SNS グラフデータにおける文脈を考慮した 適合フィードバック検索

片岡　大祐† 　加藤　誠†† 　山本　岳洋†† 　大島　裕明†† 　田中　克己††

† 京都大学工学部情報学科　〒 606–8301 京都府京都市左京区吉田本町
†† 京都大学大学院情報学研究科　〒 606–8301 京都府京都市左京区吉田本町
E-mail: †{kataoka,kato,tyamamot,ohshima,tanaka}@dl.kuis.kyoto-u.ac.jp

あらまし　Twitter データは，ショートテキストである各ツイートに対してリプライ等関連するデータが関連づけられた，構造化されたグラフデータであると見なすことができる．本稿では，グラフデータである SNS データから，適合するツイートを検索・ランキングする手法を提案する．SNS データに対するキーワード検索は，投稿ツイート自身がショートテキストでありその中に情報発信者のプロファイル情報も表現されていないため，検索性能を向上させることが難しい．本研究では，Twitter 検索を，ツイートとそれに関連づけられた他のツイートやフォロー情報を対象ツイートの「コンテキスト」（部分グラフ）とみなし，コンテキスト中の語の重みを伝播させて対象ツイートの適合度スコアを計算して検索順位を決定する手法を提案する．コンテキストは検索質問に大きく依存するため，本手法では、検索結果に対し正例・負例を与える適合フィードバックを行って，適切なコンテキストを決定する．

キーワード　グラフデータ　コンテキスト　適合フィードバック

1. はじめに

近年，LINE(注1)や Facebook(注2)のようなソーシャルネットワークサービス（SNS）の需要が高まっており，多くの人々が SNS 上で情報を発信しあっている．企業は自社が開催するイベントや自社の新商品の宣伝を SNS 上で行う一方，実際に体験した人や関心を持っている人がそれらのイベントや新商品について SNS に投稿することがよくある．特に，代表的な SNS であり 3 億人を超える月間アクティブユーザを有する Twitter(注3)では，イベントに参加した直後や商品を CM で見たときなど，人々がその場で感じたことをその瞬間に，手軽に投稿できるため多くの人々が活用している．また，特定の投稿者のプロファイルを持つ人々の投稿の分析が企業の販売戦略に役立つ場合がある．例えば，イベントや商品を支持する年代層や性別，商品について興味を持っている人の意見や，実際にイベントに参加した人の意見などを知ることで，投稿者のプロファイルに基づき販売戦略を立てることができる．

しかし，ツイートの投稿者がどういう人間か，言及している対象とどのように関わったかなどの情報をキーワードによって指定し検索をすることは難しい．Twitter には投稿文字数制限があるため文章量が少なく，あえて投稿者が自分の情報をツイート中に書き込むことは稀なためである．そのような場合，「検索結果に関連付けられた情報同士が互いにリンクで繋がっているため，1 つの情報から別の情報へと辿ることができる」という SNS の特徴を利用することで，以下に述べる手順で検索者は投稿者のプロファイルを推定することができる．まず，検索者は直接求めたい情報をクエリにより指定せず，求めたい情報を含み，より広い範囲のツイートまで網羅できるクエリを投入する．次に，検索結果中の各ツイートに付随するコンテキスト，つまり投稿者の自己紹介文や同じ投稿者の他の投稿，投稿者と繋がりのある別の投稿者の自己紹介文や投稿など，検索結果に関連付けられた情報を確認しながら，求める評判情報の発信者像と合致しているかを判断する．投稿者のプロファイルを考慮した検索をするためには，これらの作業を各検索結果に対して行いながら，検索結果を取捨選択していく必要がある．

例えば，自分の授業の評判を知りたい京都大学の教員が授業名で検索したとする．すると，その教員の授業に加え，他の大学の同じ名前の授業も検索結果中に含まれる．投稿者が京都大学の学生であることを確かめるために，まず，ツイートを投稿した投稿者の自己紹介文中に京都大学や京大と記載されているか確認する．投稿者の自己紹介文を見て分からなければ，フォロワーの自己紹介文や過去のツイートに京都大学に関わる単語が頻出するかどうかを調べることで投稿者が京都大学の学生であるかを判定する．他には，最近オープンした飲食店に「実際に行った人」の評判についても同様である．このように検索結果のツイートに関連付けられた情報を確認することで投稿者のプロファイルを推定することができる．しかし，検索結果の全てのツイートに対し，その操作をするのは多大な労力を要する．

そこで本論文ではこの問題を解決するために，前述の手動で行う適合性の判定方法をもとに次の 2 つの仮説を考える．

- 検索結果に関連付けられたコンテキスト中には発信者のプロファイルを表す情報が含まれている
- 発信者のプロファイルの推定には検索結果の近辺の情報ほど参考になり，検索結果から離れるほどノイズが増大する

(注1)：https://line.me
(注2)：https://www.facebook.com
(注3)：https://twitter.com/

この 2 つの仮説に基づき，SNS 中の検索結果とコンテキスト中の情報を合わせてグラフと見なし，検索者からのフィードバックに基づいて検索結果の再ランキングを行う手法，CRFG(Context-aware Relevance Feedback over SNS Graph data) システムを提案する．CRFG は Rocchio らが提案した適合フィードバック [1] に基づいた手法である．CRFG では，2 種類のクエリ，コンテンツクエリとコンテキストクエリを用いる．コンテンツクエリは検索結果の本文中からキーワードを探すためのクエリである．コンテンツクエリは，既存の検索システムに入力として与える．コンテキストクエリは，検索結果に現れない，あるいは現れにくいキーワードを指定するためのクエリである．前述の 1 つ目の例の場合，コンテンツクエリは授業名，コンテキストクエリは「京都大学」であり，2 つ目の例の場合，コンテンツクエリは飲食店名，コンテキストクエリは「行ったことがある」や「行った」などである．

CRFG を用いたシステムは次の手順で動作する．

（1） コンテンツクエリを入力して初期検索結果を取得．

（2） 得られた検索結果をもとに SNS グラフを生成．

（3） 各ノードの単語の重みを計算し，その重みを検索結果ノードまで減衰させながら伝播させ，文書ベクトルを生成．

（4） 検索結果とコンテキストを検索者が 1 つずつ閲覧し，システムに正例・負例・判定不可のフィードバックを与える．この際，適合性の判定にはコンテキストを参考にする．

