	175	5.8
女性-60 代以上	85	2.8
全体	3008	100.0

て，サービスログを用いた評価結果，及び考察を示す．

4.4.2　実験結果

評価対象とした 12 通りの特徴ベクトルに基づくタスク判定の精度を表 4.6 に示す．表は，各条件の組合せで生成された特徴ベクトルと，それに対する精度を示している．精度は，最良の特徴次元数，及び SVM の設定における 5 分割交差検定の平均値である．選択した特徴ベクトルによっては，92%を超える精度でタスクを判定可能であり，提案方式は，実験環境において与えられた 22 タスクに対して，十分な精度でタスク判定可能であることを確認できた．

各特徴量を個別に見た場合，単語カテゴリへの抽象化の効果が最も大きく，モデルの精度向上のために，辞書による抽象化が効果的であることが分かる．またその他の二つの特徴についても，それぞれ効果があることを確認できた．

表 4.6: 各特徴ベクトルにおけるタスク判定精度

bi-gram	単語カテゴリ/表層	重み係数	正解発話数	精度
品詞選択有り	単語カテゴリ	0.9	12807	0.924
品詞選択有り	単語カテゴリ	0.8	12802	0.924
品詞選択有り	単語カテゴリ	1.0	12801	0.924
品詞選択無し	単語カテゴリ	0.9	12731	0.918
品詞選択無し	単語カテゴリ	0.8	12709	0.917
品詞選択無し	単語カテゴリ	1.0	12657	0.913
品詞選択有り	表層	0.9	11800	0.851
品詞選択有り	表層	0.8	11788	0.850
品詞選択有り	表層	1.0	11777	0.850
品詞選択無し	表層	0.9	11605	0.837
品詞選択無し	表層	0.8	11575	0.835
品詞選択無し	表層	1.0	11512	0.831

次に，最も精度の良かった条件において，タスク毎のタスク判定精度を評価した．結果を表 4.7 に示す．表より,「音量上げ」,「音量下げ」,「地図拡大」,「地図縮小」の 4 タスクが他のタスクと比較して，精度が極端に悪いことが分かる．これらの誤りについて詳細を分析した結果，これら 4 つのタスク間でタスク判定を誤っていることがわかった．

例えば，音量上げと地図拡大の間で見られた誤りは，どちらのコーパスにも「大きく/して」や「小さ/すぎ」といったどちらのタスクともとれる曖昧な文が含まれていたことが原因となっていた．これは，曖昧な文が入力された場合には，どちらのタスクを正解とするか定め，不正解タスクに含まれる曖昧な文を修正することで解決可能である．一方で,「地図拡大」と「地図縮小」の間で見られた誤りは,「もう少し/大きな/縮尺/で/地図/を/見/たい」といった，対話システムに対する発話としては比較的長めの文において見られた．この例は,「大きな」という単語から分かる通り，地図拡大タスクが正解であるが,「大きな」以外の単語から生成され

表 4.7: タスク毎の評価結果

タスク	発話数	正解発話数	精度
目的地設定	4414	4199	0.951
施設検索	1050	991	0.944
駐車場検索	1289	1203	0.933
GS 検索	277	261	0.942
レストラン検索	638	600	0.940
喫茶店検索	558	526	0.943
渋滞情報	340	325	0.956
自宅設定	1050	991	0.944
案内中止	1137	1035	0.910
地図拡大	313	238	0.760
地図縮小	375	304	0.811
地図スケール指定	324	312	0.963
音量上げ	340	278	0.818
音量下げ	284	223	0.785
ミュート	149	139	0.933
ノースアップ	307	280	0.912
ヘディングアップ	183	169	0.923
スカイビュー	138	122	0.884
休憩	180	165	0.917
マニュアル	225	190	0.844
ハイウェイモード	100	85	0.850
ノーマルビュー	190	171	0.900
合計	13861	12807	0.924

る素性の影響により，僅差ながら地図縮小が地図拡大を上回ってしまった．これは，学習コーパスにおける低頻度の素性の学習が不十分であることが原因である．

そこで，学習コーパスを追加することにより，これらのタスクの精度向上を行った．具体的にはこれら 4 タスクについて判定を誤った発話文を抽出し，作業者 3 名により，これらの発話文に類似した文を，各タスクにつき 20 から 30 程度追加した．追加を行った後のタスク判定の精度を表 4.8 の精度 2 に示す．表に示した通り，これら 4 つのタスクについて，著しい精度改善がなされたことを確認できた．

4.5 対話システムにおける性能評価

本節では，提案方式を適用した対話システムを実際に開発し，実サービスとして提供することで，実際の対話システムにおける提案方式の有効性，及び得られた知見を示す．なお，本サービスの概要については，付録 C において述べている．

サービスログは，サービス開始後の 2012 年 12 月 9 日から 23 日までの 2 週間の間に収集したものである．収集されるログには，発話日時，音声認識後の発話文，タスク判定結果，及びクエリ抽出結果が記録される．

収集したログから 8322 文を抽出し，作業者 3 名により表 4.9 で示すタイプ 1 からタイプ 6 の 6 種類に分類した．タイプ 1 はタスク判定，クエリ抽出のいずれも正解した場合であり，タイプ 2 は，タスク判定のみ不正解であった場合である．そのため，タイプ 1 のログの数を n_1，タイプ 2 のログの数を n_2 とした場合，$\frac{n_1}{n_1+n_2}$ は，4.4 節で示したタスク判定精度の評価条件と等しく，97%を越える精度となった．

各タスクの精度について，表 4.10 に示す．表より，音量上げ/下げ，地図拡大/縮小は，いずれも 88%から 95%と高い精度を達成しており，4.4.2 項において行った学習コーパスの追加が，実システムにおいても効果があったことが分かった．また，実験環境における評価と比較して精度が良くなっているが，これは，サービス環境における発話が，比較的タスク判定の容易な目的地設定や施設検索に集中したためと考えられる．以上で述べた通り，他の誤りを考慮しないという条件の下では，ユーザの発話に対して十分な精度で正しいタスク判定可能であることが分かった．

