一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
NLC2018-7 (2018-07)

トリビア文抽出のためのトリビア度合いの推定

新名　和也†　　嶋田　和孝††

† 九州工業大学大学院 情報工学府 先端情報工学専攻 〒 820-8052 福岡県飯塚市川津 680-4
†† 九州工業大学 大学院情報工学研究院 知能情報工学研究系 〒 820-8052 福岡県飯塚市川津 680-4
E-mail: †{k_niina,shimada}@pluto.ai.kyutech.ac.jp

あらまし　近年対話システムに対して注目が集まっている．特に対話自体を目的とした非タスク指向型に関する研究は盛んに行われており，ニューラルネットワークや強化学習を用いた手法が一定の成果を挙げている．しかし，これらの手法は「はい」や「そうですね」のような文を生成しやすい傾向があるため，ユーザが対話に飽きる危険性がある．よって，より良い対話システムを構築するためには，ユーザを対話に引き込むような文を生成する必要がある．本研究では，ユーザを対話に引き込むような文として，トリビアを含んだ文に注目する．トリビアは人の興味を引く性質があるため，対話システムに用いることでユーザを対話に引き込む効果が期待できる．本研究では，このようなトリビアを含んだ文を獲得するために，Web から収集した教師付きデータを用いてトリビア度合いの推定を行う．
キーワード　トリビア，トリビア度合い推定，機械学習，ランキング学習

Estimating Trivia Score for Trivia Sentence Extraction

Kazuya NIINA† and Kazutaka SHIMADA††

† Graduate School of Computer Science and System Engineering, Kyushu Institute of Technology
680-4, Kawazu, Iizuka, Fukuoka, 820-8502 Japan
†† Department of Artificial Intelligence, Kyushu Institute of Technology
680-4, Kawazu, Iizuka, Fukuoka, 820-8502 Japan
E-mail: †{k_niina,shimada}@pluto.ai.kyutech.ac.jp

Abstract　Dialogue systems have been increasingly important these days. In particular, non-task-oriented dialogue systems have been studied because of the success of neural network approaches such as seq2seq models. However, these models tend to generate simple responses such as "yes" and "ok." To construct a dialogue system that holds user's attention continuously, we need to generate utterances that capture the interest of the user. In this paper, we propose a method to extract trivia sentences for the purpose. Trivia information perhaps adds a surprise to users. Therefore, capturing trivia information is beneficial for dialogue systems. We estimate a trivia score of a sentence by using machine learning approaches. We obtained 0.81 on the $nDCG$@10 score by a ranking method.
Key words　Trivia, Estimating Trivia Score, Machine Learning, Learn to Rank

1. ま え が き

近年，Apple の Siri や Google アシスタントなど，対話システムに対して注目が集まっている．対話システムは，ある特定のタスクの遂行を目的としたタスク指向型と，対話自体を目的とした非タスク指向型の 2 つに大きく分類される．このうち，非タスク指向型については近年盛んに研究が行われており，Sequence-to-Sequence (Seq2Seq) モデルを用いた手法や強化学習を用いて対話戦略を獲得する手法が一定の成果を挙げている [1] [2]．しかし，これらの手法では汎用的な文（「はい」や「そうですね」など，文脈に依存せず使用できる文）を生成しやすい傾向があるため，ユーザが対話に飽きてしまう危険性がある．ユーザを対話に飽きさせないためには，ユーザを楽しませる文を生成して対話に引き込む必要がある．

本研究では，ユーザを楽しませる文として，トリビアを含んだ文 (トリビア文) に着目する．本研究では，トリビア文を「ある物事についての瑣末な知識であり，かつ人の興味を引くような内容を含む文」と定義する．例えば，「アメリカザリガニにサバをあげると青くなる」のような文である．このようなトリビア文は，人の興味を引くことができる他，話題展開にも利用できるため，対話を弾ませることができる．よって，トリビア文を対話システムに組み込むことで，より楽しい対話を行うこと