（5） 与えられた正例・負例のフィードバックを元にコンテキストクエリを修正．

（6） 修正したコンテキストクエリと各検索結果の文書ベクトルとの類似度を算出し，対象ツイートの適合度スコアを計算．

（7） 適合度スコアを元に検索結果を再ランキングする．

提案手法の有効性を評価するために，10 種類のコンテンツクエリとコンテキストクエリのペアからなるテストコレクションを構築した．また，事前に各クエリの 100 件のツイートを正例・負例，あるいは判定不可の 3 種類に分類した．このテストコレクションを用いたシミュレーションによる評価を行った結果，提案手法を用いることでコンテンツのみをフィードバックとして与える既存の適合フィードバックと比べて平均 nDCG@10 が 0.337 から 0.350 に向上した．

我々の主な貢献を以下にまとめる．1) コンテンツとクエリの語彙が少ない場合に検索精度が低下する問題を解決する検索手法を提案した．2) コンテキストと伝播率を用いる適合フィードバックにより，検索の際に用いる語彙を増やす手法を提案した．3) テストコレクション，および，CRFG の評価の枠組みを構築し，提案手法の有用性を示した．

本論文の構成は以下のとおりである．2 節では，関連研究を紹介する．3 節では，本研究で用いる概念について説明を行う．4 節では，提案手法について述べる．5 節では提案手法に対する実験と評価について述べる．6 節では，今後の課題及び本論文での結論について述べる．

2. 関 連 研 究

Twitter で投稿文書以外の情報源に含まれる語彙を利用することで，属性推定の精度向上を試みる研究が行われている．奥谷ら [2] は，ユーザ間のやりとりであるメンション情報を利用して Twitter ユーザのプロファイル推定を行った．Lu ら [3] は，メンション中の抽象的な表現の属性推定を行った．Lu らの手法では，不足する語彙を補うために，ユーザ間の他のメンションや Wikipedia からの語彙を利用している．上里ら [4] は，推定対象となるユーザの相互フォロー，相互メンション関係にある周辺ユーザの属性情報を補完することで，推定対象ユーザの属性推定精度を向上させた．また，上里らは, [4] を発展させた研究として Personalized PageRank を用いて推定対象ユーザと周辺ユーザの関連度を算出し，関連度に応じて単語の重みを変化させることで属性推定の精度を向上させる手法を提案した [5]．池田ら [6] は，ツイート投稿者の普段のツイート傾向を解析することでプロファイルの推定を行った．

また，マイクロブログ文書に対して擬似適合フィードバックを用いることで検索精度を向上させる研究も行われている．マイクロブログ上での疑似適合フィードバックによるクエリ拡張には上位の検索結果の多くが非適合文書である場合に，ユーザクエリに関係ない単語が選ばれてしまう可能性があるという問題が存在する．宮西ら [7] は，検索結果のマイクロブログ文書の中から 1 つ正解文書を検索者が選択し，その文書をクエリ拡張に用いて再検索した検索に対し擬似適合フィードバックを適用することで検索精度の向上を図った．Whiting ら [8] はマイクロブログの文書に対し，PageRank を用いて文書中の単語に時間性を考慮した重み付けを行い，時間性を考慮した疑似適合フィードバックを行うことで検索性能が向上することを示した．

また，マイクロブログ上での評判情報を検索する際に検索結果中に含まれるノイズを除去する研究について述べる．伊川ら [9] は，Twitter 上で特定のイベントに言及する投稿の中で，キーワードのイベントとの関連度をもとにノイズツイートを除去する手法を提案した．

ウェブ上のデータの検索においても異なるデータセット間での検索に関する研究が行われている．Halpin ら [10] は Semantic Web の検索とハイパーテキストの Web データの検索において一方の検索結果のフィードバックを他方の検索に用いることで検索精度を向上させることができることを示した．Herzig ら [11] は，1 つのデータセットの語彙のみを用いてクエリを拡張することで，RDF で記述された Web 上の異なるデータセット間の検索を行う手法を提案した．Shekarpour ら [12] は，検索者が与えたキーワードを適切に解釈するために，予め定義された基本グラフパターンテンプレートを用いて，SPARQL クエリを生成する手法を提案した．Su ら [13] は，自然言語で記述されたクエリをグラフデータに変換し，Knowledge Graph での検索により得られたデータに対し適合フィードバックを行うことで検索性能が向上することを示した．

本研究では，クエリ拡張にツイートが投稿された時間付近の同ユーザの投稿を用いているが，同様に，時間を考慮した情報検索の研究も存在する．Massaoudi ら [14] が提案した手法では，クエリ拡張にクエリが生成された時間付近に使用された単語に大きい重みを与えている．

本研究では，ツイートに関連付けられたデータをグラフして不足する発信者の情報を補うという点で，奥谷ら [2]，上里ら [4] [5] 池田ら [6] の研究に，検索者が適合と判定したツイートをクエリ拡張に用いるという点で，宮西ら [7] の研究に類似している．宮西らの研究ではツイート本文をフィードバックとして与えているが，マイクロブログでは 1 つの投稿あたりの文書が短いという問題がある．本研究では，この問題を解決するアプローチとして [2] [4] [5] [6] を合わせてツイートの投稿者の不足する情報を補う．また，本研究ではコンテキストをクエリ拡張に使用し，グラフ上でツイート本文との距離が離れるほど単語の重みを減衰させるという点で前述の研究と異なる．

3. 概念説明

本節では，本研究における基盤となる概念について説明する．まず，本研究で使用する 2 種類のクエリについて述べる．次に，本手法におけるグラフの生成方法を Twitter を例にして述べる．

3.1 コンテンツクエリ・コンテキストクエリ

本節では，本研究で用いる 2 種類のクエリに関して述べる．従来の検索システムでは検索対象の文書中の単語を探すために 1 種類のクエリを使用する．従来のクエリと同様の意味合いを持つ検索対象の文書に対する問い合わせに用いるクエリを**コンテンツクエリ** $\mathbf{q}_{\text{contents}}$ とする．もう一つはコンテキスト中から検索者の求める情報発信者像を検索するために用いる，**コンテキストクエリ** $\mathbf{q}_{\text{context}}$ である．具体例を挙げると，データベースの授業を担当する教授が大学生が発信者のデータベースに関する投稿を見たい状況を想定する．その際ツイートの投稿者は「私は大学生で，データベース楽しみ」とは記述せず，「データベース楽しみ」と呟くことが考えられる．そこで，コンテンツクエリとしては「データベース」，コンテキストクエリとしては「大学生」と入力する．さらに探したい対象を教員にしたい場合はコンテキストクエリに「教授」や「Professor」，さらに絞りたい場合は「京都大学」などと入力する．