ただし，発話からのユーザの意図理解という文脈では，その他の誤りについても修正を行っていく必要がある．これらのうち，特に，タイプ 3，タイプ 4 として

表 4.8: タスク毎の評価結果（コーパス追加後）

タスク	発話数	正解発話数	精度
目的地設定	4414	4199	0.951
施設検索	1050	991	0.944
駐車場検索	1289	1203	0.933
GS 検索	277	261	0.942
レストラン検索	638	600	0.940
喫茶店検索	558	526	0.943
渋滞情報	340	325	0.956
自宅設定	1050	991	0.944
案内中止	1137	1031	0.907
地図拡大	338	311	0.920
地図縮小	404	378	0.936
地図スケール指定	324	312	0.963
音量上げ	365	336	0.921
音量下げ	306	277	0.905
ミュート	149	139	0.933
ノースアップ	307	280	0.912
ヘディングアップ	183	169	0.923
スカイビュー	138	122	0.884
休憩	180	165	0.917
マニュアル	225	190	0.844
ハイウェイモード	100	85	0.850
ノーマルビュー	190	171	0.900
合計	13962	13062	0.936

記録されたクエリ抽出の誤りに対応するため，今後，収集したログを元に，辞書

表 4.9: サービスログの分類

タイプ	説明	数
タイプ 1	タスク判定正解，クエリ抽出正解	5249
タイプ 2	タスク判定誤り，クエリ抽出正解	160
タイプ 3	タスク判定正解，クエリ抽出誤り	357
タイプ 4	タスク判定誤り，クエリ抽出誤り	94
タイプ 5	未実装タスクによるタスク判定不可	907
タイプ 6	音声認識誤りによるタスク判定不可	1555

の増強やクエリ抽出で用いている CRF モデルの改良を行っていく[6].

タイプ 5 のログは，人が見てユーザの意図を理解できるものの，システムが提供するいずれのタスクにも当てはめることが出来ない発話である．実際に分類を行ったところ，図 4.5 のような分布となった．最も多かったのは,「いつもありがとう」「役に立たないな」などの雑談で，未対応タスクのうちの 3/4 を占めた．また，付録 C でも述べる通り，クライアントアプリがスマートフォン上で動作することもあり，電話やメール機能を利用したいと考えるユーザが多く見られた．さらに，カーナビの機能の範疇を超えた天気やニュースなどの情報配信を希望するユーザがいることも分かった．これらは，本システムがユーザの自由な発話を許容することによって得られた新しい知見であり，実際にシステムが意図を理解して対応すべきタスクであると言える．今後は，ユーザの意図に応じて，タスクを拡大可能なシステムの実現を検討していく．

最後に，タイプ 6 のログは，音声認識誤りにより，タスク判定や地名抽出が不可能である場合に記録される．自動車内での音声操作では，ロードノイズなどによる音声認識誤りは多く発生する．実際，タイプ 6 のログの量は，タイプ 2 から 5 と比較して，かなり多くなっている．提案した機械学習ベースのタスク判定は，タイプ 6 のようないずれのタスクにも分類すべきではない文についても，用意したいずれかのタスクに分類をしてしまう．タイプ 6 のログでは，例えば案内中止の

[6]CRF を利用したクエリ抽出の詳細については付録 B を参照.

表 4.10: 実システムにおけるタスク毎の評価結果

タスク	発話数	正解発話数	精度
目的地設定	2523	2462	0.976
施設検索	969	936	0.966
駐車場検索	145	140	0.966
GS 検索	74	72	0.973
レストラン検索	508	493	0.970
喫茶店検索	257	250	0.973
渋滞情報	67	66	0.985
自宅設定	348	338	0.971
案内中止	173	166	0.960
地図拡大	64	61	0.953
地図縮小	43	38	0.884
地図スケール指定	6	6	1.000
音量上げ	70	67	0.957
音量下げ	44	40	0.909
ミュート	4	4	1.000
ノースアップ	12	12	1.000
ヘディングアップ	7	7	1.000
スカイビュー	8	7	0.875
休憩	56	54	0.964
マニュアル	24	23	0.958
ハイウェイモード	2	2	1.000
ノーマルビュー	5	5	1.000
合計	5409	5249	0.970

タスクに分類されてしまう場合があり，このような場合，目的地設定が取り消されてしまうことで，ユーザの意図とは異なる動作をしてしまっている．ここで例

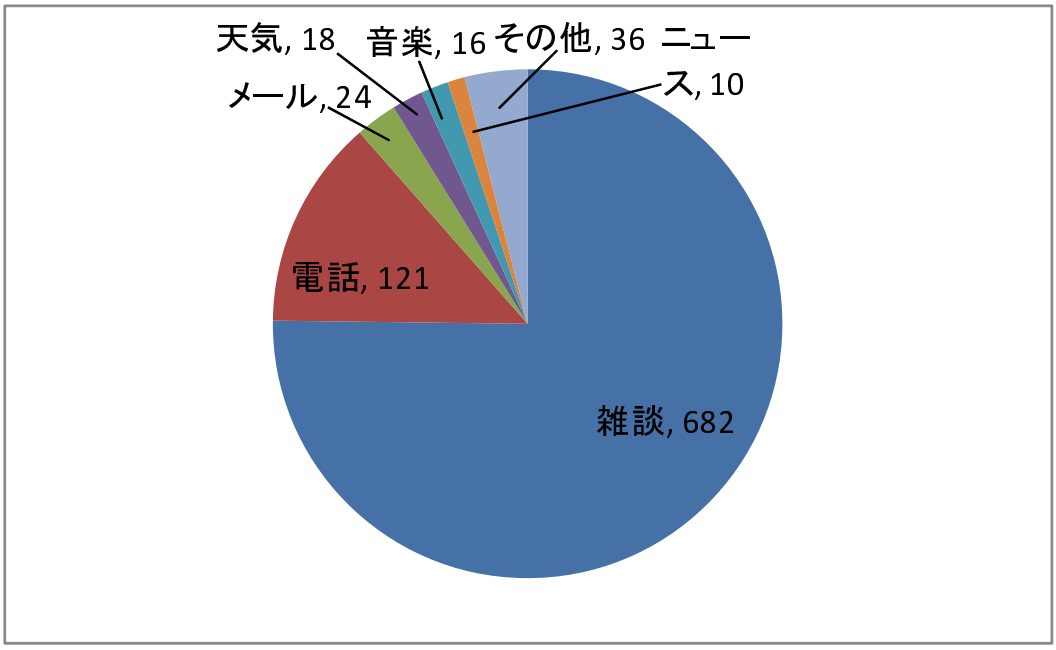

図 4.5: 未対応タスクの分布

えば，[25] では発話の音響的特徴，音声認識後の言語的特徴の両方を用い，SVM による棄却判定を行っている．今後，このような手法を適用することにより，タスク判定結果の棄却機能を実現していく．