ができると考えられる．

これまで我々は， Wikipedia(注1) から雑学文の抽出を行い，抽出した雑学文を利用して対話する対話システムの構築を行った [3]．雑学文とは，「ある物事についての瑣末な知識やあまり知られていない知識を含む文」のことを指す．この研究では，ドメインを料理関係に限定し，「珍しさ」に着目したスコアリングを行うことで，Wikipedia からの雑学文抽出を試みた．その結果，一般的に知られていないような内容を含む文が抽出できた．しかし，一般的に知られていない内容の文全てが本研究で定義するトリビア文であるとはいえない．雑学文は一般的に知られていない内容を含むが，トリビア文はその雑学文の中でも，人の興味を引くような特徴を持っているものとしている．したがって，トリビア文を抽出するためには，「珍しさ」以外にトリビアの性質が反映されている素性を利用することで，文の「トリビア度合い」を測る必要がある．

このような背景を踏まえ，本研究では，トリビア文を獲得するためのトリビア度合いの推定を行う．具体的には，Web から収集したトリビアに関する教師付きデータから，人手で設計した素性を抽出し，回帰とランキング学習を用いてトリビア度合いの推定を行う．

2. 関 連 研 究

トリビアを獲得する研究として，Prakash ら [4] の研究がある．Prakash らは，IMDB に掲載されているトリビアを教師データとして扱い，固有表現の種類や Superlative Words などを素性に Rank SVM を用いて「トリビアらしさ」のランキングを行っている．また，他の研究として Fatma ら [5] の研究がある．Fatma らは DBpedia(注2) に存在する RDF トリプルデータを対象に，人手でラベル付けしたデータを教師データとして，CNN を用いてトリビアであるかどうかの判定を行っている．具体的には，畳み込みで得られた特徴量と人手で設計した素性を組み合わせて学習する Fusion Based CNN を利用して，トリビアであるかどうかの 2 値分類を行っている．機械学習による分類ではなく，スコアリングによってトリビアを獲得する研究として，Tsurel ら [6] の研究がある．Tsurel らは，Wikipedia の記事に付与されているカテゴリに注目し，意外なカテゴリ (例えば，「バラク・オバマ」に対する「グラミー賞受賞者」) がトリビアであるとして，カテゴリ間の類似度やカテゴリの凝集性を用いたスコアリングによって，意外なカテゴリを獲得している．

トリビア文は，意外性のある内容を含む場合が多い．意外性のある文を獲得する研究として，太田ら [7] の研究がある．太田らは，Wikipedia の各記事の文に対して，TF-IDF，単語の共起頻度，文長の 3 つの指標に基づきスコアを計算し，ルールを満たす文にはペナルティを与えることで意外性のある文の抽出を行っている．また，他の研究として杉本ら [8] の研究がある．杉本らは，Wikipedia の記事中の文に対して，雑談対話システムに利用できる文であるかどうかを，記事内における位置情報を用いて分類している．文字数や特定単語を用いたフィルタリングなどを行った後，位置情報に基づいた素性を用いて分類を行っている．

(注1) : https://ja.wikipedia.org/

(注2) : https://wiki.dbpedia.org/

表 1　データセットの例

トリビア	「へえ」数	最大「へえ」数
アフロの仏像がある	77	100
入れ歯は昔木で出来ていた	83	100
比内鶏を食べると逮捕される	55	100
ガムはトロと一緒にかむとなくなる	130	200

表 2　データセットの度数分布表

トリビア度合い	個数
0.0 - 0.1	3
0.1 - 0.2	1
0.2 - 0.3	3
0.3 - 0.4	6
0.4 - 0.5	17
0.5 - 0.6	105
0.6 - 0.7	258
0.7 - 0.8	323
0.8 - 0.9	251
0.9 - 1.0	64
合計	1031

本研究では，トリビアに関する日本語の教師付きデータセットを用いる．データセットの各トリビア文にはトリビア度合いを表す値が付随しているため，各トリビア文のトリビア度合いを回帰やランキング学習を用いて推定する．また，各トリビア文から単語の組み合わせに着目した素性を抽出し，トリビア度合いの推定に利用する．

