

自然言語処理入門と活用

NTTコミュニケーションズ株式会社

村上優樹、竹越智也

「私は母と焼肉を食べた」

「私は母と焼肉を食べた」



「私は母と焼肉を食べた」



「私は母と焼肉を食べた」



「彼はそこで彼女と縄跳びをした」

「太郎は花子と公園で遊んだ。」

「彼はそこで彼女と縄跳びをした」

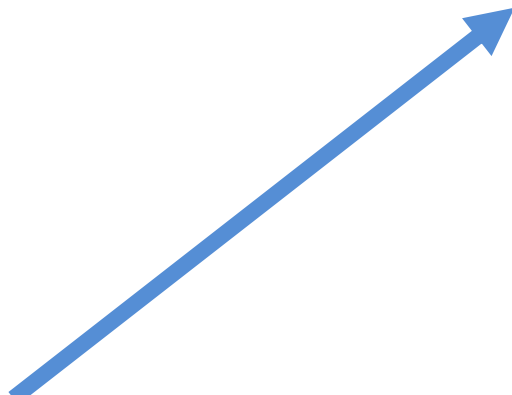
「太郎は花子と公園で遊んだ。」



「彼はそこで彼女と縄跳びをした」

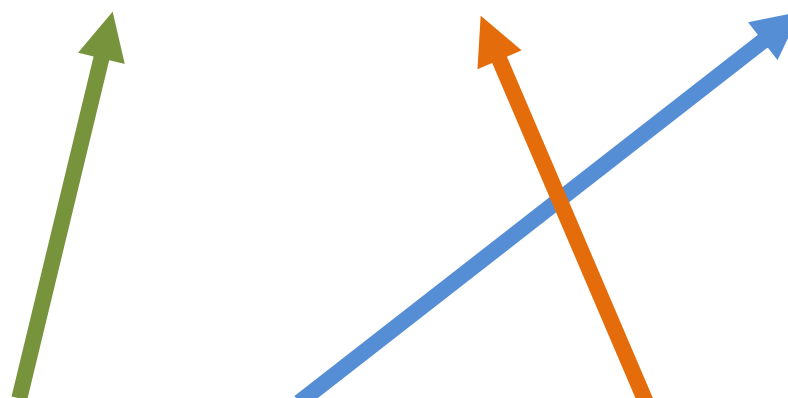
「太郎は花子と公園で遊んだ。」

「彼はそこで彼女と縄跳びをした」



「太郎は花子と公園で遊んだ。」

