

# クロスモーダル解析を用いたレコメンダーシステム ーレコメンダーエンジンの実装と導入ー

DU YAN

学籍番号：201620715 E-mail: s1620715@u.tsukuba.ac.jp

指導教員：蔡 東生 課題担当教員：蔡 東生

**あらまし** 本プロジェクトは、筑波大学大学院「高度 IT 人材育成のための実践的ソフトウェア開発専修プログラム」における研究開発プロジェクトとして、筆者を含む 4 名のチームで、株式会社 IROYA を顧客として、クロスモーダル解析を用いたレコメンダーシステムの開発を行う。本プロジェクトは Web サイトでユーザーの閲覧を取得して、好みの色分析と流行ファッションを推測して、各ユーザーに関連性高いアイテムを勧める。そしてそれがもたらす効果についての実証実験を行うものを目標としている。本プロジェクトにおいて、筆者はレコメンダーエンジンの構築とデータ分析の開発を担当する。

## 1. プロジェクトの背景

### 1.1. プロジェクトの目的

本プロジェクトの顧客株式会社 IROYA は「色から好きになる」という理念を持って、ファッションアイテムのショッピングサイト IROZA<sup>[1]</sup>を運営している会社である。伝統的なロングテール販売手法と違う、ファッション世界の流行アイテムを探して、色を中心に流行できそうな商品をユーザーに提供できるために、レコメンダーシステムの開発と実証実験協力を依頼している。

### 1.2. プロジェクトチーム体制

本プロジェクトは筆者と潘振雷, PU YUE, 董哲の四人で開発を行う。プロジェクト内容は Web アプリとレコメンドエンジンに分けて、ファッションサイトと閲覧履歴データベースの構築を PU YUE と董哲が担当する。レコメンドエンジンの開発を筆者と潘振雷が担当する。

開発項目	担当者
レコメンダーエンジン	DU YAN, 潘振雷
サーバ, データベース	董哲
Web サイト, Google Analytic 利用	PU YUE

### 1.3. クロスモーダル現象と共感覚

風鈴の音を聞くと、気温が下がったわけではないのに涼しさを感じる。赤い色に温かさを感じ、青い色に涼しさを感じる。食事でも、同じ料理であっても、器の形や色、盛り付け方で食欲をそそられる場合もある。こうした経験からも分かるように、私たちの五感はそれぞれ独立したものではなく、互いに交わり、影響を与えあって様々な感覚を感じ

じ取っている。このような感覚の仕組みをクロスモーダルと呼んでいる<sup>[2]</sup>。

音に対して色彩を感じる色聴など、一つの刺激に対して複数の感覚が同時に起こる経験である。共感覚を持つ人々は共感覚者と呼ばれ、多くは先天的なものと考えられている。共感覚は新生児が普遍的に持つと考えられている感覚であるが、脳の発達に伴う脳領域分化によって、大人になると失われてしまいます。共感覚的なマッピングを一般の感覚保持者が持っている可能性は示唆されている。<sup>[3]</sup>

### 1.4. レコメンダーシステム

レコメンダーシステムは情報データフィルタリングの一種である、目的としては閲覧履歴やアンケート結果からユーザーのデータを収集するより人の興味と傾向を把握し、関連性高いアイテムをユーザーに勧めることである。

近年、レコメンダーシステムはますます活躍していて、映画、音楽、新聞、検索など、様々な分野で使われている、特に Amazon のような電子商取引にとっても不可欠な販売促進手法である。電子商取引の規模を成長しているとともに、販売されている商品の種類と数も非常に増えている。膨大な商品リストから無駄な時間を使わずにユーザーがほしいものを早めに見つけさせることはレコメンダーシステムの役割である。

代表的なレコメンダアルゴリズムはアイテムベースとユーザーベースの協調フィルタリングである。アルゴリズムの原理はアイテムとユーザー間の類似度を計算する。アイテム A と B の類似度が高いなら、A を購入したユーザーに B を勧める。もしくはユーザー X さんと Y さんの類似度が高いなら、X さん最近購入したのも Y さんに勧める。