4.6　むすび

本章では，対話システムに対する発話を単一文書と見なし，これを高精度に理解するタスク判定手法について提案を行った．提案方式は，発話文を，あらかじめ用意された適切なタスクに分類する．高精度なタスク判定モデルを構築するために，発話文の特徴を考慮した素性や重みを提案した．また，これらの最適な組合せを実験的に探索するため，被験者 3008 人から収集した 13861 文を対象とした大規模な実験を行った．その結果，提案した全ての特徴量が有効であることを確認し，最適な組合せでは，92%を超える精度でタスク判定可能であることを示した．また，実際のサービスログを分析し，実システム環境では 97%を超える精度

でタスクを判定可能であることを示した．

本章では，対話システムを想定して，発話文を高精度に理解可能な手法を提案した．しかし，特にシステムに依存した情報は利用せず，発話文を一般的な単一文書と同様に扱った．そのため，あらかじめ学習コーパスさえ用意されれば，発話文に限らずあらゆる単一文書に対して有効な手法であると言える．ただし，提案した特徴量の組み合わせが最適であるかについては，文書データに依存するため，本章で示した通り，最適な特徴量の組み合わせを実験的に探索する必要がある．

今後は，サービスログの分析から明らかになった課題を解決し，より高精度にユーザの発話意図を理解可能な手法について検討していく．

第5章　結論

5.1　本論文のまとめ

本論文では，ユーザの意図が反映された文書を大きく4つに分類し，そのうち3つの文書データについて知識獲得技術を確立し，さらに提案技術の実アプリケーションにおける有効性を示した．

まず第1章では，文書データを介したユーザの意図理解の重要性について述べた．また，文書データを時系列と文書の粒度の二つの軸で分類し，各文書データの特徴と意図理解に必要となる技術，さらに意図理解により実現可能な応用例を示した．

第2章では，分類1（文書集合に対する非時系列分析技術）の技術として，ユーザが蓄積したWeb閲覧ログを対象とし，これにトピックモデルを適用することで，ユーザのWeb閲覧行動を高精度にモデル化する手法を提案した．提案方式は，単語の抽象化がモデルの精度向上に寄与するという自然言語分野の知見に従い，ユーザの一連のWeb閲覧行動を抽象化した．抽象化に際し，提案方式では，URL間の概念的な階層関係を表す階層型URL辞書を構築し，これを利用することでユーザの一連のWeb閲覧ログから概念的に抽象化した単語セットを抽出した．第2章ではさらに，7537人のユーザの4ヶ月間に及ぶプロキシログを対象とした大規模な実験を行い，提案方式により生成したモデルの精度をパープレキシティにより示した．また，Webページの推薦問題を例として，セレンディピティを考慮した重要度付ヒット率を評価指標として導入することで，実アプリケーションにおいても，提案方式が有効であることを示した．さらに，モデルから得られたユーザプロファイルの結果を可視化することで，主観的に提案方式の有効性を示した．

第3章では，分類2（文書集合に対する時系列分析技術）の技術として，Twitter

などユーザが時系列に生成する文書データを対象とし，時系列文書を高精度にモデル化可能な手法を提案した．提案方式は，現在のモデルと過去のモデルを平滑化することで，より時間変化にロバストなモデルを生成した．ここで，提案方式は，平滑化に際しパーティクルフィルタを適用することで，時系列のモデル変化に最適化した平滑化係数を導出した．提案方式はさらに，時系列に登場する新語に応じて，モデルが扱う語彙集合を動的に拡張した．ここで，提案方式は，各文書を出現単語とその頻度の key-value 形式で表現する実装を行うことで，語彙集合の無限の拡張による計算時間増加を抑えた．第 3 章ではさらに，合計 9ヶ月間，1 日 100 万ツイートを対象とした大規模な実験により，提案方式の有効性を示した．まず，提案方式による時系列文書のモデル化精度をパープレキシティにより示した．パープレキシティ評価では，提案した最適平滑化，動的語彙集合のいずれもモデルの精度向上に寄与していることを確認できた．さらに，ホットトピック抽出問題を例として，KL ダイバージェンスを指標とした評価結果を示し，実アプリケーションにおいても，提案方式が有効であることを示した．

第 4 章では，分類 3（単一文書に対する非時系列分析技術）の技術として，ユーザの対話システムに対する発話を対象とし，ユーザの発話意図を高精度に理解する手法を提案した．提案方式は，ユーザの発話から実行すべき適切なタスクを高精度に判定する．精度のよいタスク判定を実現するため，提案方式は特に短い文となる発話の特徴に着目した素性を利用した．第 4 章ではさらに，3008 人の被験者から収集した 13861 発話を対象とした大規模な実験により，提案方式によるタスク判定の精度を評価した．その結果，提案した全ての素性による精度向上の効果を確認し，最も良い組合せでは 92%を超える精度でタスク判定可能であることを示した．さらに，提案方式を適用した対話型カーナビゲーションシステムを開発し，実際にサービスとして提供を行ったところ，実システム環境におけるタスク判定精度は 97%とさらに高い精度を達成したことを示した．

各章で提案した知識獲得技術は，それぞれの章で対象とした特徴を持ついかなる文書に対しても適用可能な汎用的な技術である．そのため，本論文の成果の本質は，ユーザの意図を反映した上記のいずれかに分類される文書が与えられた場合に，文書を高精度にモデル化し，ユーザの意図理解を可能とする知識獲得技術