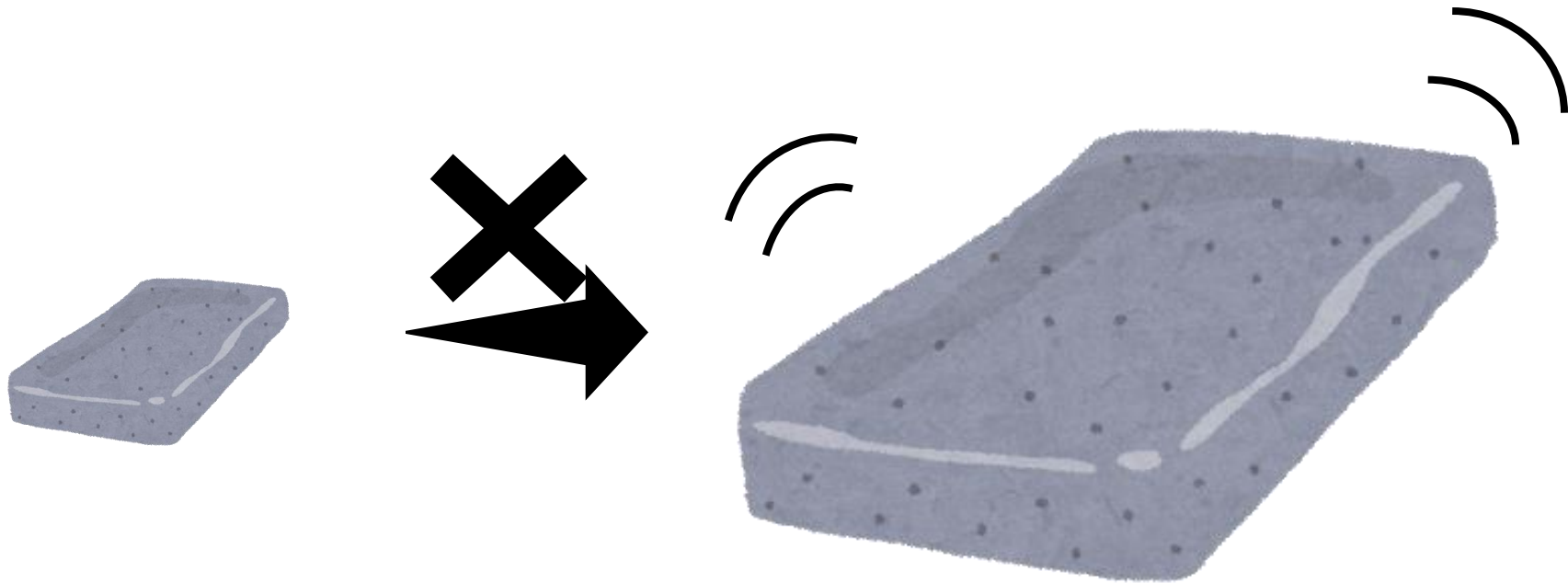
「彼はそこで彼女と縄跳びをした」



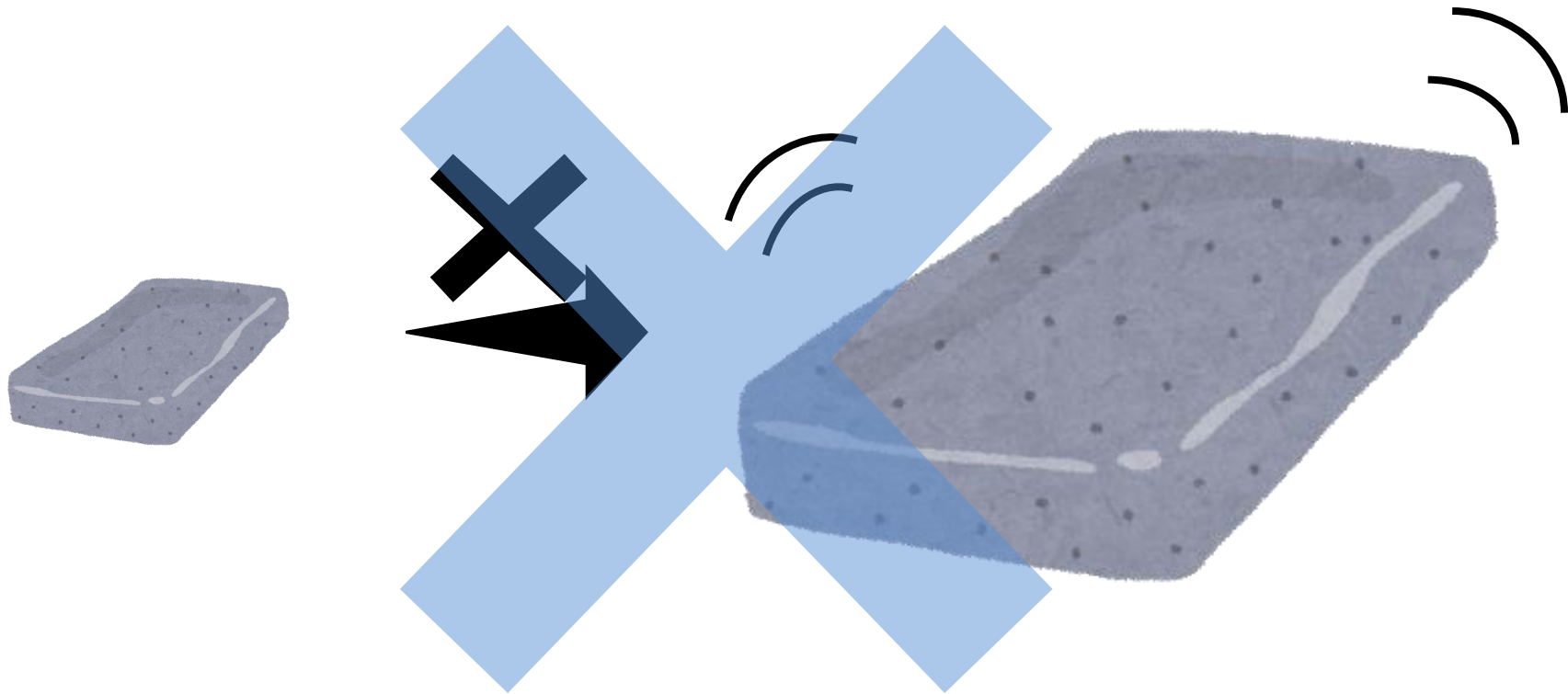
「こんにゃくは太りません」



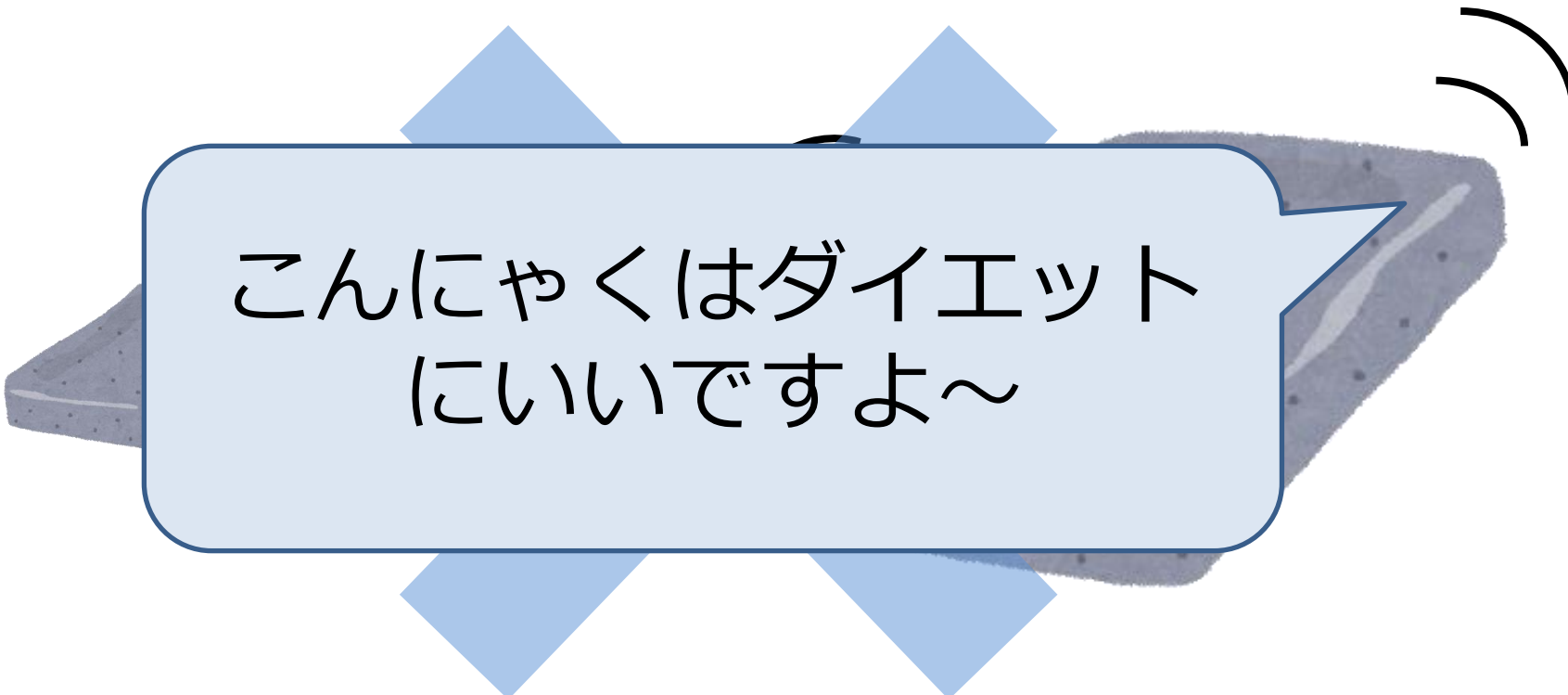
「こんにゃくは太りません」



「こんにゃくは太りません」



「こんにゃくは太りません」




こんにゃくはダイエット
にいいですよ～

アジェンダ

- ・ オープニング
- ・ **自己紹介**
- ・ 自然言語処理ってなに
- ・ 自然言語処理ってどうやってるの
- ・ 自然言語処理を使ってみる
- ・ COTOHA APIの紹介
- ・ クロージング