3.2 SNS グラフ

本節では，SNS 上のデータから作成したグラフの定義と具体例について述べる．検索結果のツイートから辿ることのできる，ツイートに関連付けられた要素をノードと見なし，互いに関連付けられたノード同士をエッジで繋いだグラフを **SNS グラフ**と呼ぶ．コンテンツクエリを Twitter の検索システムに入力し n 件からなる検索結果のツイート列 $v_{0,1}v_{0,2}\dots v_{0,n}$ が得られたとする．このとき CRFG は i 番目の検索結果 $v_{0,i}$ に対しラベルつき有向グラフ $G=(V,E,L)$ を生成する．ノード $V=\{V_0,V_1,\dots,V_k\}$ は $k+1$ 種類からなり，検索結果を除いてコンテキスト中に種類のノードが k 種類存在することを表す．ただし V_0 は $V_0=\{v_{0,1},v_{0,2},\dots v_{0,n}\}$ と表される検索結果の集合である．エッジは $E\subset V\times L\times V$ と定義され，$e=(v_{0,1},l_1,v_{1,1})$ は 1 番目の検索結果のノード $v_{0,1}$ からノード $v_{1,1}$ に，l_1 というラベルのついたエッジが張られていることを表す．Twitter におけるノードには以下のような種類がある．

user:　ツイートの投稿者．各ユーザは自己紹介文を記述．

followee・follower:　ユーザの一種．あるユーザがフォローしている人をフォロイー，フォローされている人をフォロワーと呼ぶ．フォローとは，あるユーザが別のユーザのツイートを自分のタイムラインで見られる状態にすること．

tweet:　各ユーザがした検索結果に含まれる投稿．

user_tweet:　検索結果の投稿をした投稿者による他の投稿．

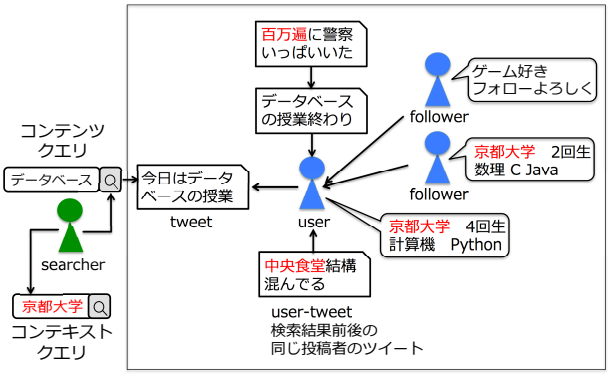

図 1　SNS グラフの概念図

各ノードには単語集合が付随し，ノードに現れる文章を形態素解析して得られる．エッジにはラベルがついており，本研究では $L=\{\text{post},\text{follow},\text{nearby-post},\text{connect}\}$ の 4 種類のラベルを用いる．次にエッジの張られ方について述べる．まず，投稿者は検索結果中に含まれるあるツイートを投稿している場合，user ノードから tweet ノードに，post というラベルのついたエッジを張る．次に，ある投稿者が別の投稿者をフォローしている場合，follower ノードから user ノードに，follow というラベルのついたエッジを張る．次に，投稿者は検索結果のツイート以外にも投稿している場合，user ノードから user_tweet に，nearby-post というラベルのついたエッジを張る．最後に，検索結果中の投稿を除き，同じ投稿者による複数の投稿がある場合，前後の投稿同士，つまり user_tweet から user_tweet に connect というラベルのついたエッジを張る．SNS グラフの概念図を図 1 に示す．

4. 提案手法

本章では我々が提案する手法について詳細を述べる．本研究の目的は，コンテンツに加え，コンテキストを用いた適合フィードバックを提案することで，SNS の検索対象の文章量が少ないことによる検索精度の低下を解決することである．従来の適合フィードバックは，検索結果の情報のみに基づき検索者が適合性の判定を行う．検索者により与えられたコンテンツのフィードバックを用いてコンテンツクエリを修正することで，検索結果を再ランキングする．

一方，本研究で提案する手法は，以下の 3 点において従来の適合フィードバックと異なる．1 点目はクエリ修正と類似度計算に用いる文書ベクトルに，コンテキスト中の単語を加えることである．例えば，Twitter の場合だとユーザの自己紹介文，過去のツイート，フォロワーの自己紹介文中の語彙を文書ベクトルに加える．2 点目は，投稿者のプロファイルを検索するためにコンテキストクエリを用いることである．検索者が入力し

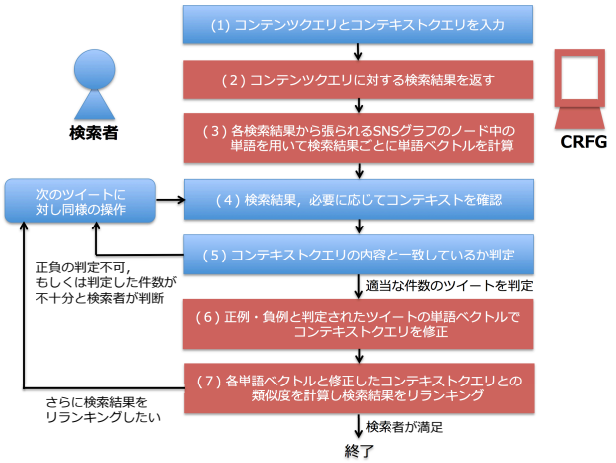

図 2　手法の流れ

たコンテキストクエリを使ってコンテンツとの類似度を計算した場合，件数が得られないことが考えられるため，本手法では適合フィードバックによりコンテキストクエリの語彙を増やし，コンテンツとの類似度を計算する．3 点目は，コンテキストのノード中の各単語について重みを計算し，その重みを検索結果ノード $v_{0,i}$ まで伝播させることである．このノードが，SNS グラフで $v_{0,i}$ から離れているほど単語の重みは減少する．より具体的には，単語が存在するノードから $v_{0,i}$ にたどり着くまでにエッジを通るたびに単語の重みが減衰する．以上の 3 点に基づき，CRFG のフレームワークを構築した．