今回のプロジェクトで、色に基づくレコメンド手法を実現するため、色聴という共感覚を利用した。音の

中でも調に注目し、調がそれぞれ持っている印象と調に対応する共感覚色を紐付ける。そうすることで色によって自分が印象付けたい性格というものがわかるようになる。

ユーザーが実際に服を購入する際に、服の色などで自分が相手に印象付けたい性格を無意識のうちに決めている。それを共感覚を使ったクロスモダリティの観点からファッション通販サイトにおけるレコメンダーシステムに応用することで今までにない形のレコメンダーシステムの基盤を構築することを目的とする。<sup>[4]</sup>

## 2. 開発内容

### 2.1. 開発全体像

本プロジェクトは Web サイト、サーバとデータベース、レコメンダーエンジン三つの部分が構成されている。システムの全体像としては、以下の図 1 のように示す：

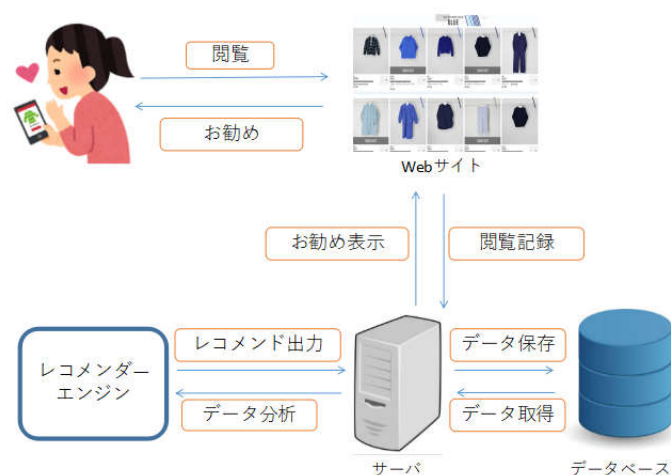


図 1. システム全体像

特に説明したいことは、レコメンダーエンジンはブラックボックスのように、サーバにレコメンド結果を提供できる。複数の推薦手法、さらに推薦結果を組み合わせる可能性も存在である。

### 2.2. 開発環境

顧客の IROYA が運営している販売サイトと同じように、Amazon のクラウドサービス AWS において今回プロジェクトの開発環境を構築した。

AWS には様々なプラットフォームが存在する。IROYA が利用されているのは Elastic Beanstalk、Web 開発に向けて専門的なプラットフォームである。Web 開発には非常に便利だが、弱いカスタマイズなので開発言語しか選べない、ソフトウェアもインストールできない。

今回のプロジェクトでレコメンダーエンジンの開発と導入するために、オープンソースとライブラリが必

要であるから、AWS の EC2 プラットフォームを選んだ。

EC2 プラットフォームにおいて Ubuntu の仮想マシンを構築してから、Web サイト、サーバとエンジンの開発を開始した。

<input type="checkbox"/>	Name	实例 ID	实例类型	可用区
<input type="checkbox"/>	http	i-0854b7b2df91ba1a1	t2.micro	us-east-2c
<input type="checkbox"/>	http2017	i-0c511e51acc998642	t2.micro	us-east-2c

図 2. AWS 上の EC2 仮想マシン

開発中、具体的に開発言語、オープンソースとツール、及びライブラリなどは以下で表示する。

開発項目	環境
レコメンダーエンジン	言語：Python ライブラリ： OpenCV scikit-learn Crab など
サーバ、 データベース	言語：PHP 環境：Apache, MySQL
Web サイト、 Google Analytic 利用	言語：HTML, JavaScript, Python

### 2.3. Web サイトとサーバ

レコメンダーシステムの効果を実証実験するために、販売サイトで顧客に商品リストを展示して、訪問された商品ページの閲覧履歴を取得する必要があるから、Web サイトを開発する。Web サイトで利用している開発言語は HTML、CSS と JavaScript である。

Web サイトを運営し、顧客の履歴を保存して、個性的なレコメンドをすることができるために、今回はサーバを構築する。サーバ側の開発は PHP である。

### 2.4. レコメンダーエンジン

サーバ側が取得した顧客の閲覧履歴データをレコメンダーエンジンに入力する。エンジンが閲覧履歴から顧客の好み色を分析し、好きになる可能性高い色とファッションを予測する。それから Web サイトのおすすめアイテム欄に提示する。