自己紹介

- ・ 名前：村上 優樹 （むらかみ ゆうき）入社2年目
- ・ 学生時代：自然言語処理・機械学習
- ・ 所属：AC部AI推進室 The logo for COTOHA API, featuring a stylized orange and yellow circle on the left, the text "Communication Engine" in small blue font above "COTOHA API" in large blue font on the right.
- ・ 業務：COTOHA, NLP, 日本語自然言語処理開発
- ・ 趣味：吹奏楽 （トロンボーン）
ボードゲーム （ドミノオンとか）
ゲームAI

自己紹介

竹越 智也

▪ 経歴

新卒入社2年目のペーペー

▪ 普段やってること

自然言語処理サービスの開発
(アプリケーション、インフラ)

▪ 最近の興味

Google Cloud Platform

▪ 趣味

卓球、食べ歩き、長眠



アジェンダ

- ・ オープニング
- ・ 自己紹介
- ・ **自然言語処理ってなに**
- ・ 自然言語処理ってどうやってるの
- ・ 自然言語処理を使ってみる
- ・ COTOHA APIの紹介
- ・ クロージング

自然言語処理ってなに

- 人間の言葉をコンピュータで自動的に処理させる技術
 - インターネットの普及
 - 大量のテキストデータがwebに溢れている
- 高速・自動的に・適切に大量のテキストを処理したい

• 自然言語

- 私達が使う言語
- 解釈が様々、曖昧

母と焼肉を食べた。

• 人工言語

- プログラミング言語など
- 解釈が1通りに決まる

[私,母] eat (焼肉)

[私] eat ([母,焼肉])