を確立したことにある．例えば，第 2 章で提案した単語抽象化手法は，単語間の概念辞書を用意することで，いかなる文書集合のモデル化にも適用可能である．また，第 3 章で提案した最適平滑化手法は，いかなる時系列文書のモデル化にも適用可能であり，文書データに新語が存在するならば，動的語彙集合に対応したモデルも，同様に適用可能である．さらに第 4 章で提案したタスク判定手法は，ユーザの自然な言語で記述・発話されたあらゆる単一文書に対して適用可能である．

5.2　今後の研究課題

本論文では，各章で提案したモデルの精度を定量的に評価し，さらに，実アプリケーションにおける有効性を示した．しかし，どのようなアプリケーションにおいても，提案方式が有効であるかについては，検証が不十分である．今後，より多くのアプリケーションへの適用を行っていくことで，提案方式の有効性をさらに検証する．

また，本論文で提案した技術は，第 1 章で分析対象とした 3 つの文書データに対して適用可能である．そのため，今後の課題としては，まず，これら 3 つの文書に属さない分類 4 の文書データに対する知識獲得技術の確立があげられる．これは，第 1 章で述べた分類によると，具体的には時系列に蓄積される単一文書である．単一文書にはトピックモデルを適用することが困難であるため，第 3 章で提案した時系列文書分析技術は適用することが出来ない．また，第 4 章で提案した単一文書の分析技術は，時系列文書の分析を考慮していない．今後，本論文で提案した技術と，[43] で提案されているようなオンラインの教師有り機械学習技術を組み合わせ，このような文書データからも逐次ユーザの意図理解が可能な，知識獲得技術の確立を目指していく．

謝辞

本研究全般に関して，懇切なる御指導と惜しみない御助言を頂きました大阪大学大学院情報科学研究科マルチメディア工学専攻 西尾章治郎教授に謹んで御礼申し上げます.

本研究を推進するにあたり，直接の御指導，御助言，御討論を頂きました大阪大学大学院情報科学研究科マルチメディア工学専攻 原隆浩准教授に衷心より感謝申し上げます.

本論文を執筆するにあたり，大変有益な御指導と御助言を多数賜りました大阪大学大学院情報科学研究科マルチメディア工学専攻 藤原融教授，前川卓也准教授に心より感謝申し上げます.

社会人ドクターとして入学する機会を頂き，また直接の御助言，御協力，御討論を頂きましたNTTドコモ執行役員　研究開発推進部長　栄藤稔博士（元大阪大学サイバーメディアセンター招へい教授）に深く御礼申し上げます.

また，社会人ドクターとしての立場をご理解頂き，日常業務の遂行にあたって多大なる支えを頂くと共に，企業研究者の先輩として多くの御指導と御助言を賜りました，NTTドコモR&D総務部担当部長　片桐雅二博士，NTTドコモサービス＆ソリューション開発部担当課長　吉村健博士（元大阪大学サイバーメディアセンター招へい准教授）に深く御礼申し上げます.

また，講義，研究生活を通じて，学問に取り組む姿勢をご教授頂きました大阪大学大学院情報科学研究科マルチメディア工学専攻 細田耕教授，下條真司教授，薦田憲久教授に深く感謝致します.

また，本研究の一部は，大阪大学サイバーメディアセンターへ2009年10月より2012年3月までの期間に設置されたドコモ（コミュニケーション構造解析）共同研究部門において行ったものです．共同研究部門において，懇切丁寧な御指導と

御助言を賜りました大阪大学サイバーメディアセンター　竹村治雄教授に深く感謝致します．また，多くの議論を通じて多大なご指導を頂きました，日本大学文理学部　尾崎知伸博士（元共同研究部門特任講師），関西大学データマイニング応用研究センター　佐野夏樹博士（元共同研究部門特任助教）に深く感謝致します．また，共同研究部門の兼任教員として多くの御助言，御示唆，御協力を賜りました，サイバーメディアセンター　間下以大助教，基礎工学研究科　土方嘉徳准教授に感謝致します．そして，共同研究部門に参加された，DOCOMO Innovations, Inc. 秋永和計氏，サービス＆ソリューション開発部　池部優佳氏には，貴重な御助言，研究を進める上での多くの協力を頂き，深く感謝致します．

また，本研究を進めるにあたり，多大なる御助言，御協力，御支援を頂きました大阪大学サイバーメディアセンター　義久智樹准教授，大阪大学大学院情報科学研究科 神崎映光助教，白川真澄特任助教に深く感謝致します．

また，著者の学生時代に研究者としての基礎を教えて頂き，技術研究の道に導いて下さった大阪大学大学院工学研究科　春本要准教授，中野賢講師に深く感謝致します．

また，本研究において，ともに研究を進め，多大なる御協力を頂いた NTT ドコモマーケティング部主査　金野晃氏に深く感謝致します．また，本論文の執筆にあたり，遠隔地にいる著者に多大なるサポートを頂きました，大阪大学大学院情報科学研究科マルチメディア工学専攻 佐々木勇和氏，杉谷卓哉氏に深く御礼申し上げます．

また，本研究を進めるにあたり，多くの御討論や御助言を頂きましたサービス＆ソリューション開発部ビッグデータ担当，及び大阪大学大学院情報科学研究科マルチメディア工学専攻西尾研究室の諸氏に深く御礼申し上げます．

最後に，本論文の執筆において，物心両面から支えてくれた妻万里子に心から感謝します．

参考文献

[1] Akiba, T., Itou, K., Fujii, A., and Ishikawa, T.: Selective Back-Off Smoothing for Incorporating Grammatical Constraints into the N-Gram Language Mode, in *Proceedings of International Conference on Spoken Language Processing*, Vol. 2, pp. 881–884 (2002).

[2] AlSumait, L., Barbara, D., and Domeniconi, C.: On-Line LDA: Adaptive Topic Models for Mining Text Streams with Applications to Topic Detection and Tracking, in *Proceedings of IEEE International Conference on Data Mining*, pp. 3–12 (2008).

[3] Aone, C., and Bennett, S. W.: Evaluating Automated and Manual Acquisition of Anaphora Resolution Strategies, in *Proceedings of 33rd Annual Meeting on Association for Computational Linguistics*, pp. 122–129 (1995).

