

# SNS を用いた視覚的日記自動生成システム 「Feeling Diary」

畑中 衛<sup>1</sup> 川瀬 智久<sup>1</sup> 菊地 慎之介<sup>1</sup> 三品 雅揮<sup>1</sup> 三井 健史<sup>2</sup> 濱川 礼<sup>1</sup>

概要：本論文では、SNS の「Twitter」に投稿された内容から、視覚的にわかる日記を自動で生成するシステム「Feeling Diary」について述べる。

Twitter への投稿から自然言語処理を用いて 1 日の特徴語及び感情情報を推定し、日々の出来事を視覚的に記録していくライフログシステム「Feeling Diary」を開発した。「Feeling Diary」は、ユーザの Twitter への投稿から自然言語処理を用いて特徴語の抽出と感情クラス分類を行い、3D モデルと 4 色の感情照明に変換して配置することで、直感的なライフログシステムを実現する。

MAMORU HATANAKA<sup>1</sup> TOMOHISA KAWASE<sup>1</sup> SHINNOSUKE KIKUCHI<sup>1</sup>  
MASAKI MISHINA<sup>1</sup> TAKEFUMI MITSUI<sup>2</sup> REI HAMAKAWA<sup>1</sup>

## 1. はじめに

本研究では、人が日記をつける際に、文字の代わりに 3D オブジェクトで表現することで、1 日を直感的に捉えることを可能とする日記の自動生成システム「Feeling Diary」について述べる。ここで本研究における視覚的日記とは、3D オブジェクトを配置することにより 1 日の出来事を表現した日記のことを指す。

## 2. 背景・目的

ライフログを用いてユーザが日常生活の振り返りを行うことを支援するサービスが注目を集めている。これは過去を振り返ることで、過去の自分を客観的に見つめなおすことができるといった利点があるためと考えられる。過去の出来事を記録する方法として日記が存在するが、日記などの手書き入力媒体は手間がかかるため毎日つけ続けることが困難である。

そこで、日常の出来事を気軽に投稿できる SNS に着目した。しかし、ほとんどの SNS はテキストベースで表現されており、視覚的に過去を振り返るのは困難である。「Feeling Diary」は SNS に投稿された文章を 3D オブジェクト化し表現することで、視覚的に振り返ることを可能にした。更

に、ユーザの投稿画像や投稿時の感情などのユーザ自身の情報を取得するほか、世間が関心を持った社会の出来事なども過去の記憶を想起させる情報であると考え、ニュース画像を取り入れるなど、記憶の想起に必要な情報の自動収集・自動生成という点に着目した。

今回は SNS の中でも外部に発信しやすく、日本での利用率が高い「Twitter」を利用する事とした [1]。

## 3. 関連研究

「Feeling Diary」ではユーザ自身が過去の 1 日分の部屋を見た際、その日何があったかを思い出すことを支援することを目的としている。関連研究について述べる。

### 3.1 ライフログの自動化

ライフログに関する自動化システムは多数存在している。その中でライフログに感情情報を付加した研究に Emote[2] がある。このシステムではユーザの投稿文書から「喜び」・「怒り」・「悲しみ」・「恐怖」・「驚き」・「期待」の 6 つの感情を当てはめ、カレンダーに色として出力することができる。感情の推定手法はナイーブベイズ分類器を用いており、事前に Twitter から約 3 万件の投稿データを学習データとして収集している。

そのデータをもとに Twitter と Facebook からデータを読み取り、一か月間の感情の移ろいをカレンダーで視認で

<sup>1</sup> 中京大学 工学部 情報工学科  
Chukyo University

<sup>2</sup> 中京大学 情報科学研究科  
Chukyo University

きるシステムである。図1は実際に Emote を一か月使用した後のユーザの感情の移ろいを示したものである。



図1 Emote 使用例

### 3.2 日記の作成

日記を作成するツールとして AirDiary : Bluetooth デバイス検出履歴を用いた半自動日記作成ツール [3] がある。これは Bluetooth デバイスを検出して得られる Social context による短文形式の日記を半自動で作成する UI である。日常的にユーザの周辺に存在するデバイスを用いログを周期的に観測することで、日常と非日常をとらえて検出した出来事をテンプレート形式で日記を作成する。