3. データセット

本研究で利用するデータセットは，「トリビアの泉 〜素晴らしきムダ知識〜」(注3) というタイトルの Wikipedia の記事から収集する．「トリビアの泉 〜素晴らしきムダ知識〜」とは，フジテレビ系列で 2002 年から 2012 年にかけて放送された，視聴者が投稿したトリビアを紹介する番組である．この番組について書かれた Wikipedia の記事には，番組内で紹介されたトリビアが記載されているため，これをデータセットとして利用する．

データセットの例を表 1 に示す．表 1 のように，各トリビア文には「へえ」数が付随している．「へえ」数とは，番組に出演していた芸能人が，そのトリビア文に対して行った評価であり，1 人最大 20「へえ」までつけることができる．また，最大「へえ」数とは，そのトリビア文が紹介された放送回の中で取る事のできる最大の「へえ」数を表す．本研究では，トリビア文に付随する「へえ」数を最大「へえ」数で正規化した値を「トリビア度合い」として用いる．

ここで，データセットの分布について説明する．データセットの度数分布表を表 2 に示す．表 2 より，収集したデータセッ

(注3) : https://www.fujitv.co.jp/b_hp/trivia/

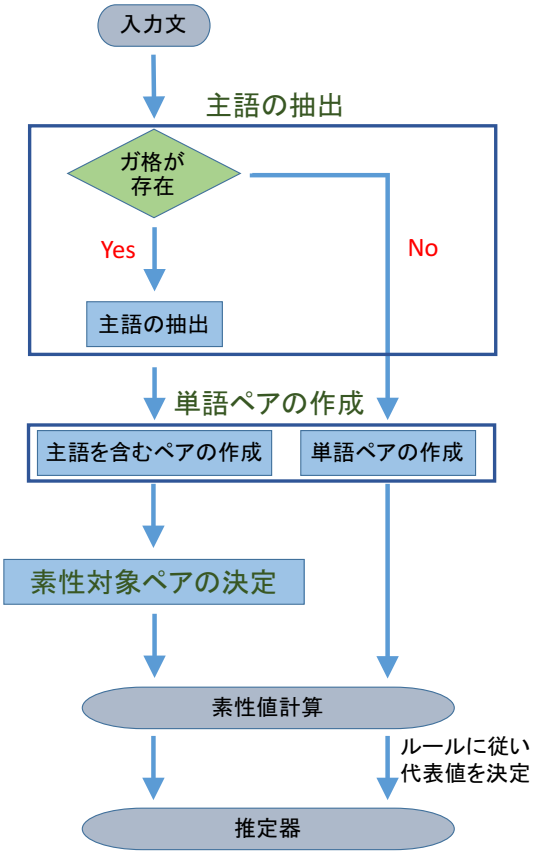

図 1　素性抽出のフローチャート

トは非常に偏っていることが分かる．これは，収集したトリビア文は，視聴者から番組に投稿されたものであり，一定のトリビア度合いを持ったものが選ばれて番組で紹介されているためであると考えられる．

4. 素　　性

本研究では，トリビア度合いの推定を行うために，各トリビア文から素性の抽出と素性値計算を行う．素性抽出のフローチャートを図 1 に示す．4.1 節では，素性値を計算するための素性の抽出について説明する．4.2 節では，計算する素性値について説明する．

4.1　素性の抽出

本研究では，素性値計算を行うために，各トリビア文に対して素性の抽出を行う．図 1 のように，素性抽出では，主語の抽出，単語ペアの作成，素性対象ペアの決定の 3 つを行う．4.1.1 節では，主語の抽出について説明する．4.1.2 節では，単語ペアの作成について説明する．4.1.3 節では，素性対象ペアの決定について説明する．

4.1.1　主語の抽出

トリビア文の持つ特徴として，「常識とのギャップ」というものがある．例えば，「アメリカザリガニにサバをあげると青くなる」というトリビア文がある．このトリビア文は，一般的に知られている「アメリカザリガニは赤い」という内容に対して，「青くなる」という常識とのギャップを持っていることで，面白いと感じるトリビア文となっている．このように，トリビア文は常識的な観点からは予想できないような内容，つまり常識とのギャップを特徴として持っている．よって，「常識とのギャップ」を表す素性を抽出することは，トリビア度合いの推定に有効であると考えられる．