[4] Berger, A., Pietra, S., and Petra, V.: A Maximum Entropy Approach to Natural Language, *Computational Linguistics*, Vol. 22, No. 1, pp. 39–71 (1996).

[5] Bessho, K.: Text Segmentation Using Word Conceptual Vectors, *Transactions of Information Processing Society of Japan*, Vol. 42, No. 11, pp. 2650–2662 (2001).

[6] Blei, D., and Lafferty, J.: Dynamic Topic Models, in *Proceedings of 23rd International Conference on Machine Learning*, pp. 113–120 (2006).

[7] Blei, D., Ng, A., and Jordan, M.: Latent Dirichlet Allocation, *Journal of Machine Learning Research*, Vol. 3, pp. 993–1022 (2003).

[8] Buchner, A. G., and Mulvenna, M. D.: Discovering Internet Marketing Intelligence through Online Analytical Web Usage Mining, *ACM SIGMOD Record*, Vol. 27, No. 4, pp. 54–61 (1998).

[9] Cavnar, W., and Trenkle, J.: N-Gram-Based Text Categorization, in *Proceedings of 3rd Annual Symposium on Document Analysis and Information Retrieval*, pp. 161–175 (1994).

[10] Charniak, E.: A Maximum-Entropy-Inspired Parser, in *Proceedings of 1st North American Chapter of the Association for Computational Linguistics Conference*, pp. 132–139 (2000).

[11] Chen, Y., and Lin, C.: Combining SVMs with Various Feature Selection Strategies, *Feature Extraction*, Vol. 207, pp. 315–324 (2006).

[12] Das, A., Datar, M., Garg, A., and Rajaram, S.: Google News Personalization: Scalable Online Collaborative Filtering, in *Proceedings of 16th International Conference on World Wide Web*, pp. 271–280 (2007).

[13] Davidov, D., Tsur, O., and Pappoport, A.: Enhanced Sentiment Learning Using Twitter Hashtags and Smileys, in *Proceedings of 23rd International Conference on Computational Linguistics*, pp. 241–249 (2010).

[14] Dharanipragada, S., and Roukos, S.: A Fast Vocabulary Independent Algorithm for Spotting Words in Speech, in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 233–236 (1998).

[15] Doyle, G., and Elkan, C.: Accounting for Burstiness in Topic Model, in *Proceedings of 26th International Conference on Machine Learning*, pp. 281–288 (2009).

[16] Elberrichi, Z., Rahmoun, A., and Bentaalah, M.: Using WordNet for Text Categorization, *International Arab Journal of Information Technology*, Vol. 5, No. 1, pp. 3–37 (2008).

[17] Fan, R., Chang, K., Hsieh, C., Wang, X., and Lin, C.: LIBLINEAR: A Library for Large Linear Classification, *Journal of Machine Learning Research*, Vol. 9, pp. 1871–1874 (2008).

[18] Finkel, J. R., and Manning, T. G. C.: Incorporating Non-Local Information into Information Extraction Systems by Gibbs Sampling, in *Proceedings of 43rd Annual Meeting on Association for Computational Linguistics*, pp. 363–370 (2005).

[19] 藤本拓, 栄藤稔, 金野晃, 秋永和計：LDA モデルを利用した Web ユーザプロファイリング方式のコンテンツ推薦への適用に関する研究, 第 20 回コンピュータ・シンポジウム (2011).

[20] 藤本拓, 栄藤稔, 金野晃, 秋永和計：階層型 URL 辞書と LDA による Web 閲覧行動のモデル化, 電子情報通信学会論文誌, Vol. J95-D, No. 2, pp. 183–192 (2012).

[21] 藤本拓, 原隆浩, 西尾章治郎：時系列の最適平滑化と動的な語彙集合を考慮した時系列文書に対するトピック解析手法, 電子情報通信学会論文誌, Vol. J96-D, No. 5, pp. 1212–1221 (2013).

[22] 藤本拓, 原隆浩, 西尾章治郎：自然な発話により操作可能なカーナビゲーションシステムの開発, 電子情報通信学会論文誌, Vol. J96-D, No. 11 (掲載決定).

[23] Fujimoto, H., Etoh, M., Kinno, A., and Akinaga, Y.: Topic Analysis of Web User Behavior Using LDA Model on Proxy Logs, in *Proceedings of 15th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Vol. 1, pp. 525–536 (2011).

[24] Fujimoto, H., Etoh, M., Kinno, A., and Akinaga, Y.: Web User Profiling on Proxy Logs and its Evaluation in Personalization, in *Proceedings of 13th Asia-Pacific Web Conference*, pp. 107–118 (2011).

[25] 藤田洋子, 竹内翔大, 川波弘道, 松井知子, 猿渡洋, 鹿野清宏：単語の頻度と音響の特徴を利用した SVM による無効入力の棄却, 情報処理学会研究報告音声言語情報処理 (SLP), Vol. 2010-SLP-80, No. 3, pp. 1–6 (2010).

[26] 福本文代, 鈴木良弥：WordNet の同義語クラスとその上位関係を利用した文書の自動分類, 情報処理学会論文誌, Vol. 43, No. 6, pp. 1852–1865 (2002).

[27] Griffiths, T., and Steyvers, M.: Finding Scientific Topics, in *Proceedings of National Academy of Sciences of the United States of America*, Vol. 101, pp. 5228–5335 (2004).

[28] Hoffman, M., and Blei, D.: Online Learning for Latent Dirichlet Allocation, in *Proceedings of Advances in Neural Information Processing Systems*, pp. 856–864 (2010).

[29] Hofmann, T.: Probabilistic Latent Semantic Analysis, in *Proceedings of 22nd Annual ACM Conference on Research and Development in Information Retrieval*, pp. 50–57 (1999).

[30] Hong, L., Yin, D., Guo, J., and Davison, B. D.: Tracking Trends:Incorporating Term Volume into Temporal Topic Models, in *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 484–492 (2011).

[31] Hsieh, C.-J., Chang, K.-W., Lin, C.-J., Keerthi, S. S., and Sundararajan, S.: A Dual Coordinate Descent Method for Large-Scale Linear SVM, in *Proceedings of 25th International Conference on Machine learning*, pp. 408–415 (2008).