筆者が担当するレコメンダーエンジンにおいて、商品の色と販売データなど、いろんな情報を分析してからレコメンドするために、OpenCV<sup>[5]</sup>と scikit-learn<sup>[6]</sup>などのライブラリを利用している。そして、今後様々なレコメンド機能を追加する予定があるので、Python 言語でレコメンドアルゴリズムを実現するオープンソース Crab<sup>[7]</sup>も使っている。

## 3. レコメンド手法

顧客として、IROYA の要望は色を中心にユーザーの

好み商品を推定して Recommend すると考えていて、「カラーベース」の Recommend 手法を導入したが、Recommend の効果を検証するため、他の二種類 Recommend アルゴリズムも導入した、一つは「ユーザーベースフィルタリング」、もう一つは「コンテンツベースフィルタリング」である。

### 3.1. カラーベースフィルタリング

#### 3.1.1 色と調性格

先行研究<sup>[4]</sup>より、色聴判定テストを行って、共感覚者のテストデータを K-means 法を用いてクラスタリングして、各音声の調に対する共感覚色を推定できた。以下の図 3 は一部の調性格と対応している色の分布である。



図 3. 調性格に対応する色

#### 3.1.2 色分析

顧客の IROYA の販売サイトから商品の画像を取得して、画像のカラー RGB 値と重みを計算する。

RGB は Red, Green, Blue 三つの原色を混ぜて幅広い色を表示する。各原色の範囲は 0 から 255 までなので、 $256^3$  通りの色を表現できる。

RGB 値で表現している画像の色は豊富であるけど、幅広い色からメイン色を抽出しないと色分析は進めない。このプロジェクトで筆者が採用している方法は以下、

##### 1. 画像から商品の枠を抽出する。

画像処理について枠を抽出する方法が多くて、例えば watershed, GrabCut などのアルゴリズムである。でも効率とスピードを考えるうえで、今回はシンプルな閾値分割法を採用することにした。

閾値分割法は閾値を設定することによって、前景の商品と背景の色を区別する。



図 4. 商品の画像から前景を抽出

2. 画像を HSV 空間に変更し近い色を合流させて、HSV 空間の画像のカラーヒストグラムを求め
  3. 画像を RGB 空間に戻って、主要色の RGB 値を求める
  4. ヒストグラムより各色の重みを計算する
- 図 5 のように、商品の画像の主要色と重みを求めることができる。

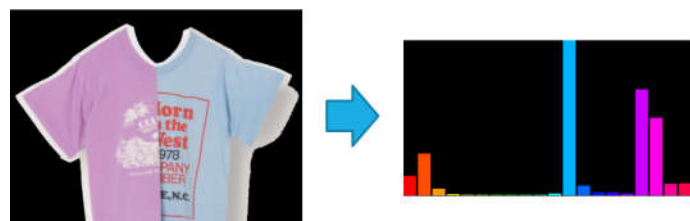


図 5. 商品の主要色を抽出

目で見て色近い画像だとしても実際に違う成分がある。メイン色抽出の安定性を考えるうえで、画像の中に細やかな色を捨てるべきだと考えているから、ピクセル少ない色を無視する。最後に取得した結果は図 6 のように表示されている。