日常の記録を自動化する研究では、ウェアラブルデバイスとライフログを用いてブログ記事の自動作成を行う BlogWear [4] がある。このシステムではユーザが GPS やカメラ、小型 PC を装着し位置情報と周辺画像からブログの記事を自動作成する。記事はユーザが装着したカメラで撮影した風景と、撮影した時間ユーザが居た場所を掲載する。ユーザが居た場所は GPS データの緯度経度情報に対し、逆ジオエンコーディングを行い取得する。逆ジオエンコーディングとは、緯度経度情報から住所・郵便番号や施設名を取得することである。

### 3.3 「Feeling Diary」との比較

「Feeling Diary」では、ユーザが SNS へ投稿した際の感情分析を「自動的」に行い日記として生成する。ユーザが自身の日記に表示されている状態を見ることで、当時の記憶を思い出せるように支援することが目的である。Emote は一か月分のデータをまとめて振り返るものである。また、当時を思い出させるデータが感情データのみであり、記憶を想起させるのは難しいと思われる。

AirDiary は Bluetooth デバイスのログを周期的に観測することで、日常と非日常を検出し出来事を日記に作成する。しかし、AirDiary はテンプレート形式で日記を作成するため、表現方法が類似してしまうことがある。

BlogWear は小型 PC と周辺機器をウェアラブルデバイスコンピュータを身につけて情報を収集する。このため、これらの装置を身につけて生活する必要がある。

これらの研究に対し「Feeling Diary」では、当時の感情

に加え、その日の状態を 3D オブジェクトやニュース画像などの情報を付加することや 1 日ごとかつ時系列で表現することによって記憶の想起をより支援する。また、ユーザの SNS への投稿を利用することで、生活スタイルを変えることなく日常生活を送るだけで自動的に日記を生成する。これらの点が、他の研究・サービスとの差異である。

## 4. 投稿文書から 3D モデル配置への流れ

「Feeling Diary」では出力のため投稿文書を 3D モデルに変換する処理を行う。3D モデル変換への流れを示す。投稿文書から特徴語を抽出する処理を行い、抽出した特徴語を検索語として Web 上の 3D モデル配布サイトから自動で 3D モデル取得を行う。取得した 3D モデルは Unity で作成した部屋へ出力する。

図2に示す投稿文書から 3D モデル出力への具体例では、投稿文書から特徴語として「電車」・「蜘蛛」・「ベッド」を抽出し Web 上から特徴語に対応する 3D モデルを取得・配置している。

### 4.1 3D モデル取得における特徴語抽出方法

ユーザの投稿情報から特徴単語を抽出する処理には  $tf-idf$  法を用いる。 $tf-idf$  法とは、文書中の単語の重みから特徴語を抽出する方法であり  $tf$  (単語頻出度) と  $idf$  (逆文書頻度) の 2 つの指標に基づいて計算される。「Feeling Diary」では、 $tf-idf$  法を用いて投稿文章の語数が 3 語以上の名詞から特徴語となる単語を推定している。例として、文章数を  $M$  とする総文章  $D$  という投稿情報があったとする。 $D$  中の 1 文章を  $d$ 、 $d$  の中に含まれている単語を  $t$  とすると以下のような条件式が成り立つ。

$$D \in (d_1, \dots, d_M)$$

$$d = (t_1, \dots, t_n)$$

上記の条件式から、以下のようにして文章中の単語の重みである  $tf$ 、 $idf$  の値を求める。

なお、 $n_{i,j}$  は文章  $d_j$  中に存在する単語  $t_i$  の数  $|d : d \ni t_i|$  は  $t_i$  が存在した  $d$  の個数である。

$$tfidf = tf \cdot idf$$

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{j,k}}$$

$$idf_i = \log \frac{|M|}{|\{d : d \ni t_i\}|}$$

### 4.2 3D モデル取得

3D モデル取得は上記で抽出した特徴語をもとに Web 上から取得し、ファイルを展開後サーバ上に保存する。3D モデル取得には clara.io を利用する [9]。clara.io はユーザ



図 2 投稿と出力の例

が無料で 3D モデルを作成し公開できる Web サービスである。

Python3 の urllib3 ライブラリを用いて clara.io の API にアクセスし、特徴語に対応する OBJ 形式の 3D モデルをダウンロードする。モデルデータは JSON データベースで管理されており、各モデルを識別している SceneID を指定することでダウンロードする 3D モデルを決定する。ダウンロードした 3D モデルは圧縮されているため、ファイルを展開後サーバ上に保存する。