4.1　システム動作

この節では CRFG を用いたシステムの，具体的なシステム動作について述べる．CRFG の動作を図 4.1 に示す．図 4.1 に基づき，システム実行の具体例を示す．検索者が大学生が投稿したのデータベースの授業に関するツイートを取得したいとする．ツイートの投稿者は自分が大学生であること，授業という単語をツイート本文に明記しない可能性が考えられるので (1) 検索者はコンテンツクエリ「データベース」，コンテキストクエリ「大学生 授業」という入力をシステムに与える．(2) システムは「データベース」という単語を本文中に含むツイートからなる検索結果を返す．(3) システムは検索結果を取得した際に，検索結果ごとにコンテンツ・コンテキストの単語から文書ベクトルを計算する．(4) 検索者は 1 件ずつ表示される検索結果に対しツイート本文や投稿者の自己紹介文，過去のツイートやフォロワーを閲覧し「大学生」を示す内容が含まれているかを確認する．(5) ツイートの発信者が大学生であると推定されれば適合，大学生でないことが判明すれば不適合と判定し，正例負例のフィードバックをシステムに与える．コンテキストを確認して判定できなかった場合は，次の検索結果に対して (4) の操作を行う．適当な件数のツイートの適合性を判定し，(6) に進む．(6) システムは正例・負例と判定した検索結果の文書ベクトルを用いてコンテキストクエリを修正する．正例の文書ベクトルには「大学生」を表す単語に高い重みが与えられていることが期待される．「大学」や「Univ」といった単語は，投稿者の自己紹介文，フォロワーの自己紹介文中に記載されており，「試験」や「単位」，「授業」といった単語は投稿者の他のツイート中に含まれている．(7) システムは，修正したコンテキストクエリと各検索結果の文書ベクトルの類似度を計算し，検索結果を再ランキングする．システムは上位に大学生が投稿したツイートが並ぶ検索結果を出力する．

4.2　単語の重み

本研究では，各検索結果の文書ベクトルを計算する前段階の処理として，SNS グラフの各ノードの全ての単語の重みを計算する．本研究では，単語の重みの計算に TF-IDF を用いる．あるノード v 中のある単語 t の TF-IDF による重み $w(v,t)$ は以下の方法で計算される．$\mathrm{tf}_{v,t}$ はノード v における単語 t の出現回数とし．idf_t は以下の式に従い算出する．idf_t は以下の式に従い算出する．

$$\mathrm{idf}_t = \log_2 \frac{|V|}{n_t} \tag{1}$$

ただし，V は全ての検索結果から張られた SNS グラフ上の全ノード集合，n_t はある単語 t を含むノードの数を表す．

単語 t に対する重み，TF-IDF スコアは，TF スコア $\mathrm{tf}_{v,t}$，idf_t を用いて以下の式 (2) に従い算出する．

$$w(v,t) = \mathrm{tf}_{v,t} \cdot \mathrm{idf}_t \tag{2}$$

4.3　適合度スコア

検索結果の各ツイートがコンテキストクエリとどの程度一致しているかを表すスコアを**適合度スコア**とする．適合度スコアが高い順に検索結果を再ランキングする．適合度スコアを計算するスコア関数 f は以下の式で定義される．

$$f(\mathbf{q}_{\mathrm{context}}, \mathbf{x}_{0,i}, P, N) = \cos(\mathbf{q}'_{\mathrm{context}}, \mathbf{x}'_{0,i}) \tag{3}$$

ここで $\mathbf{q}_{\mathrm{context}}$ はコンテキストクエリのベクトル表現，$\mathbf{x}_i$ は i 番目の検索結果のコンテンツの文書ベクトル，$\mathbf{x}'_i$ は i 番目の検索結果 $v_{0,i} \in V_0$ のコンテンツとコンテキスト中の全てのノードから生成される文書ベクトルである．P は正例集合，N は負例集合であり，それぞれ n 件の検索結果の文書ベクトルの集合 $\{\mathbf{x}'_{0,1}, \mathbf{x}'_{0,2}, \ldots, \mathbf{x}'_{0,n}\}$ の部分集合である．$\mathbf{q}'_{\mathrm{context}}$ は P, N のフィードバックによって修正を加えたコンテキストクエリのベクトル表現である．スコア関数 f を用いて，正例・負例の文書ベクトルを用いて修正したコンテキストクエリと各検索結果の文書ベクトルとコサイン類似度を計算し，全ての検索結果のスコアを求める．

i 番目の検索結果 $v_{0,i}$ の文書ベクトル $\mathbf{x}_{0,i}$ の算出方法を述べる．$v_{0,i}$ から張られた 1 つの SNS グラフ G 中の全ノードを V とし，全ノードの語彙を N とすると $\mathbf{x}$ は N 次元のベクトルとなる．$\mathbf{x}$ は V 中の各ノードに付随する全ての単語の重みを累積することで求められる．SNS グラフ上の各ノード v から $v_{0,i}$ への単語のについて概念図を図 3 に示す．

各単語の重みは 4.2 節で述べた方法で予め計算し，さらにノード間をホップする際に，通るパスの種類に応じて単語の重みを減衰させる．ノード v からノード v' へのパス $P_{v \to v'}$ における伝播率の積 $\delta(P_{v \to v'})$ を以下の式で定義する．

$$P_{v \to v'} = (e_1, e_2, \ldots, e_n) \tag{4}$$

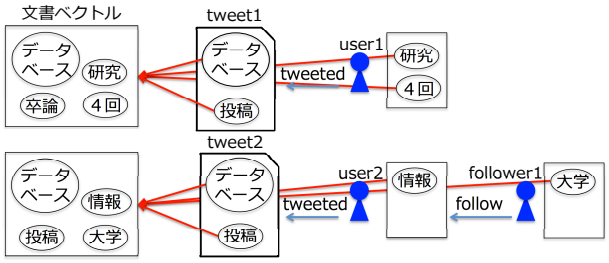

図 3　単語の伝播の概念図

$$\delta(P_{v\to v'}) = \prod_{i=1}^{n} d(l_{e_i}) \qquad (5)$$

ただし，$(e_1, e_2, \ldots, e_n)$ は v から v' に到達するまでに通るエッジを順に並べたタプル，l_{e_i} はエッジ e_i に付随するラベル，n はノード v からノード v' に到達するまでに通るエッジの数である．$d(l_{e_i})$ はラベルの種類ごとに予め定められる伝播率であり，$0 \leqq d(l_{e_i}) \leqq 1$ を満たす．具体的には user ノードと tweet ノード間のエッジに張られたラベル post，user_tweet ノードと user ノード間のエッジに張られたラベル nearby-post を用いて $d(\text{post}) = 0.8$，$d(\text{nearby-post}) = 0.5$ とすると，$\delta(p_{\text{user\_tweet}\to\text{tweet}}) = d(\text{nearby-post})\cdot d(\text{post}) = 0.5\cdot 0.8 = 0.4$ となる．