この「常識とのギャップ」を表す素性を抽出するためには，主語が重要な役割を果たす．先ほどの例は，アメリカザリガニに関するトリビア文であるため，「アメリカザリガニ」という単語を中心に素性を抽出する必要がある．一方，「サバ」を中心にした場合 (つまり主語であるとした場合)，「サバは青い」という一般的な内容を表す素性を抽出してしまう可能性がある．よって，トリビア文の主語を特定することは，トリビア度合い推定に有効であると考えられる．したがって，本研究では，各トリビア文に対して主語の抽出を行う．

主語の抽出は，述語項構造解析器 ChaPAS(注4) を用いて行う．ChaPAS は Watanabe ら [9] の多言語意味役割付与モデルを日本語向けに改良したもので，ある文に出現する述語とその項構造を同定するシステムである．本研究では，ChaPAS の解析結果からガ格を持つ名詞を主語として抽出する．

4.1.2　単語ペアの作成

トリビア文に含まれる主語と，文中に含まれる単語との関係から素性を抽出することは，トリビア文の持つ「常識とのギャップ」を表すことができるため，トリビア度合いの推定に有効であると考えられる．よって，本研究では，4.1.1 節で抽出した主語と文中の単語とのペアを作成する．

ペアを作成する単語は，文中の名詞と動詞を用いる．例えば，「海底にも郵便ポストはある」という文から「郵便ポスト」が主語として抽出された場合，文中の名詞である「海底」とのペア (郵便ポスト，海底) を作成する．なお，動詞について「ある，する，いる，なる」といった語は，特徴を表す語として不適切であるため，ペア作成の際は除外する．

また，4.1.1 節で主語を抽出できなかった場合は，文中の全ての名詞と動詞から名詞-名詞ペアと名詞-動詞ペアを作成する．

4.1.3　素性対象ペアの決定

4.1.2 節で作成したペアのうち，珍しい組み合わせであるペアから素性値を計算することは，トリビア度合いの推定に有効であると考えられる．したがって，本研究では，素性として抽出するペアを決定する．

素性対象ペアの決定は，共起頻度に基づくスコア $coFreq$ を用いて行う．4.1.2 節で作成したペアに対して，$coFreq$ を式 (1) を用いて計算し，$coFreq$ の値が最小であったペアを素性として抽出する．

$$coFreq(s,w) = \frac{pair\text{-}freq(s,w)}{freq(s)} \qquad (1)$$

式 (1) の s は 4.1.1 節で抽出した主語，w は s とペアとなる単語，$pair\text{-}freq(s,w)$ はペア (s,w) の共起頻度，$freq(s)$ は s の出現頻度である．$pair\text{-}freq(s,w)$ と $freq(s)$ については，Web

(注4)：https://sites.google.com/site/yotarow/chapas

日本語 N グラム [10] の 7-gram を用いて算出する．

なお，4.1.1 節で主語を抽出できなかった場合は，4.1.2 節で作成した全てのペアを素性対象として抽出する．そして，各ペアについて次節で素性値を計算した後，ルールに基づき代表値を決定する．

4.2 素性値の計算

計算を行う素性値について説明する．本研究では，次の素性値を計算する．

- 主語の IDF
- 主語-名詞ペアの Inverse Entity Frequency (IEF)
- 主語-動詞ペアの IEF
- 主語-名詞ペアの類似度
- 主語の分散表現

4.2.1 節以降で各素性値の詳細について説明する．

4.2.1 主語の IDF

一般的によく知られている単語についてのトリビア文は，あまり知られていない単語についてのトリビア文よりもトリビア度合いが高くなると考えられる．よって，単語の IDF はトリビア度合いの推定に有効であると考えられる．したがって，本研究では，4.1.1 節で抽出した主語の IDF を素性として利用する．IDF については，Wikipedia の全記事を用いて算出する．