[32] 今村賢治, 齋藤邦子, 浅野久子：テキストからの知識抽出の基盤となる日本語基本解析技術, NTT 技術ジャーナル, Vol. 2008.6, pp. 20–23 (2008).

[33] 入江友紀, 松原茂樹, 河口信夫, 山口由紀子, 稲垣康善：意図タグつきコーパスを用いた発話意図推定手法, 人工知能学会言語・音声理解と対話処理研究会資料, 第 38 巻, pp. 7–12 (2003).

[34] 石田栄美, 辻慶太：日本語テキストの自動分類のための特徴素抽出手法の比較, 情報処理学会研究報告自然言語処理 (NL), Vol. 2002, No. 87, pp. 81–86 (2002).

[35] Iwata, T., Watanabe, S., Yamada, T., and Ueda, N.: Topic Tracking Model for Analyzing Consumer Purchase Behavior, in *Proceedings of International Joint Conferences on Artificial Intelligence*, pp. 1427–1432 (2009).

[36] Iwata, T., Yamada, T., Sakurai, Y., and Ueda, N.: Online Multiscale Dynamic Topic Models, in *Proceedings of 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 663–672 (2010).

[37] Jin, X., Zhou, Y., and Mobasher, B.: Web Usage Mining Based on Probabilistic Latent Semantic Analysis, in *Proceedings of 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 197–205 (2004).

[38] Joachims, T.: A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization, in *Proceedings of 14th International Conference on Machine Learning*, pp. 143–151 (1997).

[39] Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features, in *Proceedings of 10th European Conference on Machine Learning*, pp. 137–142 (1998).

[40] 川添佳洋, 小林載, 吉田実, 外山聡一：言語モデルを用いたカーナビゲーション音声認識技術の開発, *PIONEER R&D*, Vol. 19, No. 1, pp. 25–29 (2009).

[41] 北研二, 辻井潤一：確率的言語モデル 第 3 章 N グラムモデル, 東京大学出版会 (1999).

[42] Kitagawa, G.: Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models, *Journal of Computational and Graphical Statistics*, Vol. 5, No. 1, pp. 1–25 (1996).

[43] Kivinen, J., Smola, A. J., and Williamson, R. C.: Online Learning with Kernels, *IEEE Transactions on Signal Processing*, Vol. 52, No. 8, pp. 2156–2176 (2002).

[44] Kleinberg, J.: Bursty and Hierarchical Structure in Streams, in *Proceedings of 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 91–101 (2002).

[45] 倉田岳人, 市川治, 西村雅史：ユーザの発話傾向分析に基づく車載機器操作のための音声入力手法, 電子情報通信学会論文誌, Vol. J93-D, No. 10, pp. 2107–2117 (2010).

[46] Kwak, H., Lee, C., Park, H., and Moon, S.: What is Twitter, a Social Network or a News Media?, in *Proceedings of 19th International Conference on World Wide Web*, pp. 591–600 (2010).

[47] Lafferty, J., Mccallum, A., and Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data, in *Proceedings of 18th International Conference on Machine Learning*, pp. 282–289 (2001).

[48] Leibler, S. K. R.: On Information and Sufficiency, *Annals of Mathematical Statistics*, Vol. 22, pp. 79–86 (1951).

[49] Lin, C., Xue, G., Zeng, H., and Yu, Y.: Using Probabilistic Latent Semantic Analysis for Personalized Web Search, *Springer-Verlag Berlin Heidelberg, LNCS*, Vol. 3399, pp. 707–717 (2005).

[50] Liu, J.: The Collapsed Gibbs Sampler in Bayesian Computations with Application to a Gene Regulation Problem, *Journal of the American Statistical Association*, Vol. 89, pp. 958–966 (1994).

[51] Marlin, B.: Modeling User Rating Profiles for Collaborative Filtering, *Advances in Neural Information Processing Systems*, Vol. 16, pp. 627–634 (2004).

[52] 益岡隆志, 田窪行則：基礎日本語文法, くろしお出版 (1992).

[53] 南和江, 藤井康寿, 土屋雅稔, 中川聖一：大規模コーパスを用いた固有表現抽出手法の検討, 言語処理学会年次大会発表論文集, 第 17 巻, pp. 2–10 (2011).

[54] Mobasher, B., Cooley, R., and Srivastava, J.: Creating Adaptive Web Sites through Usage-based Clustering of URLs, in *Proceedings of 1999 Workshop on Knowledge and Data Engineering Exchange*, p. 19 (1999).

[55] Mobasher, B., Dai, H., Luo, T., and Nakagawa, M.: Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization, *Data Mining and Knowledge Discovery*, Vol. 6, No. 1, pp. 61–82 (2002).

[56] 長尾真, 辻井潤一, 山上明, 建部周二：国語辞書の記憶と日本語文の自動分割, 情報処理, Vol. 19, No. 6, pp. 514–521 (1978).

[57] 長岡諒, 松本光弘, 沼尾正行, 栗原聡：Web における実世界の位置情報類推に関する研究, 人工知能学会全国大会論文集, 第 23 巻 (2009).

[58] Nallapati, R., Ditmore, S., Lafferty, J., and Ung, K.: Multiscale Topic Tomography, in *Proceedings of 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 520–529 (2007).

[59] Ngu, D. W., and Sitehelper, X. W.: SiteHelper: A Localized Agent That Helps Incremental Exploration of the World Wide Web, in *Proceedings of 6th International World Wide Web Conference* (1997).

[60] Nigam, K., McCallum, A., Thrun, S., and Mitchell, T.: Text Classification from Labeled and Unlabeled Documents using EM, *Machine Learning*, Vol. 39, No. 2, pp. 103–132 (2000).

[61] 奥村学：ブログマイニング技術の最新動向, 電子情報通信学会誌, Vol. 91, No. 12, pp. 1054–1059 (2008).

[62] 大淵康成, 畑岡信夫：音声認識技術の実用化に向けた自動車内実環境での評価実験, 情報処理学会研究報告音声言語情報処理 (SLP), Vol. 2006, No. 40, pp. 1–6 (2006).