本研究で扱う SNS グラフは木構造のため，パスが閉路になることはない．そのため，i 番目の検索結果 $v_{0,i}$ から張られる SNS グラフ中の全ての文書ベクトル $\{\mathbf{x}_{0,i}, \mathbf{x}_{1,i}, \ldots, \mathbf{x}_{|V|,i}\}$ と予め設定した伝播率の積 δ，パラメータ α を用いて i 番目の検索結果の文書ベクトル $\mathbf{x}'_{0,i}$ は以下の式で表される．

$$\mathbf{x}'_{0,i} = (1-\alpha)\mathbf{x}_{0,i} + \alpha \sum_{v_{j,i}\in V\setminus v_{0,i}}^{|V|} \delta(P_{v_{j,i}\to v_{0,i}})\mathbf{x}_{j,i} \qquad (6)$$

α はコンテンツとコンテキストの単語の重みの与え方の割合を調整するパラメータであり，$0 \leqq \alpha \leqq 1$ を満たす．

$\mathbf{q}'_{\text{context}}$ の算出方法について述べる．$\mathbf{q}'_{\text{context}}$ は $\mathbf{q}_{\text{context}}$ を正例・負例の文書ベクトルを用いて修正したベクトルであり，以下の式で表される．

$$\mathbf{q}'_{\text{context}} = \mathbf{q}_{\text{context}} + \frac{1}{|P|}\sum_{\mathbf{x}'_{0,k}\in P}^{|P|} \mathbf{x}'_{0,k} - \frac{1}{|N|}\sum_{\mathbf{x}'_{0,k}\in N}^{|N|} \mathbf{x}'_{0,k} \qquad (7)$$

式 (6) において，$\alpha = 0$ の場合，第二項が 0 になるため，文書ベクトルはコンテンツの単語のみから生成される．$\alpha = 0$ の場合の単語ベクトルを式 (7) に適用すると従来の適合フィードバックと見なすことができる．

式 (7) は正例，負例の文書ベクトルを用いてクエリの文書ベクトルを修正するという点では既存の適合フィードバックと同じであるが，修正に加える文書ベクトルの単語をコンテンツ以外から取得する点，重みを計算するのに伝播率を用いる点が，従来の適合フィードバックと異なる．

このようにして求められた $\mathbf{q}'_{\text{context}}$ と文書ベクトル $\mathbf{x}'_{0,i}$ とのコサイン類似度をとることにより，i 番目の検索結果の適合度スコアが計算できる．

5. 実　　験

本節では，提案手法を用いて行った実験とその評価について述べ，結果についての考察を行う．本実験の目的は提案手法の有効性について検証することである．

5.1 概　　要

本研究では実験 1，実験 2 に分けて評価実験を行った．

実験 1 では，初めに (1)Twitter の現在の検索機能，(2) コンテンツのみからなる文書ベクトルを用いてコンテンツクエリを修正した従来の適合フィードバック，(3)(1) により得られた検索結果に対しコンテキストクエリを用いて再検索する手法，(4) コンテキストクエリにコンテンツの文書ベクトルを加える適合フィードバックの 4 つの手法を比較する．(1) と (2) の手法の比較により，適合フィードバックの導入による精度の変化を調べ，(1) と (3) の手法の比較により，コンテキストクエリの有無による精度の変化を調べる．(4) は式 (6) 中の $alpha = 0$ とした場合と同様であり，(2) がコンテンツクエリを修正するのに対し，(4) ではコンテキストクエリを修正する．

次に，式 (6) 中の α を $0.25, 0.5, 0.75, 1.0$ の 4 段階に変化させた場合の精度の変化を調べる．$\alpha > 0$ の CRFG では，コンテキストを文書ベクトルに加える．仮説 1 に基づきコンテキストを文書ベクトルに加えることで，精度が向上することを検証する．

実験 2 では CRFG において，エッジに付随するラベルの種類ごとに伝播率 d をパラメータとして変化させる．この実験では，仮定 2 に基づき，

- 伝播率の設定により精度が向上すること
- 最適な伝播率が存在すること

を示す．

5.2 実 験 設 定

5.2.1　評価データ

実験には 25 件のクエリを用いた．クエリの例を表 1 に示す．同じコンテンツクエリに対しても異なるコンテキストクエリが存在することも考えられるので．コンテンツクエリの重複は認めた．P@10 はフィードバックに用いた上位 10 件中の正解数であり，P@100 は取得したツイート 100 件中の正解数である．また，検索者がコンテキストクエリにより取得したい情報を表中に記載した．

今回の実験は Twitter を対象とした．Twitter REST API(注4) を利用し，各クエリに対し，ツイートの取得日時に近い順に 100 件のツイートを取得した．グラフのノードに使用するデータとして各ツイートに対して投稿者の自己紹介文，フォロワー，検索結果のツイートの前後の投稿を取得した．API の制限から，投稿者のフォロワー数が 1200 人以下の場合は全員，1200 人を超える場合 1200 人ずつフォロワーの自己紹介文を取得した．投稿者の検索結果の前後のツイート (user_tweet) はそれぞれ最大 25 件ずつ取得した．取得した全てのツイートに対し，ツイート本文とコンテキストを確認し，予め手動で適合性判定を

(注4)：https://dev.twitter.com/rest/public

表 1　実験に使用したクエリの例

	コンテンツクエリ	コンテキストクエリ	P@10	P@100	求める情報
q_1	センター試験	受験生	0	8	受験生の感想
q_2	物理学基礎論	京都大学	3	15	京都大学の授業に関する投稿
q_3	PSVR	やってみたい	5	44	興味を持っている人の投稿
q_4	センター試験	解いた人	1	13	今年のセンター試験を解いた人の感想
q_5	貴腐ワイン	飲んだことがある人	5	48	飲んだことがある人の投稿