なお，4.1.1 節で主語が抽出できなかった場合は，文中の全ての名詞に対して IDF を算出し，その最小値を素性値として用いる．

4.2.2 主語-名詞ペアの類似度

トリビア文は意外な単語の組み合わせを含む場合が多い．例えば，「卒塔婆専用のプリンターがある」というトリビア文の「卒塔婆」と「プリンター」である．このような意外な組み合わせは，単語間の類似度が低くなると考えられる．よって，単語間の類似度はトリビア度合いの推定に有効であると考えられる．したがって，本研究では 4.1.2 節で作成された主語-名詞ペアの類似度を素性として利用する．類似度は，ペアの単語それぞれの分散表現 [11] を用いて cos 尺度で算出する．

なお，4.1.1 節で主語が抽出できなかった場合は，4.1.2 節で作成された全ての名詞-名詞ペアの類似度を計算し，その最小値を素性値として用いる．

4.2.3 主語-名詞ペアおよび主語-動詞ペアの IEF

4.2.2 節で述べたような意外な単語の組み合わせにおいて，一方の単語と，もう一方の単語が属するカテゴリとの関連性は低いということが考えられる．例えば，「卒塔婆」と「プリンター」という組み合わせにおいて，「卒塔婆」が属するカテゴリ (仏教，墓など) と「プリンター」との関連性は明らかに低いといえる．よって，一方の単語が属するカテゴリともう一方の単語との関連性を数値化することは，トリビア度合いに推定に有用であると考えられる．したがって，本研究では，主語が属するカテゴリと主語とペアになっている単語 (ペア単語) との関連性を表す IEF を素性として利用する．IEF は，Wikipedia を用いて次の手順によって算出する．

（1） 主語とタイトルが一致する Wikipedia の記事を取得
（2） 記事に付与されているカテゴリを取得
（3） カテゴリに属する Wikipedia の記事集合を取得
（4） 記事集合内におけるペア単語の IDF を算出
（5） 算出した IDF を基に IEF を式 (2) を用いて算出

$$IEF(s,w) = \frac{IDF_{C_s}(w)}{log(|C_s|+1)} \tag{2}$$

$$IDF_{C_s}(w) = log\frac{|C_s|+1}{df_{C_s}(w)+1}$$

式 (2) の s は 4.1.1 節で抽出した主語，w は s とペアとなる単語，C_s は s が属するカテゴリの記事集合，$IDF_{C_s}(w)$ は記事集合 C_s 内における w の IDF，$|C_s|$ は記事集合の要素数，$df_{C_s}(w)$ は記事集合 C_s における単語 w の文書頻度である．

なお，4.1.1 節で主語が抽出できなかった場合は，4.1.2 節で作成された全てのペアに対して IEF を算出し，その最大値を素性値として用いる．

4.2.4 主語の分散表現

単語の分散表現は，近年機械学習を用いる研究で素性としてよく利用されている．よって，本研究においても有効である可能性があるため，単語の分散表現を素性値として用いる．本研究では，4.1.1 節で抽出された主語の分散表現を素性値として利用する．

なお，4.1.1 節で主語を抽出できなかった場合は，文中の名詞から分散表現を獲得し、それを素性値として利用する．

また，次元数を固定長にするため，本研究では分散表現を獲得する単語の個数を 3 つとする．分散表現を獲得する単語が 3 つ未満の場合は，足りない分を零ベクトルでパディングする．また，分散表現を獲得する単語が 3 つより多い場合，多い分の単語は無視する．

5. 実　　験

4. 節で抽出した素性を用いて，回帰とランキング学習によるトリビア度合いの推定を行った．5.1 節では，回帰とランキング学習両方に共通する実験設定について説明する．5.2 節では，回帰を用いたトリビア度合い推定について説明する．5.3 節では，ランキング学習を用いたトリビア度合い推定について説明する．

5.1 実験設定

4.2.1 節および 4.2.3 節で用いる Wikipedia の記事については，2017 年 5 月 21 日付でウィキメディア財団より提供されているデータベース・データ(注5)から入手したものを用いた．また，カテゴリの取得については，日本語版 DBpedia(注6)から取得した．