[63] Pennacchiotti, M., and Popescu, A.-M.: A Machine Learning Approach to Twitter User Classification, in *Proceedings of Fifth International AAAI Conference on Weblogs and Social Media*, Vol. 1, pp. 281–288 (2011).

[64] Pennacchiotti, M., and Popescu, A.-M.: Democrats, Republicans and Starbucks Afficionados: User Classification in Twitter, in *Proceedings of 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 430–438 (2011).

[65] Perkowitz, M., and Etzioni, O.: Adaptive Web Sites: Automatically Synthesizing Web Pages, in *Proceedings of 15th National Conference on Artificial Intelligence*, pp. 727–732 (1998).

[66] Pruteanu-Malinici, I., Ren, L., Paisley, J., Wang, E., and Carin, L.: Hierarchical Bayesian Modeling of Topics in Time-Stamped Documents, *IEEE Transactions on Pattern Analysis Machine Intelligence*, Vol. 32, No. 6, pp. 996–1011 (2010).

[67] Ramage, D., Dumais, S., and Liebling, D.: Characterizing Microblogs with Topic Models, in *Proceedings of 4th International AAAI Conference on Weblogs and Social Media*, pp. 130–137 (2010).

[68] Ramage, D., Hall, D., Nallapati, R., and Manning, C. D.: Labeled LDA: a Supervised Topic Model for Credit Attribution in Multi-Labeled Corpora, in *Proceedings of 2009 Conference on Empirical Methods in Natural Language Processing*, Vol. 1, pp. 248–256 (2009).

[69] Rao, D., Yarowsky, D., Shreevats, A., and Gupta, M.: Classifying Latent User Attributes in Twitter, in *Proceedings of International Workshop on Search and Mining User-Generated Contents*, pp. 37–44 (2010).

[70] Scott, S., and Matwin, S.: Text Classification Using WordNet Hypernyms, in *Proceedings of Conference, Association for Computational Linguistics*, pp. 45–52 (1998).

[71] Sebastiani, F.: Machine Learning in Automated Text Categorization, *ACM Computing Surveys*, Vol. 34, No. 1, pp. 1–7 (2002).

[72] 白木将幸, 伊藤敏彦, 甲斐充彦, 中谷広正：自然発話文における統計的な意図理解手法の検討, 情報処理学会研究報告音声言語情報処理 (SLP), Vol. 2004, No. 15, pp. 69–74 (2004).

[73] 首藤公昭・楢原登志子・吉田将：日本語の機械処理のための文節構造モデル, 電子情報通信学会論文誌, Vol. J62-D, No. 12, pp. 872–879 (1979).

[74] Tang, J., Yao, L., Zhang, D., and Zhang, J.: A Combination Approach to Web User Profiling, *ACM Transactions on Knowledge Discovery from Data*, Vol. 5, No. 1 (2010).

[75] Tipping, M. E.: Sparse Bayesian Learning and the Relevance Vector Machine, *Journal of Machine Learning Research Archive*, Vol. 1, pp. 211–244 (2001).

[76] Wang, C., Blei, D., and Heckerman, D.: Continuous Time Dynamic Topic Models, in *Proceedings of 20th Conference on Uncertainty in Artificial Intelligence*, pp. 579–586 (2008).

[77] Wang, X., and McCallum, A.: Topics Over Time: A Non-Markov Continuous Time Model of Topical Trends, in *Proceedings of 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 424–433 (2006).

[78] Wang, Y., Bai, H., Stanton, M., Chen, W., and Chang, E.: PLDA: Parallel Latent Dirichlet Allocation for Large-Scale Applications, in *Proceedings of 5th International Conference on Algorithmic Aspects in Information and Management*, pp. 301–314 (2009).

[79] Wei, X., Sun, J., and Wang, X.: Dynamic Mixture Models for Multiple Time Series, in *Proceedings of International Joint Conferences on Artificial Intelligence*, pp. 2909–2914 (2007).

[80] Weng, J., E.P. Lim, J. J., and He, Q.: TwitterRank: Finding Topic-Sensitive Influential Twitterers, in *Proceedings of 3rd ACM International Conference on Web Search and Data Mining*, pp. 261–270 (2010).

[81] Wu, X., Yan, J., Liu, N., Yan, S., Chen, Y., and Chen, Z.: Probabilistic Latent Semantic User Segmentation for Behavioral Targeted Advertising, in *Proceedings of 3rd International Workshop on Data Mining and Audience Intelligence for Advertising*, pp. 10–17 (2009).

[82] Xie, Y., and PhohYa, V. V.: Web User Clustering from Access Log Using Belief Function, in *Proceedings of First International Conference on Knowledge Capture*, pp. 202–208 (2001).

[83] Xu, G., Zhang, Y., and Yi, X.: Modeling User Behavior for Web Recommendation Using LDA Model, in *Proceedings of 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technol*, pp. 529–532 (2008).

[84] Xu, G., Zhang, Y., and Zhou, X.: A Web Recommendation Technique Based on Probabilistic Latent Semantic Analysis, in *Proceedings of 6th International Conference on Web Information System Engineering*, pp. 15–28 (2005).

[85] Xu, G., Zhang, Y., and Zhou, X.: Using Probabilistic Latent Semantic Analysis for Web Page Grouping, in *Proceedings of Research Issues in Data Engineering: Stream Data Mining and Applications*, pp. 29–36 (2005).

[86] 吉村健：しゃべってコンシェルと言語処理, 情報処理学会研究報告音声言語情報処理 (SLP), Vol. 2012-SLP-93, No. 4, pp. 1–6 (2012).

[87] CRF++: Yet Another CRF toolkit, *http://crfpp.googlecode.com/svn/trunk/doc/index.html*.

[88] ドコモ ドライブネット ホーム｜NTT ドコモ, *http://docomo-drivenet.jp/*.

[89] Open Graph, *http://developers.facebook.com/docs/opengraph/*.

[90] アップル - iOS 7 - Siri, *http://www.apple.com/jp/ios/siri/*.

[91] YAHOO! JAPAN カテゴリ, *http://dir.yahoo.co.jp/*.