### 4.3 3D モデル配置

ユーザが 1 日のどの時間帯に投稿をしたかを示すために、壁面を時間軸とし 3D モデルを配置する。Unity へのインポートは OBJ 形式のファイルをリアルタイムにインポートが可能な UnityAsset である RuntimeObjImporter を使用した [13]。「Feeling Diary」は Web アプリケーションであり、ファイルを読み込む際に通信を行う必要がある。しかし、RuntimeObjImporter はファイル読み込みの際に通信を行わないため、ファイル読み込み方法を変更する必要があった。

## 5. 感情と色との関係



図 3 Plutchik の感情の輪

「Feeling Diary」では、心理学者 Robert Plutchik が考案した感情モデル「感情の輪」[5][6]を色彩と感情の対応付

けに用いた。また、感情カテゴリには日本人の感情の捉え方「喜怒哀楽」を元にして感情とそれに関連する色を選択した。図 4 は、「Feeling Diary」での感情と色の対応を示したものである。

喜	黄
怒	赤
哀	青
楽	緑
無分類	白

図 4 Feeling Diary の感情と色の対応図

## 6. 感情分類手法

ユーザの投稿情報から感情を抽出する処理にはナイーブベイズ分類器を用いる。ナイーブベイズ分類器とはベイズの公式を用いた確率的分類器であり、未知文章に対して事後確率が最大となるクラスに分類を行うものである。「Feeling Diary」ではこのナイーブベイズ分類器を用いて、投稿情報に含まれる感情の推定を行う。ナイーブベイズ分類器では、投稿文章を単語のリストに変換した投稿文章ベクトルを  $sentense = (word_1, \dots, word_n)$ 、文章の感情カテゴリを  $feeling \in (joy, angry, sad, pleasant)$  とした場合事後確率  $P(feeling|sentense)$  の値が最大となる感情カテゴリ  $feeling$  を投稿文章の感情カテゴリとする処理を行う。

ベイズの定理を用いて事後確率  $P(feeling|sentense)$  を求めることができるが、 $P(sentense)$  の値は  $P(feeling)$  の値にかかわらず一定であるため無視することができる。そのため、感情カテゴリの出現確率  $P(feeling)$  と感情カテゴリごとの文章ベクトル  $P(sentense|feeling)$  の積を求め、

その値が最大になる感情カテゴリを求めれば良いことになる。

ここで単語の出現確率は本来独立性は成り立たないが、文章ベクトルは多様な値をとるため事後確率  $P(feeling|sentence)$  を直接算出するのは困難である。そのため出現確率が独立であると仮定すると、以下の式が成り立つ。「Feeling Diary」では以下の式を使用した。

$$feeling = \arg \max_{feeling} P(feeling) \prod_{i=1}^n P(word_i|feeling)$$

## 7. システム構成

「Feeling Diary」は、Web アプリケーションとして開発されており、情報取得部、情報処理部、出力部の3部により構成されている(図5)。

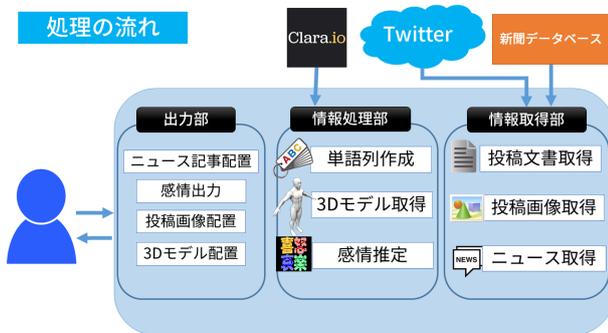


図5 システム構成図

### 7.1 情報取得部

情報取得部では、登録ユーザの SNS への投稿や、投稿当時のニュースなどの情報の集積を行う。

#### 7.1.1 投稿文書取得部

投稿文書取得部では、登録ユーザの SNS への投稿文章を集積する処理を行う。

Twitter API を利用し、任意の 1 日までの投稿文章を取得し、解析を妨げる要素を取り除いた上でデータベースに保存する。実装には Ruby の twitter ライブラリを用いた。