4.2.2 節および 4.2.4 節で用いる単語の分散表現については，Word2Vec(注7) を用いて獲得した．なお，次元数は 200 とした．

(注5)：https://dumps.wikimedia.org/jawiki/
(注6)：http://ja.dbpedia.org/
(注7)：https://radimrehurek.com/gensim/models/word2vec.html

表 3　回帰を用いたトリビア度合い推定結果

手法	MAE	MSE	R2
提案手法	**0.2529**	**0.0852**	**0.0089**
ベースライン	0.2556	0.0859	0.0000

表 4　ランキング学習を用いたトリビア度合い推定結果

手法	$nDCG@5$	$nDCG@10$
Scoring	0.7220	0.7393
Regression	0.7408	0.7748
Ranknet	**0.7482**	**0.8072**

5.2　回帰を用いたトリビア度合い推定

4. 節で抽出した素性を基に，回帰を用いてトリビア度合い推定を行った．5.2.1 節では，実験方法について説明する．5.2.2 節では，トリビア度合い推定の結果について説明する．

5.2.1　実 験 方 法

本研究では，Support Vector Regression (SVR) [12] を用いて回帰を行った．SVR とは，SVM を回帰分析に応用した線形の回帰分析手法であり，カーネルトリックを用いることで非線形回帰モデルを作成することができる．本研究では，カーネルに RBF カーネルを用いた SVR で回帰を行った．なお，C の値は 1，γ の値は 1/604 とした．

また，本研究で利用するデータセットは，表 2 から分かるように非常に偏りのあるデータセットとなっている．よって，データセットをそのまま利用した場合，データセット中のトリビア度合いの平均値に近い値で推定を行ってしまい，本研究の目的にそぐわない結果になる可能性がある．したがって，本研究では，データセットを等間隔に分割し，擬似的なトリビア度合いを付与することで，偏りのないデータセットを擬似的に作成した．擬似的なデータセットの作成は，次の手順で行った．

（1） データセットをトリビア度合いで降順にソート
（2） 上位のデータから擬似トリビア度合いを付与
（3） データ 100 個毎に擬似トリビア度合いを 0.1 減少
（4） (2) から (3) を繰り返し

なお，擬似トリビア度合いの初期値は 0.95 とした(注8)．

評価については，10 分割交差検定で評価を行った．評価指標については，平均絶対誤差 (MAE)，平均 2 乗誤差 (MSE)，決定係数 (R2) を用いた．また，ベースラインについては，データセット中のトリビア度合いの平均値を使って予測するモデルを用いた．

5.2.2　結　　果

回帰を用いたトリビア度合い推定の結果を表 3 に示す．表 3 より，平均値を用いて予測を行ったベースラインと比べ，MAE，MSE，R2 全てにおいて僅かながら提案手法が上回っていることが分かる．よって，本研究で抽出した素性は，トリビア度合いの推定に僅かながら有効であると考えられる．

しかしながら，推定値の分布を調査したところ，おおよそ 0.65 - 0.45 の範囲で推定していることが分かった．つまり，現在のモデルは十分な分解能を持っているとは言い難い状態であるといえる．したがって，より高精度に推定を行うためには，新たな素性の抽出やデータセット調整方法の検討を行い，トリビア度合いの高い文 (0.85 - 0.95) や逆に低い文 (0.45 以下) などをより正確に推定できるモデルの構築が必要である．

5.3　ランキング学習を用いたトリビア度合い推定

4. 節で抽出した素性を基に，ランキング学習を用いてトリビア度合い推定を行った．ランキング学習を用いた理由は，将来的な応用先として対話システムを想定しているためである．対話システムで利用したいトリビア文は，トリビア度合いの高い文のみであるため，トリビア度合いでランキングした場合の上位の文のみを用いることになる．したがって，データセットのランキングを学習するランキング学習は，本研究の目的に対して適切であるといえる．