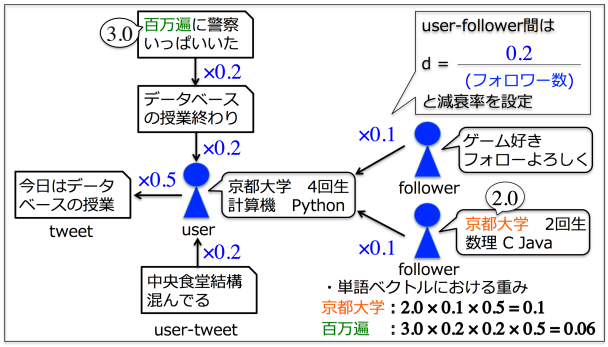

図 4　実験に使用したグラフ

表 2　グラフのノード・エッジ数とコンテキストの有無による語彙

ノード数	エッジ数	語彙 (コンテキスト無)	語彙 (コンテキスト有)
31547.64	32985.52	**731.68**	**64580.96**

行った．コンテキストクエリと一致しているツイートを正例，コンテキストクエリに一致していないツイートは負例，コンテキストを確認しても判定できなかった場合は判定不可として負例と同様の扱いとした．

5.2.2　適合フィードバックの方法

取得した 100 件のツイートを最も新しいツイートから順に並べ，上位 10 件のツイートをフィードバックを用いる．今回の実験では，10 件のフィードバックでクエリを修正し，1 度再ランキングした結果に対して評価値を計算する．評価値の算出には最初にフィードバックに用いた 10 件を省いた 90 件のツイートを用いる．

5.2.3　SNS グラフ

実験に用いた SNS グラフは図 4 のようになる．今回の実験では，tweet ノードと user ノード，user ノードと各 follower ノードをエッジで接続した．user_tweet ノードは検索結果の投稿に最も近い時刻の投稿を user ノードと接続し他の user_tweet ノードは最も投稿された時間が近い user_tweet ノード同士を接続した．また，表 2 に実験 1，実験 2 に使用した各クエリのノード数，エッジ数の平均，コンテンツのみの語彙とコンテンツにコンテキストを加えた語彙の平均を示す．コンテキストがある場合の語彙はコンテンツのみの場合に比べて平均で約 88 倍の語彙になることが読み取れる．

5.2.4　ツイート取得時の検索設定・ツイートの前処理

本節ではツイートの取得時の検索設定，取得したツイートに対する前処理について述べる．取得するツイートの種類を増やすために，TwitterAPI のオプションを用いてリツイート (注5) は除外した．また，他の言語を許容するとノイズが極端に多くなってしまうため，日本語の検索結果のみを取得した．さらに取得したツイートから，TwitterAPI のオプションで除外することのできない以下のような場合の投稿を削除した．

・ URL を含む投稿．URL を含む投稿は Web サイトの情報を再配信しているものが多いため，URL を含む投稿を除外した．

・ 同じ投稿者が 2 回以上，同じ内容を投稿している場合．全く同じ文章を同じ投稿者が投稿した場合，bot(自動的に投稿を行うプログラム) による投稿の可能性がある．また，1 つの投稿の適合性を判定すると，その他の同じ投稿者による同じ内容の全ての投稿が判定した適合性と同じになるため削除した．

5.2.5　単語の処理

各ノード中の文書の形態素解析には MeCab (注6) を利用した．1 文字の単語，まし，ない，です，ます，RT を禁止語として除去した．

5.2.6　評 価 指 標

本研究の目的は適合フィードバックを用いて文書を順付けることである．検索手法の評価を行うため，評価指標として normalized Discounted Cumulative Gain(nDCG) を用いる．

nDCG は多値適合性に基づく評価指標であり，検索順位を考慮した性能を比較できる．順位付けを k 位まで行う場合の nDCG は DCG@k [15] を理想的な順位で適合文書が並んだと仮定した DCG'@k で正規化した指標であり，DCG@k は以下の式で定義される．

$$\mathrm{DCG@}k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{8}$$

ただし，rel_i は i 番目の検索結果の関連度であり，今回の実験においては i 位の文書が適合文書の場合は $rel_i = 1$，不適合文書の場合は $rel_i = 0$ とした．

5.3　実　験　1

5.1 節で述べた実験 1 で行う実験内容とその結果，考察について述べる．実験 1 ではまず，次の 4 通りの手法を比較する．

(1) Baseline: ツイート取得時の Twitter の検索機能
(2) RF: コンテンツクエリを修正する適合フィードバック
(3) Context: Baseline とコンテキストクエリの
コサイン類似度による再ランキング
(4) CRFG(0): 式 (6) 中の α を 0 とさせた場合

4 通りの手法間の関係を表 3 に示す．Baseline は Twitter の現在の検索機能を反映した検索結果であり，取得した日時に近い順にツイートがランキングされている．投稿者のプロファイルを表すクエリを追加することによる影響を調べるために Context を用意する．Context は Baseline の検索結果に対しコンテキストクエリとのコサイン類似度を計算することで再ランキングした検索結果である．CRFG と同様の条件にするため，Baseline, Context ともに上位 10 件のツイートを除外した 90 件の検索結果を用いる．RF はコンテンツの文書ベクトルのみを用いてコンテンツクエリを修正する従来の適合フィードバックである．CRFG(α) の α は式 (6) 中の α を表す．CRFG(0)

(注5)：情報を拡散させるため他のユーザのツイートを再投稿したツイート

(注6)：https://code.google.com/p/mecab

表 3　手法間の関係図 (実験 1)

	フィードバックなし	フィードバックあり
コンテンツクエリによる検索	(1)Baseline	(2) RF
コンテキストクエリによる検索	(3)Context	(4) CRFG(0)

表 4　nDCG による評価結果 (実験 1)

k	Baseline	Context	RF	CRFG(0)
5	0.311	0.246	**0.377**	0.356
10	0.332	0.245	**0.336**	0.325
20	0.332	0.292	0.362	**0.364**
30	0.379	0.336	**0.419**	0.409
k	CRFG(0.25)	CRFG(0.5)	CRFG(0.75)	CRFG(1.0)
5	0.355	0.340	0.351	**0.380**
10	0.335	0.335	0.361	**0.410**
20	0.367	0.385	**0.416**	0.413
30	0.421	0.426	**0.440**	0.433