#### 7.1.2 投稿画像取得部

投稿画像取得部では、登録ユーザの SNS への投稿に含まれる画像などを集積し、サーバに保存する処理を行う。

Twitter API を利用し、直近 200 件の投稿から画像を抽出して、保存する。実装には Ruby の twitter ライブラリを用いた。

#### 7.1.3 ニュース取得部

ニュース取得部では、該当する 1 日を想起させることを支援するため、その日のニュース画像を取得し、保存する処理を行う。

朝日新聞社が朝日新聞デジタルにて配信する

RSS/RDF[12] を使用し、朝日新聞デジタルに掲載された最新ニュース記事に含まれる画像を取得し、保存する。

### 7.2 情報処理部

情報処理部では、取得した各種情報を出力に変換するための各種処理を行う。

#### 7.2.1 単語列作成部

単語列作成部では、3D モデル取得部及び感情推定部に必要な文字列の作成を行う。

文章を感情クラスに分類するための事前処理として、オープンソースの形態素解析エンジン MeCab[7] を用いて形態素解析を行い、名詞・動詞・形容詞を抽出する。

また、3D モデル取得部にて必要な単語列においては、2 文字以上の名詞に限定して収集し、 $tf-idf$  法を用いて集められた単語列から特徴語を抽出し、それらを多言語オンライン辞書 Glosbe[8] の API を用いて英語変換を行う。

#### 7.2.2 3D モデル取得部

3D モデル取得部では、SNS 部で作成された特徴語をもとに Web 上から 3D モデルを取得し、サーバ上に保存する。

Python3 の urllib3 ライブラリを用いて clara.io の API にアクセスし、特徴語に対応する OBJ 形式の 3D モデルをダウンロードする。その後取得したファイルを展開し、サーバ上に保存する。

#### 7.2.3 感情推定部

感情推定部ではナイーブベイズ分類器を用いた。学習データとして、感情関係性を Wordnet-Affect[10][11] から収集した後それらを日本語に翻訳し、喜怒哀楽の 4 つの感情に分類し直した約 8000 語のデータを使用した。

### 7.3 出力部

出力部では、取得した 3D モデル・ニュース記事・推定した感情を Unity 上で作成した部屋へ、投稿画像を回廊に出力する処理を行う。部屋は 2 週間分保存されており、部屋から部屋への移動は回廊を通して行う。

#### 7.3.1 3D モデル出力部

3D モデル出力部では、Unity 上の正方形の部屋に取得した 3D モデルを配置する。上記で述べた通り特定の壁を時間軸とし、投稿時間に合致した位置に 3D モデルを配置する。図6の例では、ベッドと電車が配置されている。ベッドは 0 時から 4 時を示すオブジェクトの前に配置されているため、0 時から 4 時の間の投稿から出力されたことがわかる。電車は 12 時から 16 時の間の投稿から出力されたことがわかる。

#### 7.3.2 ニュース出力部

ニュース出力部では、取得したニュース画像を Unity 上の部屋に配置する。朝日新聞デジタル RSS から取得した画像を板状のオブジェクトにテキストチャとして貼り付け壁に配置する。

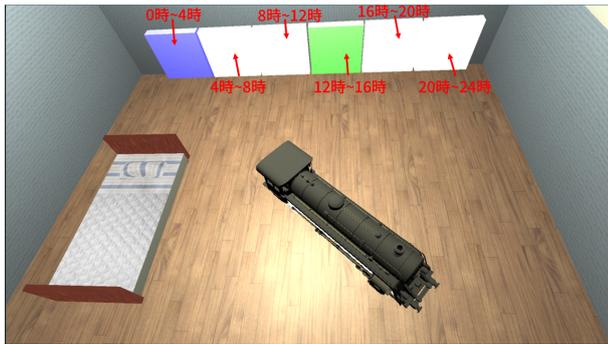


図 6 3D モデル配置例

下に示す図 7 は、ニュース画像の例である。2016 年 6 月 30 日のニュース画像であり、集団登校中の小学生の列に乗用車が突っ込むという事故（当時）が発生、全国ニュースで報じられる [14] など、社会が関心を持ったニュースである。



図 7 ニュース出力例

### 7.3.3 感情出力部

感情部では、推定した感情により色が変わる立方体のオブジェクトを部屋の壁に配置する。

感情を示すオブジェクトは 3D モデル同様特定の壁を時間軸とし、配置する。

図 8 は、本システムを利用した際 24 時間を左から 0 時～4 時、4 時～8 時というように 6 区間に分けそれぞれの区間が「怒」、「喜」、「哀」、「楽」、「喜」、「哀」と判定され表示された例である。