5.3.1 節では，実験方法について説明する．5.3.2 節では，トリビア度合い推定の結果について説明する．

5.3.1　実 験 方 法

本研究では，ランキング学習に Ranknet [13] を用いた．Ranknet とは，pairwise 手法を用いてランキング学習を行うニューラルネットワークのことである．本研究では，Ranknet のニューラルネットワークモデルに多層パーセプトロンを用いた．なお，隠れ層の数は 2，隠れ層の次元数は 1024 とした．また，データセットをランダムに分割し，90% を訓練データに，10% をテストデータとして用いた．

評価指標については $nDCG@k$ を用いた．なお，k の値は $k = 5, 10$ とした．また，ベースラインについては，[3] のスコアリングに基づくランキング (Scoring) と，回帰で予測した値に基づくランキング (Regression) を用いた，

5.3.2　結　　果

ランキング学習を用いたトリビア度合い推定の結果を表 4 に示す．表 4 より，$k = 5$, $k = 10$ 共に Ranknet が最も良い結果となっていることが分かる．よって，本研究で抽出した素性が，ランキング学習を用いたトリビア度合いの推定に有効であったことが分かる．また，Regression の結果が Scoring を上回っていることも分かる．この結果と 5.2.2 節の結果と合わせると，回帰に基づくトリビア度合いの推定について，トリビア度合いそのものの推定精度は低いが，トリビア度合いの大小関係については比較的高精度に推定できていることが分かる．

この結果を踏まえ，ランキング学習によって推定されたランキングの調査を行った．推定されたランキングの上位 n 件と下位 n 件それぞれについて各素性値の平均値を計算し，データセット全体の平均値との比較を行った．なお，n の値は $n = 5, 10$ とした．その結果，IEF (主語-名詞) と 4.2.2 節の類似度においてある特徴が見られた．IEF(主語-名詞) および類似度に関する調査結果を表 5 に示す．なお，表 5 の括弧内の数値は，真のランキングにおける平均値を表す．IEF は意外な単語の組み合わせほど値が高くなり，類似度は意外な単語の組み合わせほ

(注8)：つまり，上位 100 個のデータに 0.95 という擬似的なトリビア度合いを割り振り，次の 100 個には 0.85 を割り振り・・・というようにしてデータセットを作成した．

表 5　類似度および IEF (主語-名詞) の平均値

範囲	IEF (主語-名詞)	類似度
推定ランキング上位 5 件	0.6156 (0.7293)	0.2843 (0.2460)
推定ランキング上位 10 件	0.5742 (0.6432)	0.3609 (0.2994)
推定ランキング下位 5 件	0.8456 (0.6000)	0.2400 (0.35634)
推定ランキング下位 10 件	0.6563 (0.5554)	0.2634 (0.3238)
テストデータ全体	0.3340	0.6078

ど値が低くなる．つまり，ランキング上位の文であるほど IEF は高い値に，類似度は低い値になっていることが望ましい．表 5 の真のランキング (括弧内の数値) は実際にそのような結果となっているため，本研究で提案した IEF と類似度は，トリビア度合いの推定に適切な素性値であると考えられる．

しかし，表 5 より，推定ランキングにおいて，ランキング上位の平均値は全体の平均値と同程度かそれよりも低く，ランキング下位の平均値は全体の平均値よりも高くなっていることが分かる．IEF は意外な組み合わせであるほど高い値を示すため，「意外な組み合わせを含まないトリビア文はトリビア度合いが高い」と誤って学習していることがわかる．また，真のランキングにおける IEF (主語-名詞) の平均値は，上位のトリビア文ほどその値が高くなっている．よって，他の素性値の影響により，IEF の特徴をうまく捉えることができなかったため，IEF (主語-名詞) について誤った学習を行ったと考えられる．

類似度においても表 5 より，IEF (主語-名詞) と同様のことがいえる．つまり，真のランキングにおいて，上位のトリビア文ほど類似度の平均値が低くなるという傾向が見られるが，推定されたランキングではその傾向が見られない．よって，類似度についても IEF (主語-名詞) と同様に，他の素性値の影響により，類似度の特徴をうまく捉えることができなかったため，類似度について誤った学習を行ったと考えられる．