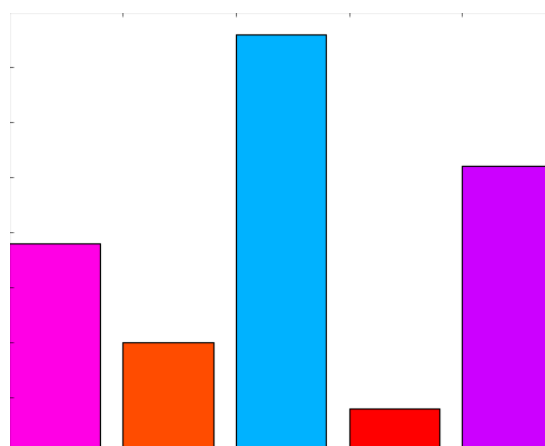


図 6. 最終のメイン色成分

上記は一枚画像のメイン色を抽出できた。実際にユーザーが複数の商品を購入するので、複数の画像に対するメイン色も抽出しなければならない。

複数の画像の場合、まずは別々に色分析して、そしてある色とその画像に占めているピクセルの比率を手

に入れる．それから複数枚の画像の色と重み率を合流して，各メイン色と重みを計算できる．

図 7 のように，三枚の画像の場合，計算して取得した色の結果を示している．

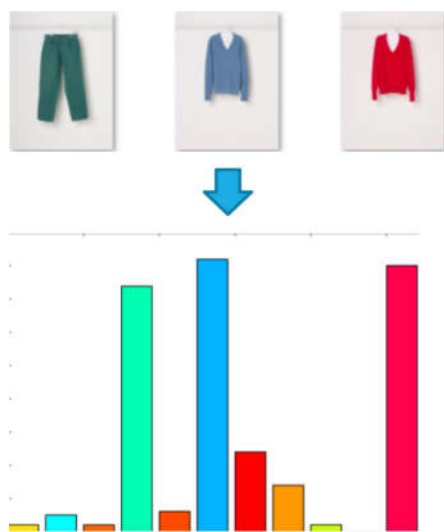


図 7. 複数の画像のメイン色成分

### 3.1.3 EMD 距離より調性格推定

EMD (Earth Mover's Distance) は，ユークリッド距離のような距離尺度の一つで，二つの分布の間の距離を測ることができ，画像処理や音声処理では比較的名な距離尺度である．類似画像検索の分野でよく用いられている．<sup>[8]</sup>

従来のヒストグラムによる色分布表現では，ヒストグラムのビンが固定であるため表現力が不十分であり，またヒストグラム間の比較尺度が人間の感覚に合わないことがあるといった欠点である．シグネチャの場合は対象に応じて要素数，ビンの代表値を変えることができ，柔軟な表現が可能である．また，EMD はほかの尺度より人間の感覚に近いことが報告されている．<sup>[9]</sup>

今回のプロジェクトで RGB 色を採用するので，色の RGB 値と重みの集合で EMD 距離を計算する．

図 8 のように，商品の色と各調性格の色の EMD 距離を計算して，最低値をもらえる調性格は，この商品の調性格だと推定する．すなわちこのユーザーがこういう調性格の色が好きだと判断できる．

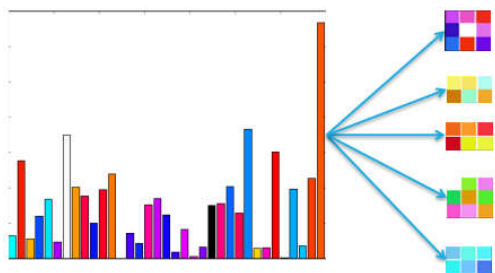


図 8. EMD 計算より調性格推定

### 3.1.4 レコメンド

ここで，購入履歴よりユーザーの調性格を推定できると考えている．続いては販売商品リストから同じ調性格の色を持つ商品を抽出して，ユーザーにレコメンドする．

ユーザーの購入履歴を更新するとともに，その調性格を変更される可能性も存在である．

## 3.2. ユーザーベースフィルタリング

### 3.2.1 マトリックス

レコメンドシステムにおいて，ユーザーとアイテムが不可欠な要素であり，すべてのデータを用いてマトリックスを作成できる．

	お菓子	靴	靴	英語本	Switch
UserA	1	0	1	0	1
UserB	0	1	1	0	1
UserC	1	1	0	0	0
UserD	0	0	0	0	1

図 9. マトリックスの様子

上図 9 のように，各ユーザーが何の商品を購入したかを表示している，購入した場合は 1，購入していない場合は 0．一般的に，そのマトリックスが疎行列である．理由は，販売商品数と比べて，一人のユーザーが購入した商品の種類は少ないである．

### 3.2.2 ユーザーベース

協調フィルタリングの一種である，目標はマトリックスに空いているものを予測する．具体的にはユーザーの購入履歴をベクトルにして，各ユーザーの類似度を計算し，お勧めするアイテムを決める手法である．ここで利用されている計算手法はコサイン距離である．

図 3 を例にすると，UserA のベクトルは (10101)，UserB は (01101)，UserC は (11000)．お互いにコサイン距離はどのように，

$$\cos(A, B) = \frac{0+0+1+0+1}{\sqrt{1^2+1^2+1^2} \cdot \sqrt{1^2+1^2+1^2}} = 0.67$$

$$\cos(A, C) = \frac{1+0+0+0+0}{\sqrt{1^2+1^2+1^2} \cdot \sqrt{1^2+1^2+1^2}} = 0.33$$