は $\alpha = 0$ の場合である．$\alpha = 0$ のときはコンテキストの割合が 0 であり，RF と同様にコンテンツの文書ベクトルのみを用いる．CRFG(0) では，修正するクエリがコンテンツクエリでなくコンテキストクエリである点で RF と異なる．この実験では，フィードバックを与えた場合と与えない場合の比較，コンテンツクエリのみを用いた場合とコンテキストクエリを用いた場合を比較する．

次に，仮説 1 を実証するために，CRFG(α) の α を $0.25, 0.5, 0.75, 1.0$ の 4 段階に変化させた場合の精度の変化を調べる．α の値を 5 段階に変化させることで，コンテキストの単語の重みの割合を変化させ，精度を比較する．

式 (7) に基づき，CRFG(α) では，コンテンツクエリではなく，コンテキストクエリを修正する．CRFG(0.25)〜CRFG(1.0) では，コンテンツの単語の重みに対するコンテキストの単語の重みの割合を上昇させる．CRFG(1.0) はツイート本文中の単語を全く考慮せず，コンテキストの単語のみを用いて文書ベクトルを生成する．実験 1 では，CRFG(α) において，コンテキストの単語を伝播させる際の伝播率をエッジのラベルの種類に応じて，$d(\text{post}) = 0.8$, $d(\text{follow}) = 0.5$, $d(\text{nearby-post}) = 0.2$, $d(\text{connect}) = 0.2$ のように設定して行った．ただし，今回の実験においては $d(\text{follow})$ をそのまま使用するのではなく，フォロワーの数で割った $d'(\text{follow})$ を用いる．$d'(\text{follow})$ を用いる理由は，フォロワー数で割らなかった場合，明らかにその検索結果の文書ベクトルがフォロワーの数に依存して変動するためである．例えば，フォロワーが 1000 人の場合を考えると，その検索結果の単語の重みが他のノードに比べ，フォロワーの自己紹介文に大きく依存して決定されてしまう．

実験の結果の nDCG@k を表 4 に示す．ただし k は検索結果の上位 k 件に対して評価値を計算したことを表す．実験の結果，Context，CRFG(0) が RF に比べて精度が低いことからコンテンツに対してコンテキストクエリを用いても，検索性能が向上しないことが確かめられた．一方，コンテキストを文書ベクトルに加える CRFG を用いた場合，上位 5, 10 件では CRFG(1.0) が，上位 20, 30 件において全ての CRFG において

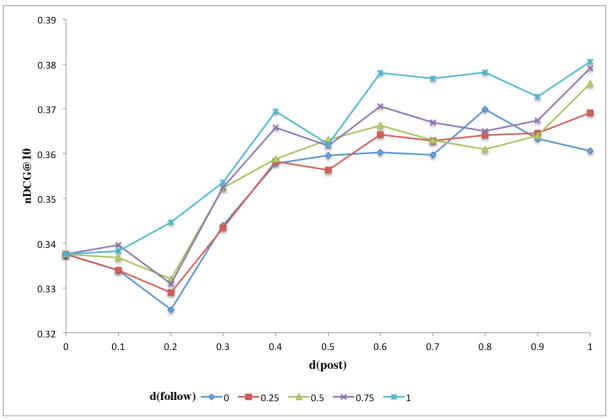

図 5　$\alpha = 0.75$ のもとで伝播率を変化させたときの nDCG@10

既存の RF を上回った．ランダム化 TukeyHSD テスト [16](注7) ($\alpha = 0.05$) によれば，Context と CRFG(0.75)，Context と CRFG(1.0) の差は統計的有意であり，他の手法間においては統計的有意差は認められなかった．

5.4 実　験　2

5.1 節で述べた実験 2 で行う実験内容とその結果について述べる．実験 2 の評価実験ではノード間で単語の重みを伝播させる際の伝播率を設定する．仮説 2 に基づくと検索結果ノードから離れたノードの単語をそのまま伝播させた場合，文書ベクトルにノイズが多く含まれる．伝播率を変化させた場合に最適なパラメータが存在することの検証が実験 2 の目的である．

実験 1 により，コンテキストを使った場合，上位 5, 10 件において $\alpha = 1.0$ の場合，上位 20, 30 件において $\alpha = 0.75$ の場合が最も高い精度を示すことが判明した．これは，伝播率を適当に設定した $d(\text{post}) = 0.8, d(\text{follow}) = 0.5, d(\text{nearby-post}) = d(\text{connect}) = 0.2$ の場合の結果である．

$\alpha = 1.0$ の場合はコンテンツの情報を全く使わないため，実験 2 では，$\alpha = 0.75$ のもとで，各伝播率のパラメータを変化させることで，上位 10,30 件それぞれの場合における最適なパラメータの存在について議論する．縦軸を nDCG，横軸を $d(\text{post})$，5 本の線を $d(\text{follow})$ を 0 から 1.0 まで 0.25 ずつ変化させた値とする．パラメータの数を減らすため，実験 1 と同様に $d(\text{nearby-post}) = d(\text{connect}) = 0.2$ として実験を行った．

nDCG@10 のグラフを図 5，nDCG@30 のグラフを図 6 に示す．図 5，図 6 のグラフから，$\alpha = 0.75$ のもとで，$d(\text{post})$ を 0.1 ずつ変化させると，nDCG@10 においては $d(\text{post}) = d(\text{follow}) = 1.0$ の場合に，最も高い精度を示した．一方，nDCG@30 では $d(\text{post}) = 1.0, d(\text{follow}) = 0.5$ の場合に，最も高い精度を示した．

$d(\text{follower})$ の値による精度の変化に比べ，横軸の $d(\text{post})$ の値による精度の変化が大きいことから，検索結果に近いノードが精度に与える影響が高いと考えられる．この理由として，user ノード以外のノードの単語を伝播させる際に user ノードを通ることが考えられる．また，$d(\text{post})$ と比べて $d(\text{follow})$ に高い伝播率を設定した方が高い精度を示したことから，検索結