自然言語処理ってなに

- ・ 自然言語処理は解釈が様々、曖昧

「黒い瞳の大きな女の子」



⇒ 解析することでその解釈を示す

自然言語処理ってなに

実は結構身近に存在する

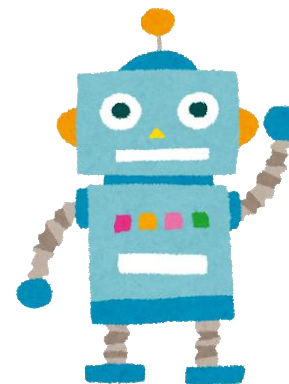
・ 検索



・ メールフィルタ



・ チャットボット



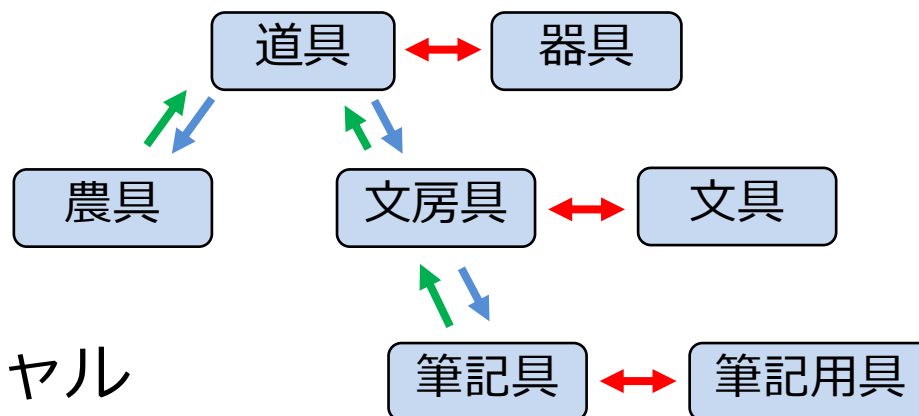
自然言語処理事例

■ 検索

シソーラスの利用によりあいまいな単語でも検索可能

シソーラス：単語を、上位/下位関係、部分/全体関係、同義関係、類義関係などによって分類、体系づけた語彙集

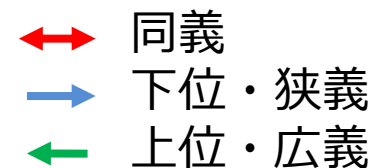
- ・ 関連語に対応



- ・ 表記ゆれに対応

ヴァーチャル、バーチャル
引越、引っ越し

- ・ タイポの訂正
もしかして：



自然言語処理事例

■ メールフィルタ

ベイジアンフィルタで、メールの怪しさを評価
メール中から怪しい単語を探し、全体の怪しさを評価

ベイジアンフィルタ：文中の単語の組み合わせにより、文書を複数
クラスにクラスタリングするための分類器

自然言語処理をしない場合

最近裸眼だと見えづらくなっ
てきたからメガネにしたよ。

迷惑メールへ

自然言語処理をする場合

最近裸眼だと見えづらくなっ
てきたからメガネにしたよ。

迷惑メールへ振り分けられない

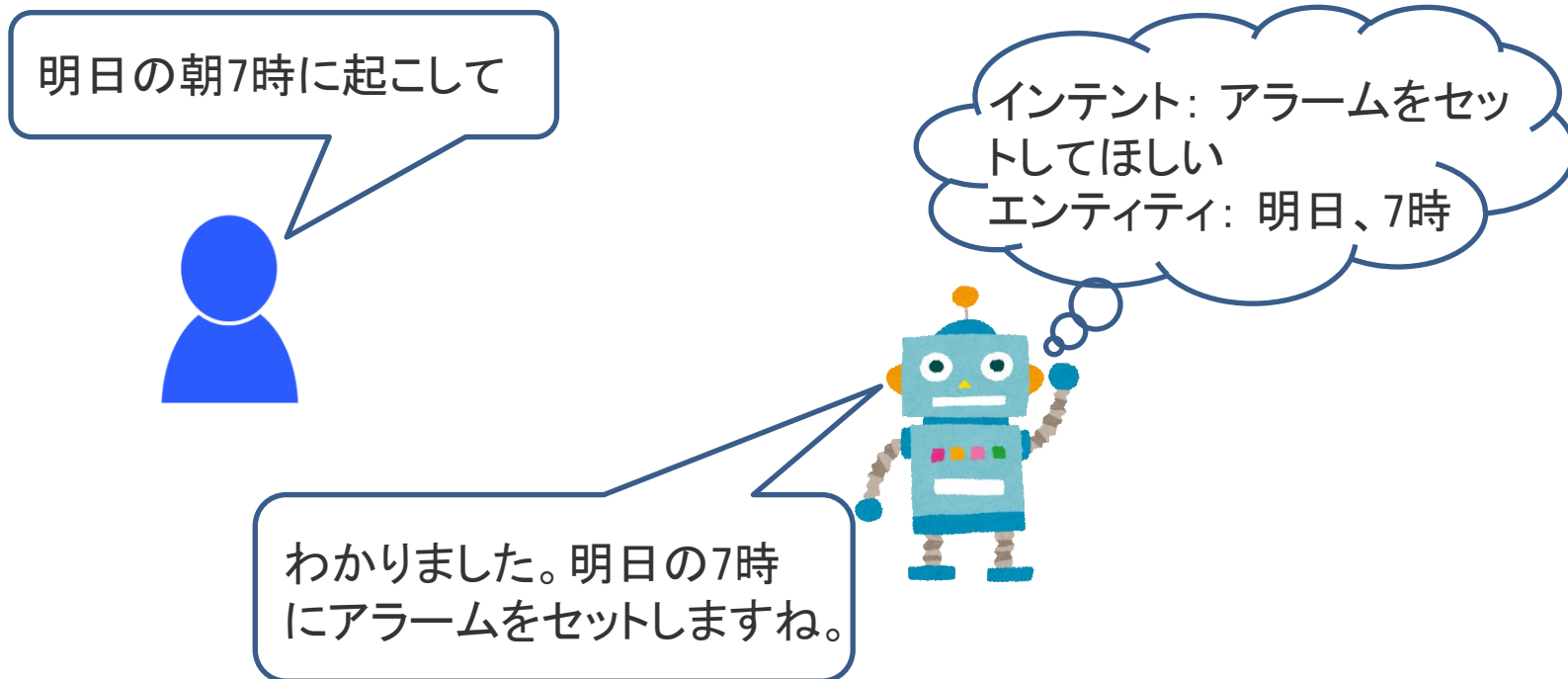
自然言語処理事例

■ チャットボット

ユーザの発言からインテントとエンティティを抽出

インテント：ユーザの発言が、こういった意図を持っているのか

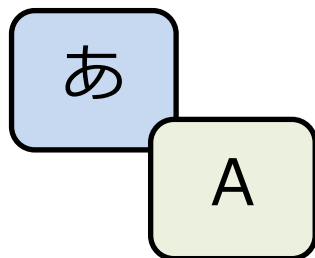
エンティティ：ユーザの発言のなかにある、意味のある単語



自然言語処理事例

紹介したほかにも

・ 翻訳



・ レコメンド



・ 文書分類



などなど応用先多数

アジェンダ

- ・ オープニング
- ・ 自己紹介
- ・ 自然言語処理ってなに
- ・ **自然言語処理ってどうやってるの**
- ・ 自然言語処理を使ってみる
- ・ COTOHA APIの紹介
- ・ クロージング