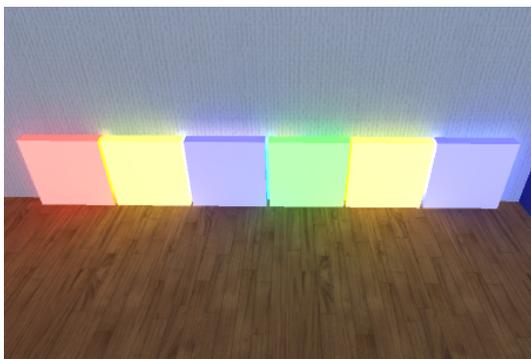


図 8 感情出力例

### 7.3.4 投稿画像出力部

投稿画像出力部では、投稿画像を Unity 上の回廊に配置する。保存された各投稿画像を額縁型 3D モデルにテキストチャとして貼り付け、時系列に沿って、回廊の壁面に配置する。取得してきた画像を板状のオブジェクトにテキストチャとして貼り付け、回廊の左右壁面に配置する。

図 9 に示す画像では、ユーザの投稿画像が回廊のドアの左右に出力されている。

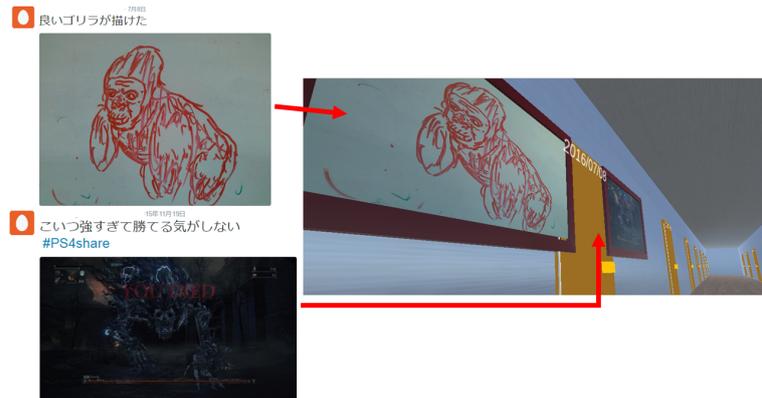


図 9 投稿画像出力例

## 8. 評価・考察

学生 8 名に対し、手書きの日記を 3 日間書いてもらい、その後「Feeling Diary」を使用してもらい、アンケートを実施した。内容は、「「Feeling Diary」により記録されたものは、自身の記憶と合致していたか」、「「Feeling Diary」を使用することで、過去の出来事を思い出すことができたか」、「日記にその日の感情変化や画像を残すことは、過去を振り返る上で必要だと感じたか」、「「Feeling Diary」と紙に書いた日記とを比較して、どちらが気軽に継続できそうだと感じたか」の 4 項目を評価してもらった。

「Feeling Diary」による出力は認識と合致していたかという質問に対しては、評価者全体の 50% を占める 4 名が「概ね合致していた」と回答したが、「どちらともいえない」「あまり合致していなかった」「ほとんど合致していなかった」の回答の合計も 50% を占めており、出力の表現方法と精度に課題があることがわかった。

過去の出来事を思い出すことができたかという質問に対しては 1 名が「出来た」、6 名が「やや出来た」と回答し、合計で評価者全体の 75% から肯定的な回答を得られた。これにより、記憶の振り返りという用途において「Feeling Diary」の有用性が示されたが、上記ユーザの認識との合致率から過去を正確に表現できておらず、振り返りの際の記憶の補完程度の働きに留まっている。