付 録A　Collapsed Gibbs Samplerによるトピック推定

本節では，Collapsed Gibbs Sampler による $\boldsymbol{\beta}$，及び $\boldsymbol{\theta}$ の推定手法について述べる．まず，[27] で提案されている以下の式に基づき，$\boldsymbol{k}$ を推定する．

$$\boldsymbol{P(k_{dn} = k | k^{-dn}, w) \propto (n_{dk}^{-dn} + \alpha) \cdot \frac{n_{vk}^{-dn} + \eta}{n_k^{-dn} + |W|\eta}} \tag{A.1}$$

上記は，$\boldsymbol{k_{dn}}$ の更新式であり，ランダムにサンプリングした $\boldsymbol{w}$ に対して繰り返し計算する．ここで，$\boldsymbol{k_{dn}}$ は，文書 $\boldsymbol{d}$ の $\boldsymbol{n}$ 番目の単語が持つトピック，$\boldsymbol{n_{dk}}$ は文書 $\boldsymbol{d}$ のトピック $\boldsymbol{k}$ を持つ単語数，$\boldsymbol{v}$ は文書 $\boldsymbol{d}$ の $\boldsymbol{n}$ 番目の単語であり，$\boldsymbol{n_{vk}}$ はトピック $\boldsymbol{k}$ を持つ単語 $\boldsymbol{v}$ の数であり，$\boldsymbol{n_k}$ はトピック $\boldsymbol{k}$ を持つ全単語数であり，$\boldsymbol{-dn}$ は，当該サンプリング対象の単語を除くことを意味する．

次に，$\boldsymbol{\beta}$，$\boldsymbol{\theta}$ の各要素を以下の式で導出する．

$$\boldsymbol{\beta_{kw} = \frac{n_{vk} + \eta}{n_k + |W|\eta}} \tag{A.2}$$

$$\boldsymbol{\theta_{dk} = \frac{n_{dk} + \alpha}{n_d + |K|\alpha}} \tag{A.3}$$

ここで，$\boldsymbol{n_d}$ は文書 $\boldsymbol{d}$ の単語数である．

付 録B　発話文からのクエリ抽出

本節では，4 章で述べたシステムにおけるクエリ抽出について述べる．クエリ抽出は基本的には，大量の辞書を利用したキーワードマッチにより行う．ただし，本システムにおけるクエリ抽出は，辞書により抽出可能な単純な地名だけではなく，電話番号や郵便番号などの数字の羅列からも，一意の地図データを取得する．これは，正規表現を利用して，実現している．

さらに，固有名詞と名詞や名詞接尾辞など，複数の形態素の組合せによりクエリとなる場合がある．例えば,「横浜駅西口」は，固有名詞である「横浜駅」と名詞接尾辞の「西口」に形態素が分割される.「横浜駅」ではなく「横浜駅西口」をクエリとして抽出することを考えた場合，形態素に対するキーワードマッチで行うことは出来ない．例えば形態素単位ではなく発話文全体に対するキーワードマッチを行う方法も考えられるが，マッチングの処理が増大してしまい，現実的ではない.「横浜駅西口」をあらかじめ辞書に登録することで，この問題は解決可能であるが，他にも「ドコモ本社」など，固有名詞と名詞の組合せによりクエリとなる例は数多く存在するため，全てを辞書で網羅することも難しい.

上記は，発話文を構成する形態素の系列から，クエリとなる形態素を抽出する問題となる．このような問題に対しては，CRF（Conditional Random Fields）[47] が適している．例えば，[53] [57] では CRF を利用して，文章から地名の抽出を行っている．CRF は系列ラベリングのアルゴリズムであり，ラベルが付与された形態素の列で構成される文の集合を収集し，ラベル付き形態素の前後関係を学習することで，入力された文の各形態素に対してラベルを付与することが可能である．CRF を対話システムのクエリ抽出に適用した場合，複数の形態素の組合せで構成されるクエリを抽出するために，クエリが開始される形態素とこの形態素に連なる形態素のそれぞれにラベルを付与し，これらを連ねて抽出すれば良い.

表 B.1: CRF による学習時の特徴ベクトル

横浜駅/西口/の/マクドナルド/に/行く/たい

表層	品詞	属性	ラベル
＊	名詞	地名	B
西口	名詞接尾辞		I
の	格助詞		O
＊	名詞		O
に	格助詞		O
行く	動詞		O
たい	動詞接尾辞		O

ドコモ/本社/に/案内/して

表層	品詞	属性	ラベル
＊	名詞	組織名	B
本社	名詞		I
に	格名詞		O
案内	名詞		O
して	動詞接尾辞		O

本システムでは，CRF の実装として CRF++ [87] を利用した．また，学習コーパスとしてクエリを含む 860 の発話文を収集し，モデルを生成した．この際素性は，表 B.1 に示すように，前処理済みの各形態素を，表層，品詞，さらに固有名詞の場合は地名，組織名，人名など独自に定義した属性で表現し，これらの前後関係から特徴ベクトルを生成した．また，無数の固有名詞に対して一般性を保った学習となるように，固有名詞の表層を「＊」に変換しマスクした．さらに，各形態素には，地名が開始する形態素には「B」,「B」から開始する地名に含まれる形態素には「I」，それ以外には「O」といったラベルを付与した．こうして学習した CRF のモデルにより，上記 860 文例の交差検定においては，正しいクエリを 100%抽出可能となった．

付 録C　ドコモドライブネット

4 章で提案したシステムを元に，NTT ドコモは，2012 年 12 月に，ユーザの自然で多様な発話によって，音声操作可能なカーナビサービスであるドコモドライブネットの提供を開始している [88]．本節ではその概要について述べる．

ドコモドライブネットは，図 4.1 に示した対話システムをそのまま実現したものである．ユーザインタフェースは，スマートフォン上で実装されている．図 C.1 に示す通り，近隣の施設の地図上への表示，地図による目的地までの案内，音量や地図縮尺の変更など，通常のカーナビと同等の操作を，自然発話により行うことがが可能である．発話により操作可能な機能の一覧は，表 4.1, 4.2 に示した通りである．また，音声認識機能，タスク判定機能，クエリ抽出機能は，それぞれサーバ装置として実装されており，特にタスク判定機能は 4 章で提案した手法を利用している．

図 C.1: ドコモドライブネットのユーザインタフェース