(注7)：http://www.f.waseda.jp/tetsuya/tools.html

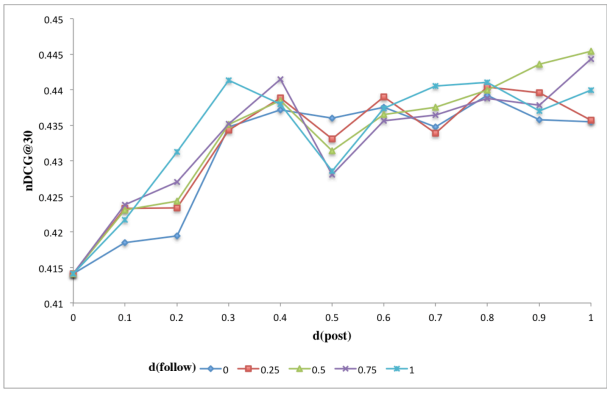

図 6　$\alpha = 0.75$ のもとでパラメータを変化させたときの nDCG@30

果に近い情報ほど投稿者のプロファイルの推定に有用であるという仮説 2 が実証されたと結論づけられる．

5.5 考　　察

5.3，5.4 節の結果をもとに，今後の課題について述べる．

本研究では，SNS の投稿は文章量が少ないという問題を解決するために，コンテキストの単語をエッジに応じて重みを減衰させながら文書ベクトルに加えた．実験の結果，上位 5, 10, 20, 30 件の nDCG においてコンテキストの単語を用いる CRFG の方が，既存の適合フィードバックよりも高い精度を示した．しかし，コンテキストの単語を用いる問題点として，コンテキスト中の膨大な語彙の中で，検索者の意図を表すキーワードはわずかであり，大半のキーワードがノイズとして作用することが考えられる．検索者の意図を表すキーワードに高い重みを与える方法として次の 3 点が考えられる．

1:　予め機械学習により，コンテキストクエリの周辺語彙を学習させ，周辺語彙にも高い重みを与える

2:　検索者が適合性判定をする直前に見たノード中の単語に高い重みを与える

3:　フィードバックの粒度を細かくする．例) ノード，単語

インタラクションの観点から考えると，手法 1 を用いた場合，システムにフィードバックを与える必要がなくなり，検索者の負担を減らすことが可能になる．一方，宮西ら [7] が文書に対して選択を行ったように単語を指定することで CRFG の精度が上昇することも期待される．検索者に負担を与えるシステムは好ましくないため，インタラクションを減らし，かつ良い精度の手法を提案することが今後の課題である．

6. おわりに

本論文では，コンテンツの語彙が少ない SNS において検索性能を上昇させるために，コンテキスト中の語彙を，伝播させる際に減衰させながら文書ベクトルに加える適合フィードバック (CRFG) を提案した．

我々は次の 2 つの仮説に基づき CRFG を定式化した：(1) 検索結果に関連付けられたコンテキスト中には発信者のプロファイルを表す情報が含まれている，(2) 検索結果から張られたグラフ上で検索結果ノードとの距離が離れるほど，発信者のプロファイルとは無関係なノイズが増大する

また，本研究では Twitter から得られた投稿者のツイートとその前後のツイート，投稿者，フォロワーの自己紹介文のデータを用いて評価実験を行い，コンテキストを使った適合フィードバックが従来の適合フィードバックを上回ることを示すとともに，2 つの仮説の妥当性を示した．

今後は，クエリ数を増やし，単語の重みの計算方法，適切な伝播率の設定方法について検討する予定である．

文　　献

[1] Joseph John Rocchio. Relevance feedback in information retrieval. *In The Smart system-experiments in automatic document processing*, 1971.

[2] 奥谷貴史，山名早人. メンション情報を利用した twitter ユーザープロフィール推定. **日本データベース学会和文論文誌**, 13(1):1–6, 2014.

[3] J-L Lu, M. P. Kato, T. Yamamoto, and K/ Tanaka. Entity identification on microblogs by CRF model with adaptive dependency. *The Institute of Electronics, Information and Communication Engineers Transactions*, 99-D(9):2295–2305, 2016.

[4] 上里和也，浅井洋樹，奥野峻弥，山名早人. Twitter ユーザを対象とした属性推定の精度向上-周辺ユーザの属性補完を利用して-. In **第 7 回データ工学と情報マネジメントに関するフォーラム**, 2015.

[5] 上里和也，浅井洋樹，山名早人. Personalized pagerank を利用した網羅的 Twitter ユーザ属性推定. In **第 8 回データ工学と情報マネジメントに関するフォーラム**, 2016.

[6] 池田和史，服部元，松本一則，小野智弘，東野輝夫ほか. マーケット分析のための twitter 投稿者プロフィール推定手法. **情報処理学会論文誌コンシューマ・デバイス & システム** *(CDS)*, 2(1):82–93, 2012.

[7] 宮西大樹，関和広，上原邦昭. マイクロブログ文書の選択による適合フィードバックを用いた疑似適合フィードバックの検索性能改善. **情報処理学会論文誌**, 55(5):1585–1594, 2014.

[8] Stewart Whiting, Iraklis A Klampanos, and Joemon M Jose. Temporal pseudo-relevance feedback in microblog retrieval. In *ECIR*, pages 522–526, 2012.

[9] 伊川洋平，村上明子. Twitter におけるイベントモニタリングのためのノイズ除去. In **第 7 回データ工学と情報マネジメントに関するフォーラム**, 2015.

[10] Harry Halpin and Victor Lavrenko. Relevance feedback between web search and the semantic web. In *IJCAI*, pages 2250–2255, 2011.

[11] Daniel M Herzig and Thanh Tran. Heterogeneous web data search using relevance-based on the fly data integration. In *WWW*, pages 141–150, 2012.

[12] Saeedeh Shekarpour, Sören Auer, Axel-Cyrille Ngonga Ngomo, Daniel Gerber, Sebastian Hellmann, and Claus Stadler. Generating sparql queries using templates. *An International Journal Web Intelligence and Agent Systems*, 11(3):283–295, 2013.

[13] Yu Su, Shengqi Yang, Huan Sun, Mudhakar Srivatsa, Sue Kase, Michelle Vanni, and Xifeng Yan. Exploiting relevance feedback in knowledge graph search. In *KDD*, pages 1135–1144, 2015.

[14] Kamran Massoudi, Manos Tsagkias, Maarten De Rijke, and Wouter Weerkamp. Incorporating query expansion and quality indicators in searching microblog posts. In *ECIR*, pages 362–367, 2011.

[15] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM TOIS*, 20(4):422–446, 2002.

[16] Benjamin A Carterette. Multiple testing in statistical analysis of systems-based information retrieval experiments. *ACM TOIS*, 30(1):4, 2012.