UserA と UserB の距離が 0.67，UserA と UserC の距離より 1 に近くて，すなわち比較的に UserA と UserB の購買類似度が高いと考えられる．

### 3.2.3 レコメンド

類似度高いユーザーの間に潜在的な関連性があるから，その関連性を利用してマトリックスに空いている情報を推定できる．先ほど図 3 を例にした UserA と



UserB の類似度が高くて、UserB が靴を購入したから、UserA が靴を購入する可能性もある。逆に、UserA がお菓子を購入したから、UserB もお菓子を購入かもしれない。

ここで、ユーザーベースフィルタリングを用いてレコメンドできた。

### 3.3. コンテンツベースフィルタリング

#### 3.3.1 コンテンツベース

ユーザーベースのユーザーたちの類似度を考察することと比べて、コンテンツベースはアイテムの特徴量を捉えて、例えば映画であれば、俳優、ジャンル、監督など、お気に入りアイテムの属性を抽出する。

図 10 のように、顧客の IROYA の通販サイトに、色々商品のキーワードと説明内容が載せている。今回は TF-IDF 手法を使って、商品の属性を分析してレコメンドすると考えている。



図 10. 顧客 IROYA の販売商品の説明内容

#### 3.3.2 TF-IDF

TF-IDF は文章中に含まれる単語の重要度を評価する手法の一つであり、主に情報検索やトピック分析など分野で用いられている。TF-IDF は TF と IDF の二つの指標に基づいて計算される。<sup>[10]</sup>

TF は Term Frequency, すなわち目標の単語が文章に出現した程度である。TF の評価方式は以下、

$$tf_{ij} = \frac{n_{ij}}{\sum_k n_{kj}}$$

分子は目標単語が文章に現れた回数、分母は文章になかですべての単語の出現回数である。これより単語の出現程度を評価できる。例えばある文章に「遺伝子、癌」の出現頻度が高いなら、ほぼ間違いなく医療文章だと考えられる。

日本語だけではなく、すべての言語も同じように、明確な意思を持つキーワードより、文章には「が」、「は」、「です」、「ます」などの補助用語が大量に使われるから、出現頻度も非常に高いはずである。この問題を避けるために、IDF も必要である。

IDF は Inverse Document Frequency, 逆文書頻度という。先ほどの例にしたら、医療文章だけではなく、どの文章にも「が」、「は」、「です」、「ます」などの一般語も頻繁に出現したから、逆にそれらの単語の重要度が下がると判断する。IDF の評価方式は以下、

$$idf_i = \log \frac{|D|}{|\{d : t_i \in d\}|}$$

分子は総文章数、分母は単語  $t$  を含む文章数である。

TF と IDF を掛けて、無意味な一般語を無視できるから、特定の文章にしか出現しない単語も抽出し、その単語の重要度を上げる役割を果たせる。

#### 3.3.3 レコメンド

例えばユーザーが図 4 の商品を購入したら、TF-IDF 手法でその商品の重要度高いキーワード「丈夫」、「シンプル」などを抽出する。それから同じキーワードを持つ商品を探して、ユーザーにレコメンドする。

ここで、コンテンツベースフィルタリングを用いてレコメンドできた。

## 4. 開発進捗

### 4.1. 商品データ取得

レコメンドエンジンにとって、カラーベースは商品の画像、コンテンツベースは商品のキーワードと説明内容が必要であるから、IROYA の HP から自動的にダウンロードすることにした。

具体的には、まずは IROYA から提供してもらった Google Analytics のソースコードを用いて、指摘期間のユーザーの観覧商品 URL 履歴をダウンロードする。それらの URL を用いて、Python 言語で自動的に URL のページを訪問し、画像ファイルと商品説明内容をダウンロードすることができた。