要素技術の紹介

黒い瞳の大きな女の子

形態素解析



構文解析

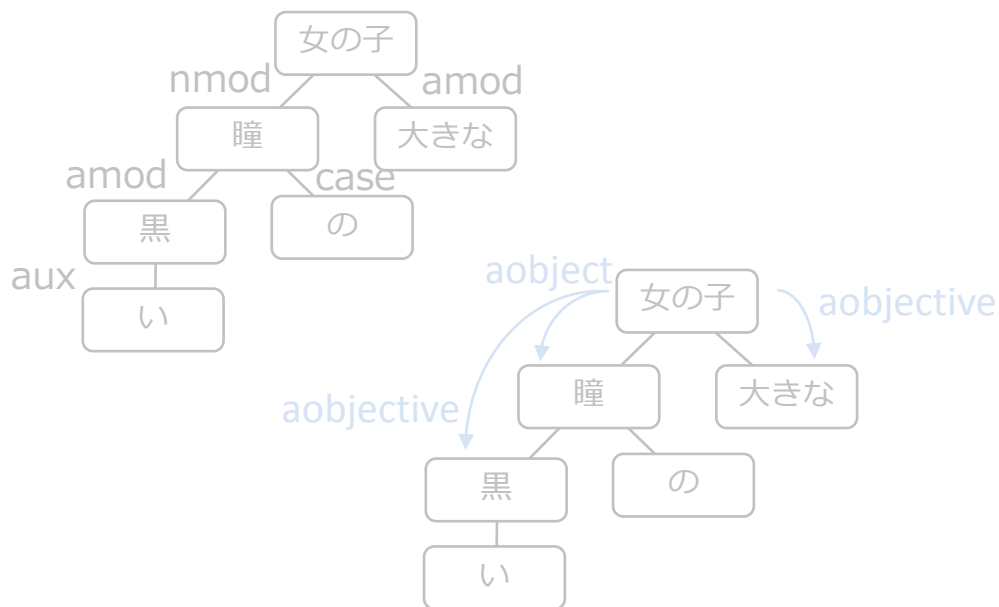


意味解析



文脈解析

黒 い 瞳 の 大きな 女の子



黒い瞳の大きな女の子にあった。

彼女は可愛かった。

要素技術の紹介

黒い瞳の大きな女の子

形態素解析



構文解析

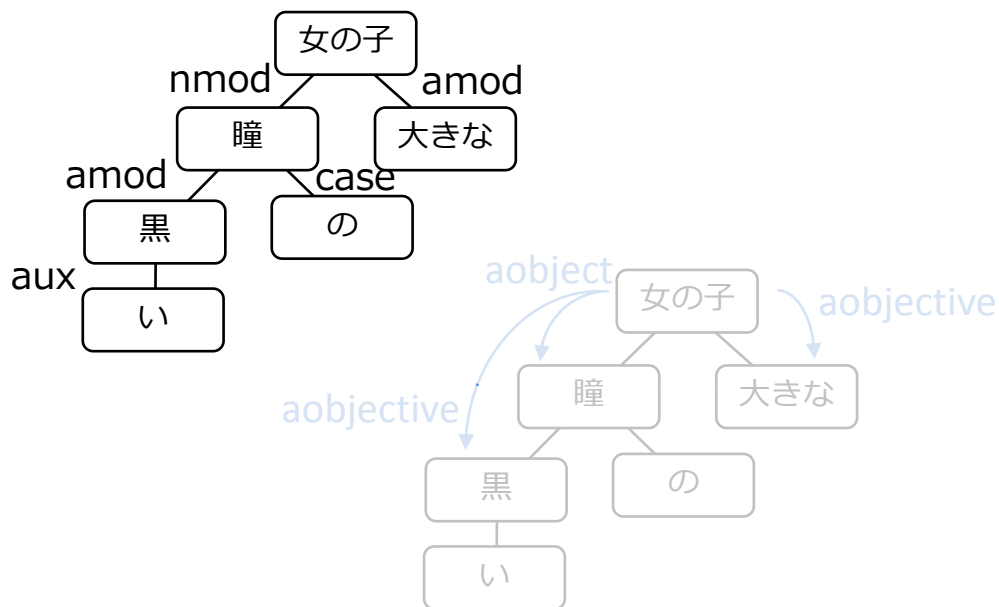


意味解析



文脈解析

黒 い 瞳 の 大きな 女の子



黒い瞳の大きな女の子にあった。

彼女は可愛かった。

要素技術の紹介

黒い瞳の大きな女の子

形態素解析



構文解析

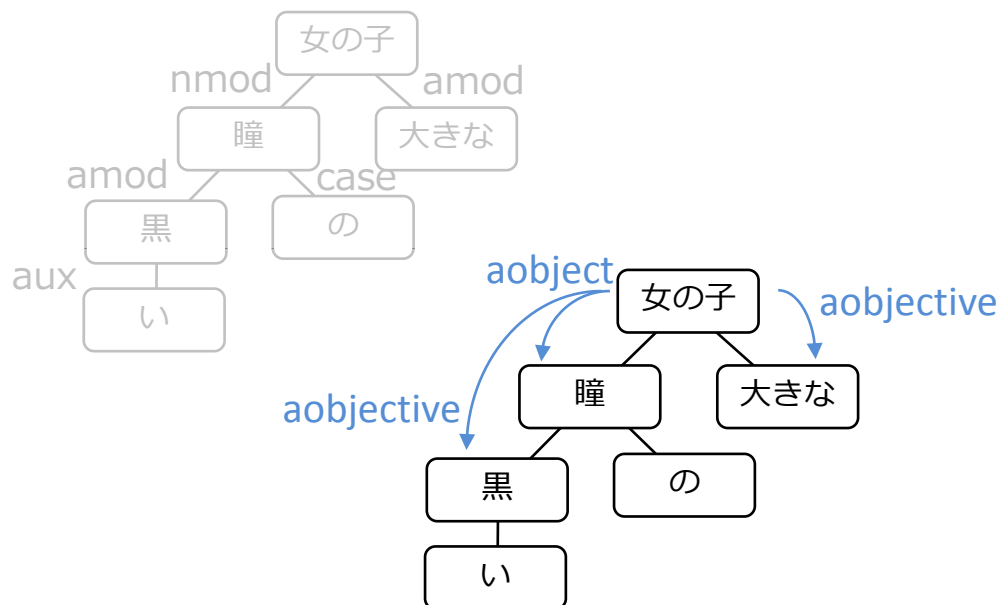


意味解析



文脈解析

黒 い 瞳 の 大きな 女の子



黒い瞳の大きな女の子にあった。

彼女は可愛かった。

要素技術の紹介

黒い瞳の大きな女の子

形態素解析



構文解析

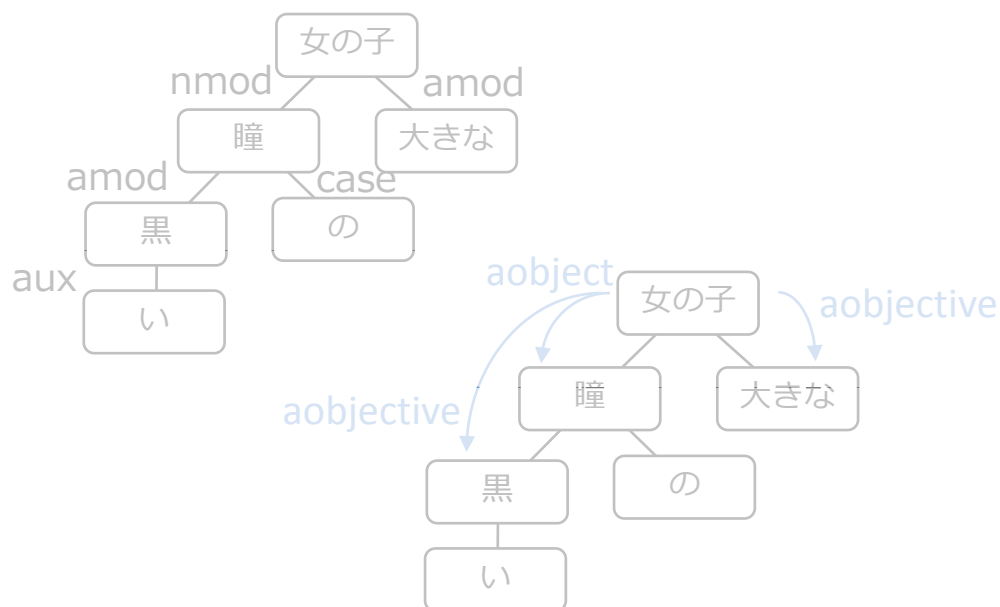


意味解析



文脈解析

黒 い 瞳 の 大きな 女の子



黒い瞳の大きな女の子にあった。

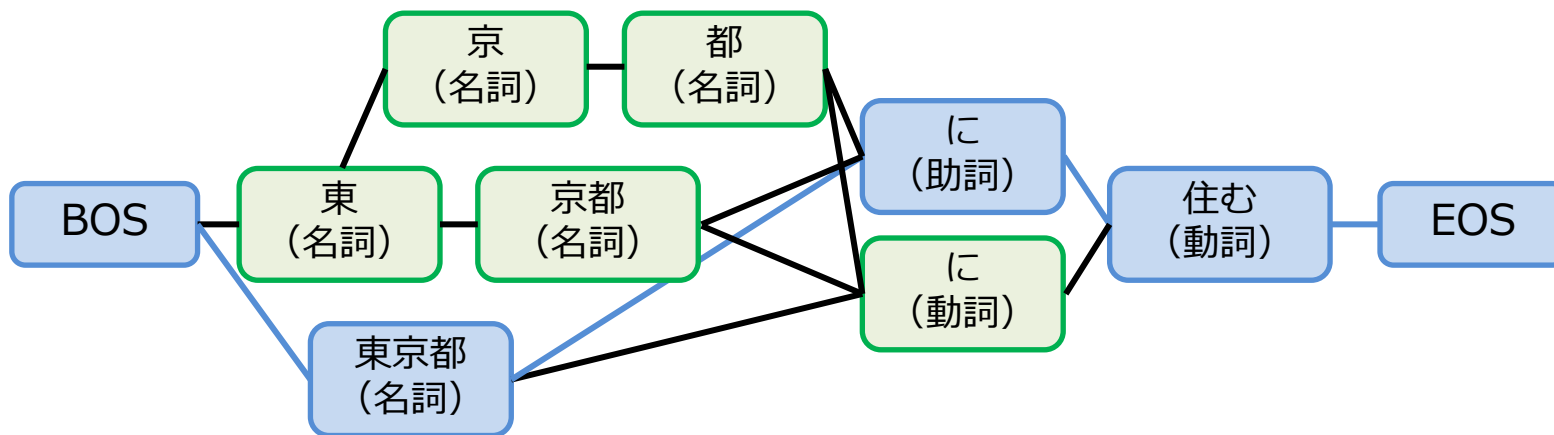