過去を振り返る際に感情変化や画像の記録は必要であるかという質問に対しては、評価者全体の 50% を占める 4 名

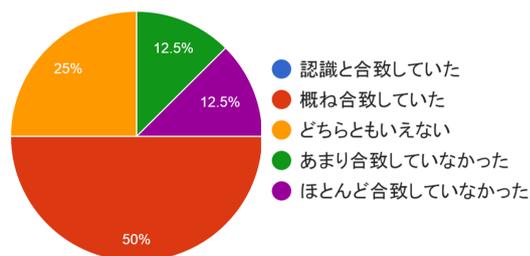


図 10 Feeling Diary の出力とユーザの認識との合致率

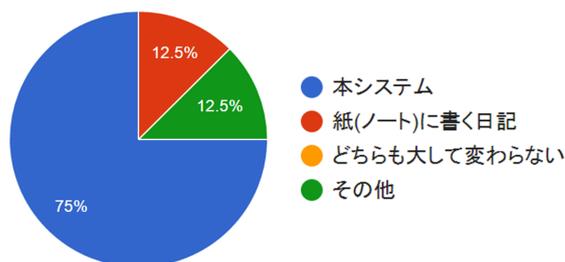


図 13 手書き日記と比較した Feeling Diary の簡易性, 継続性

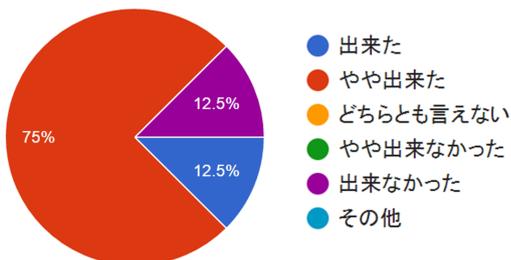


図 11 振り返り用途における Feeling Diary の有用性

が「必要だと思う」と回答し, 37.5% を占める 3 名が「必要ではないが, あると良いと思う」と回答した. これにより, 感情変化や画像の記録の存在が記憶の振り返り工程に於いて有用であることが示された.

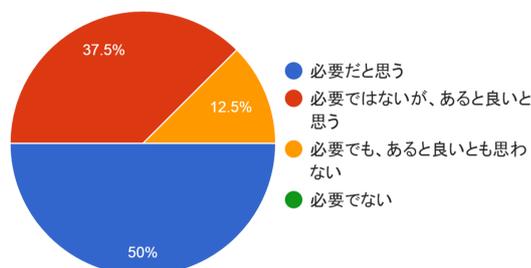


図 12 振り返り用途における感情変化や画像の記録の有用性

「Feeling Diary」と手書き日記の簡易性と継続性の比較についての質問には, 評価者全体の内 75% を占める 6 名が「本システム」(の方が優れる)と回答した. これにより「Feeling Diary」の目的の 1 つである「記憶の振り返りにおける簡易性と利便性の向上」が達成できていることが示された.

#### 参考文献

[1] ICT 総研 2015 年度 SNS 利用動向に関する調査  
<http://ictr.co.jp/report/20150729000088-2.html>  
 [2] 三井健史, 伊藤智也, 中西勇人, 濱川礼, “SNS の投稿を用いた感情記録ライフログシステム~Emote~”, 情報処理学会第 32 回エンタテインメントコンピューティング研究発表会 (EC) (2014,6)

[3] 牛越達也, 河野恭之, “AirDiary : Bluetooth デバイス検出履歴を用いた半自動日記作成ツール”, 情報処理学会第 142 回ヒューマンコンピュータインタラクション研究発表会 (HCI) (2011.3)  
 [4] 小菅徹, 吉野孝, “ライフログを用いたブログ記事自動生成システム BlogWear の開発と評価”, 情報処理学会研究報告マルチメディア通信と分散処理 (DPS) (2009.11)  
 [5] Robert Plutchik: Emotion: Theory, Research, and Experience, New York: Academic  
 [6] Robert Plutchik: “The nature of emotions,” American Scientist, Vol. 89, Iss. 4.  
 [7] MeCab: Yet Another Part-of-Speech and Morphological Analyzer  
<http://taku910.github.io/mecab/>  
 [8] Glosbe-多言語オンライン辞書  
<https://ja.glosbe.com/>  
 [9] Clara.io is a full-featured cloud-based 3D modeling, animation and rendering software tool that runs in web browser.  
<https://clara.io/>  
 [10] Bernardo Magnini and Gabriela Cavaglia'. Integrating Subject Field Codes into WordNet. In Gavrilidou M., Crayannis G., Markantonatu S., Piperidis S. and Stainhaouer G. (Eds.) Proceedings of LREC-2000, Second International Conference on Language Resources and Evaluation, pp. 1413-1418 Athens, Greece, 31 May - 2 June, 2000.  
 [11] Carlo Strapparava and Alessandro Valitutti. WordNet-Affect: an affective extension of WordNet. In Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004), Lisbon, May 2004, pp. 1083-1086.  
 [12] 朝日新聞社 RSS/RDF サービス  
<http://www.asahi.com/information/service/rss.html>  
 [13] <https://www.assetstore.unity3d.com/jp/#!/content/49547>  
 [14] <http://www.asahi.com/articles/ASJ6Z2TRJJ6ZOHGB001.html>