下図 11 のように、顧客の IROYA の販売サイトから何百枚の服の画像をダウンロードし、名前も取得したキーワードを用いて付けた。

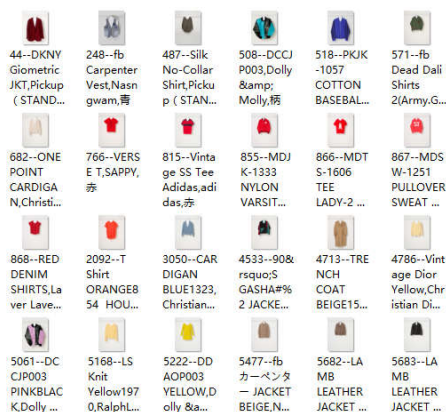


図 11. 顧客 IROYA の販売商品とキーワード内容

## 4.2. クラウドサーバーの構築

ASW の仮想マシンで、PHP と Apache を用いたサーバを構築した。

サーバの役割は、Web から入力されたユーザーの購入と評価情報を受け取って保存し、さらにエンジンを呼び出して、レコメンド商品を Web に送って、ユーザーに進めることである。

## 4.3. エンジン実装と導入

### 4.3.1 カラーベースフィルタリング

AWS の仮想マシンで OpenCV などのツールをインストールしてから、カラーベースフィルタリングのコードを実行できた。

なお、EMD 距離計算のためにオープンソースを利用した。利用されたオープンソースは <https://github.com/pdinges/python-emd> からダウンロードできる。

### 4.3.2 ユーザーベースフィルタリング

ユーザーベースフィルタリングを実現するために、既存のオープンソース Crab を利用して実現できた。Crab は Python 言語で実現できた協調フィルタリングライブラリである。

オープンソース Crab ライブラリのコードは <https://github.com/muricoca/crab> からダウンロードできる。数値計算と表示のため、numpy, scipy, matplotlib などいろんなパッケージをインストールしなければならない。

### 4.3.3 コンテンツベースフィルタリング

コンテンツベースフィルタリング部分はオープンソースを利用した。

<https://github.com/groveco/content-engine>, このオープンソースが TF-IDF を利用して、指定の短文と相似性高い他の短文を探せる。

コンテンツベースエンジンを実現するため、すべての商品の ID と説明内容をトレーニングデータとして用意する。それから指定商品の ID を入力し、その商品とキーワード近い商品を見つけてから、レコメンドする結果として出力する。

## 4.4. Web フロント設計

レコメンド効果を検証するために、テスト用の Web ページを作った。そのページを訪問して、IROYA の商品を観覧し、購入と評価することができる。

## 5. まとめと今後の予定

顧客の IROYA とミーティングして、実装方針と役割分担について意思を確認して、プロジェクトチームと顧客の関係を明確した。そしてアジャイル開発手法を使って開発を進めるを決めて、二週間に一つスプリン

トでスケジュールを計画した。最初のスプリントでは要件定義を行い、チームメンバーはそれぞれの担当範囲で調査し、勉強しながら開発環境構築などを行っている段階である。

今後の予定について、現時点では以下の通りに計画している。

11-12 月

- ・成果物の設計と実装
- ・ユーザーテストで評価

12-1 月

- ・研究報告書を作成
- ・顧客へ納品

今後はスケジュールに従って、各機能を実装して、レコメンダーシステムを完成することを目指す。そのうちに、顧客と課題担当教員ともミーティングも定期的に行い、進捗を報告する。

## 文 献

- [1] IROYA 社, 色彩特化型 E コマースサイト & アパレルセレクトショップ<<https://iroza.jp>>
- [2] 株式会社キビテク<[iot-jp.com](http://iot-jp.com)>
- [3] 蔡東生, “色聴共感覚の一貫性実験によるクロスモーダルマッピング”, 情報処理学会研究報告. グラフィクスと CAD 研究会報告/2013-CG-150(4)/pp.1-8, 2013-02
- [4] 渡邊翔, “クロスモーダル解析を用いたレコメンダーシステム”, 筑波大学, 情報学群, 情報メディア創成学類, 卒業研究論文, 2016
- [5] OpenCV Developers Team, “ABOUT | OpenCV”<<http://opencv.org/about.html>>
- [6] David Cournapeau, a free software machine learning library for the Python programming language<<http://scikit-learn.org/stable/>>
- [7] Muriçoca Labs, A Recommender Framework in Python<<http://muricoca.github.io/crab/>>
- [8] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas  
Computer Science Department, Stanford University  
Stanford, CA 94305
- [9] 井上泰助, 共感覚を用いた音楽と映像のノンバーバルマッピング, 筑波大学修士論文(2008)
- [10] TF-IDF, a numerical statistic that is intended to reflect how important a word is to adocument in a collection or corpus<<https://ja.wikipedia.org/wiki/Tf-idf>>