彼女は可愛かった。

形態素解析

- ・ 文を最小の意味を持つ言語単位にまで分解し、それらの単位の性質を明らかにする処理。
- ・ 日本語や中国語など、単語の切れ目がない言語で必要

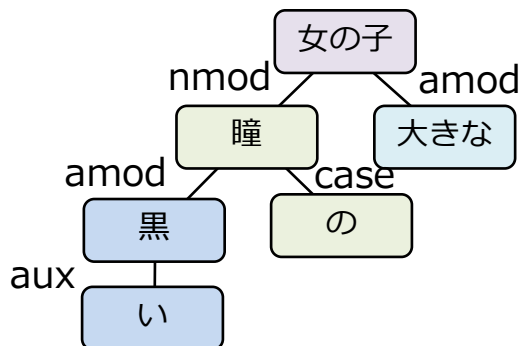
「すももももももものうち」
 スモモ も 桃 も 桃 の 内

「東京都に住む」

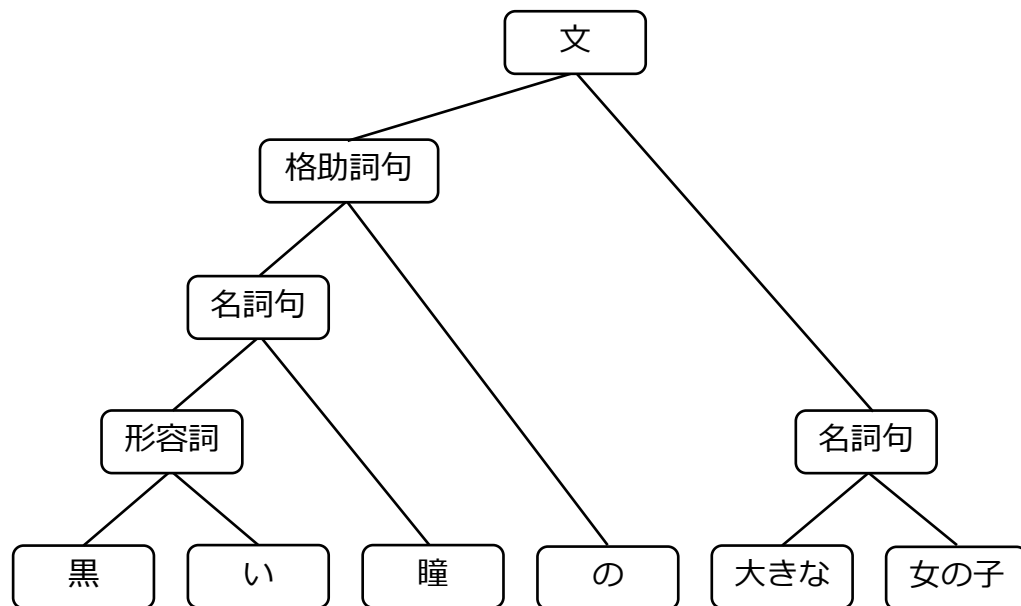


構文解析

- 係り受け解析
単語と単語のつながりを解析
単語をまとめた文節を認識

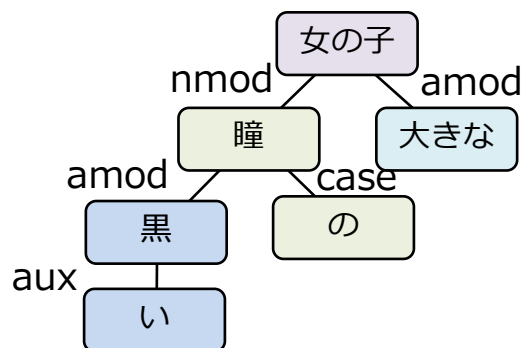


- 句構造解析
句としてのまとまりを解析



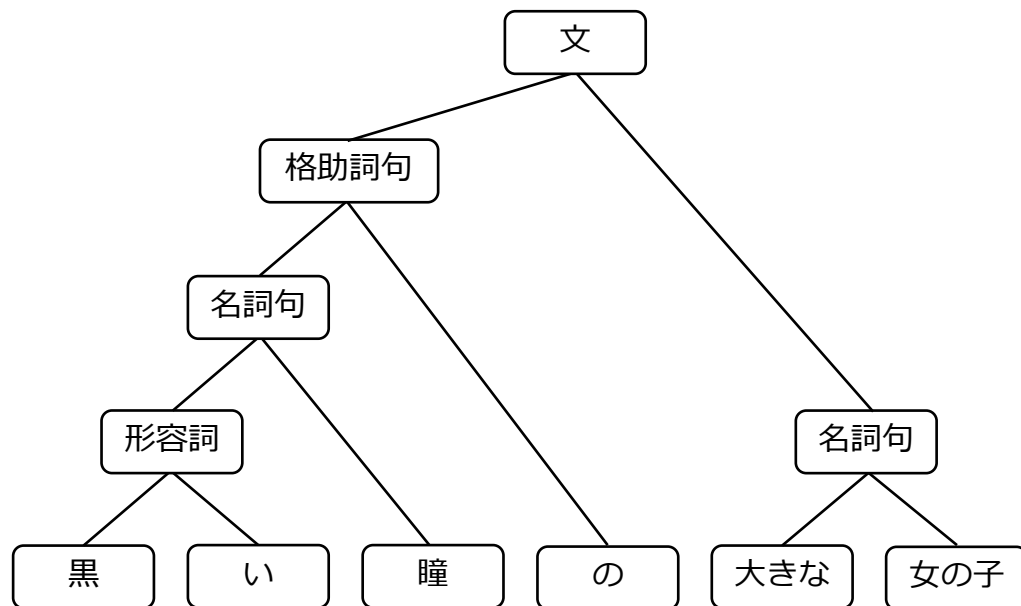
構文解析

- 係り受け解析
単語と単語のつながりを解析
単語をまとめた文節を認識



語順が自由な日本語でよく用いられる

- 句構造解析
句としてのまとまりを解析

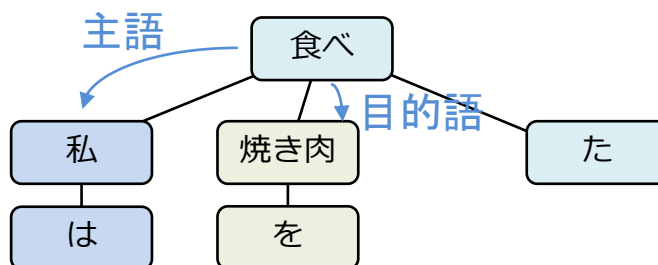


語順が自由な日本語には向かない

意味解析

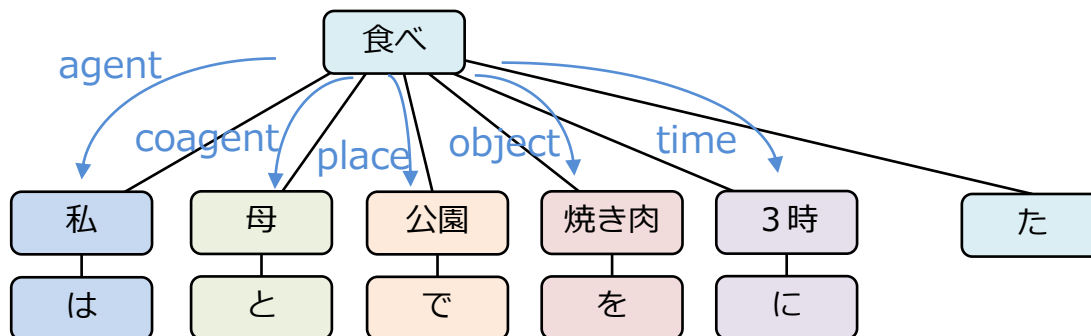