これらの調査結果から，より高精度に推定を行うためには，最適な素性値の組み合わせの調査や新たな素性値の計算を行うことで，トリビア度合いの大小関係をより正確に推定できるモデルの構築が必要である．

また，本研究では入力文にトリビア文を用いているが，Web 上からトリビア文を抽出する際には，トリビア文ではない一般的な文も入力文として与えることになる．よって，一般的な文を入力した時，正しくトリビア度合いを見積もることができるのか (つまり，トリビア度合いが低いと判定できるのか) について調査を行う必要がある．

6. むすび

本研究では，トリビア抽出のために，Web から収集した教師付きデータを用いたトリビア度合い推定を行った．各トリビアから IDF や IEF といった素性値を計算し，回帰とランキング学習を用いてトリビア度合いの推定を行った．実験の結果，回帰に基づくトリビア度合い推定については，平均値を用いて推定した場合と比べ僅かに良い結果となり，トリビア度合いの大小関係について比較的高精度に推定することができた．また，ランキング学習に基づくトリビア度合い推定については，ベースラインよりも比較的高い精度でトリビア度合いを推定することができた．さらに，推定結果を調査すると，IEF (主語-名詞) と 4.2.2 節の類似度において，他の素性値の影響により，それぞれの特徴をうまく捉えることができていないことが分かった．

今後の課題として，各素性値の有効性の検証や新たな素性値の設計，既存の素性値の見直し，一般的な文に対するトリビア度合いの調査がある．また，実際に抽出したトリビア文を利用する対話システムの構築も今後の課題として挙げられる．具体的には，トリビア文を出力するタイミングの制御方法の検討や，出力するトリビア文の選択方法の検討，それらに基づいた [3] の対話システムの改良などが挙げられる，

文　献

[1] Oriol Vinyals and Quoc Le. A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015.

[2] Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing,D16-112*, pp. 1192–1202, Valencia, Spain, November 2016.

[3] 新名和也, 嶋田和孝. シズルワードから想起される料理の雑学を話す対話システム. 電子情報通信学会, 言語理解とコミュニケーション研究会 (NLC), 第 4 回自然言語処理シンポジウム, 第 117 巻, pp. 77–82, 2017.

[4] Abhay Prakash, Manoj Kumar Chinnakotla, Dhaval Patel, and Puneet Garg. Did you know?—mining interesting trivia for entities from wikipedia. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, pp. 3164–3170, 2015.

[5] Nausheen Fatma, Manoj K Chinnakotla, and Manish Shrivastava. The unusual suspects: Deep learning based mining of interesting entity trivia from knowledge graphs. In *Thirty-First AAAI Conference on Artificial Intelligence*, pp. 1107–1113, 2017.

[6] David Tsurel, Dan Pelleg, Ido Guy, and Dafna Shahaf. Fun facts: Automatic trivia fact extraction from wikipedia. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pp. 345–354. ACM, 2017.

[7] 太田知宏, 鳥海不二夫, 石井健一郎. 発話生成を目的とした wikipedia からの文抽出. 人工知能学会第 23 回全国大会, 2G1-NFC5-11, 2009.

[8] 杉本俊, 植木拓, 林宏幸, Nichols Eric, 中野幹生. Wikipedia からの特定ドメインの雑談対話システムのための発話候補文集合の獲得. 人工知能学会第 31 回全国大会, 3A1-3, 2017.

[9] Yotaro Watanabe, Masayuki Asahara, and Yuji Matsumoto. A structured model for joint learning of argument roles and predicate senses. In *Proceedings of the ACL 2010 Conference Short Papers*, pp. 98–102. Association for Computational Linguistics, 2010.

[10] 工藤拓, 賀沢秀人. Web 日本語 N グラム第 1 版. 言語資源協会発行, 2007.

[11] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, Vol. abs/1301.3781, pp. 1–12, 2013.

[12] Harris Drucker, Christopher JC Burges, Linda Kaufman, Alex J Smola, and Vladimir Vapnik. Support vector regression machines. In *Advances in neural information processing systems*, pp. 155–161, 1997.

[13] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, pp. 89–96. ACM, 2005.