・ 述語項構造解析

述語から見たときの主語・目的語を明らかにする



・ 意味役割解析

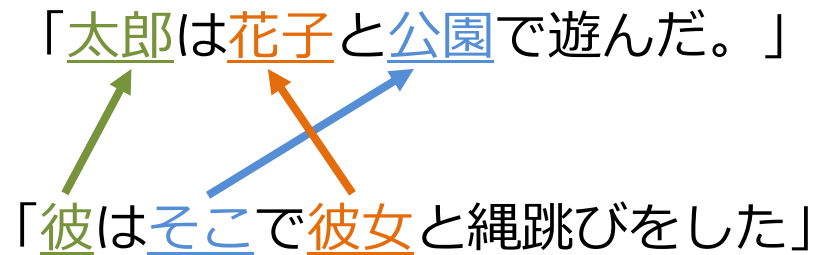
述語から見たときの各項の意味役割を明らかにする



agent:動作主
 coagent:動作主と行動を共にする人
 object:対象
 place:場所
 time:時間

文脈解析

- 文章全体での文間の関係を明らかにする
- 照応解析
文章中で指示代名詞の示す対象を明らかにする



- 談話構造解析
文間の役割関係を明らかにする

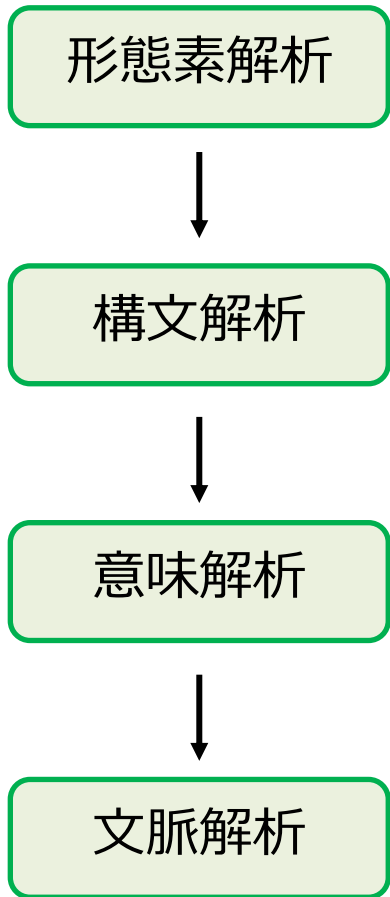
「自然言語処理は人間の言葉を処理する技術である」

↓ 例示

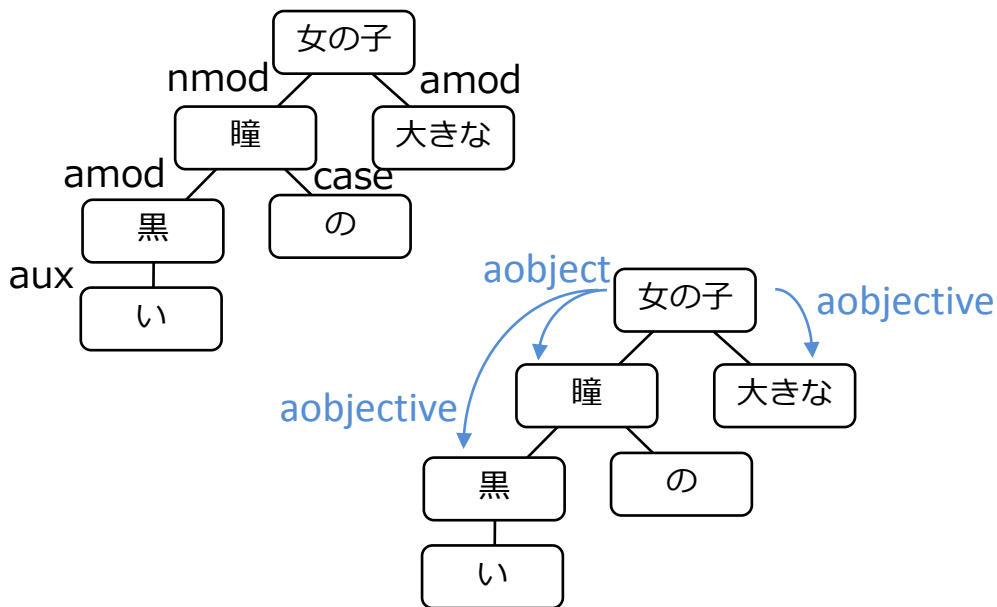
「形態素解析や構文解析などがある」

要素技術の紹介

黒い瞳の大きな女の子



黒 い 瞳 の 大きな 女の子



黒い瞳の大きな女の子にあった。

彼女は可愛かった。

アジェンダ

- ・ オープニング
- ・ 自己紹介
- ・ 自然言語処理ってなに
- ・ 自然言語処理ってどうやってるの
- ・ **自然言語処理を使ってみる**
- ・ COTOHA APIの紹介
- ・ クロージング

自然言語処理を使ってマジカルバナナ

- マジカルバナナとは
前の言葉から連想できる言葉を答えていくゲーム



バナナ

と言ったら

すべる

と言ったら

氷

と言ったら

冷蔵庫

自然言語処理を使ってマジカルバナナ

- ・あるお題に対してコンピュータに連想される語を答えさせたい

バナナ

赤色

青

地面

黄色

赤

空



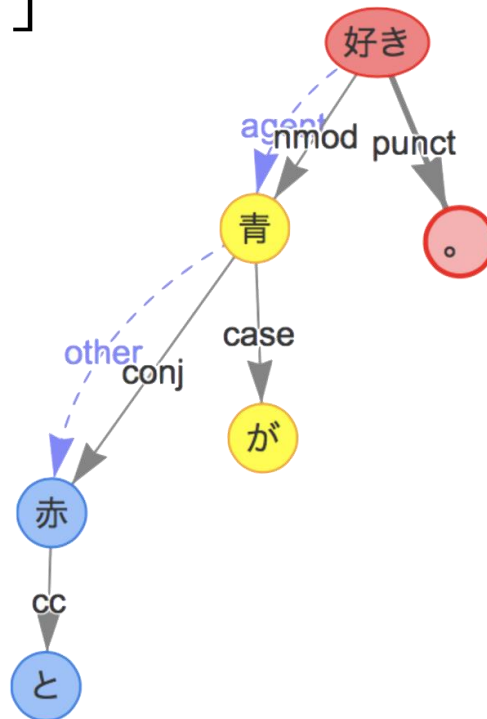
自然言語処理を使ってマジカルバナナ

- 返すべき語
 - 同じ種類の単語
赤ー青
 - 対応する語
空ー地面
 - 名詞ー動詞
鳥ー飛ぶ
 - 名詞ー形容詞
バナナー黄色い
 - クラスーインスタンス
電車ー山手線

→構文情報からペアをルールで抽出する

ルールによる構文情報からペアを抽出

- 同じ種類の語、対応する語
「赤と青が好き」
赤—青

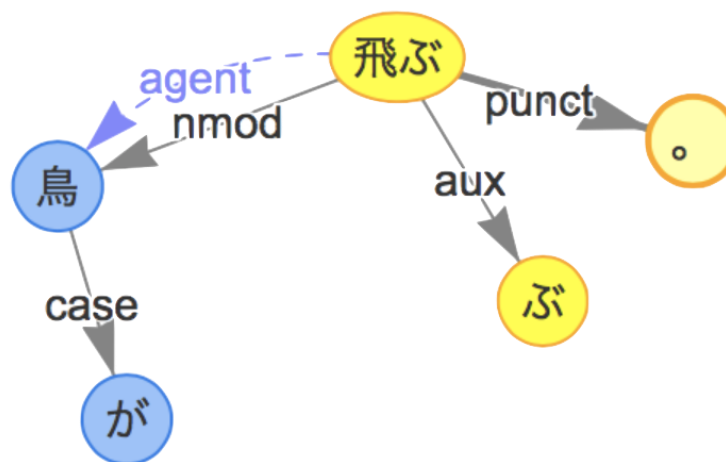


N: 名詞
V: 動詞
A: 形容詞
X: 品詞問わず

$[N1, conj, N2] \ \& \ [N1, other, N2] \ ==\ ==\ ==\ ==\ > \ (N1, N2)$

ルールによる構文情報からペアを抽出

- 名詞－動詞
「鳥が飛ぶ」
鳥－飛ぶ



N: 名詞
V: 動詞
A: 形容詞
X: 品詞問わず

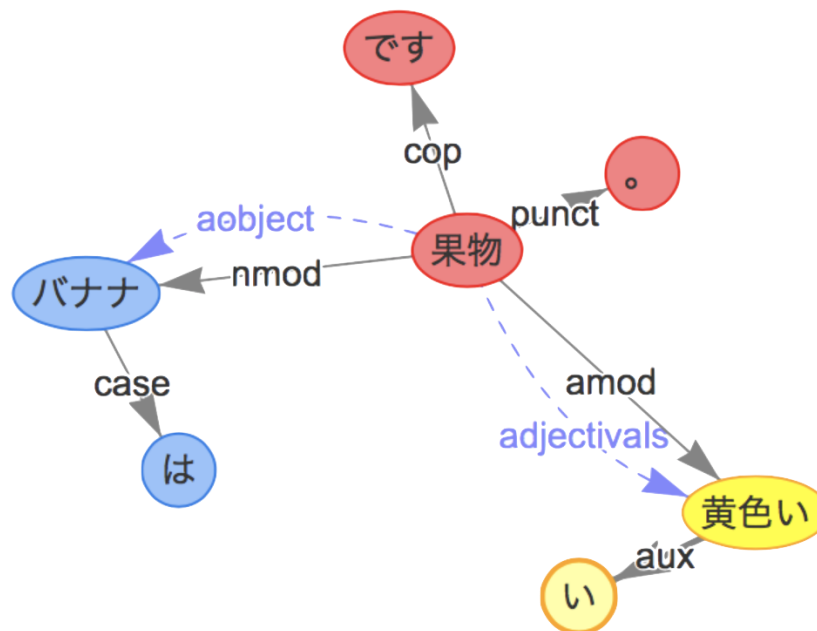
$[V, agent, N] == => (N, V)$

ルールによる構文情報からペアを抽出

- 名詞－形容詞

「バナナは黄色い果物です」

バナナ－黄色い

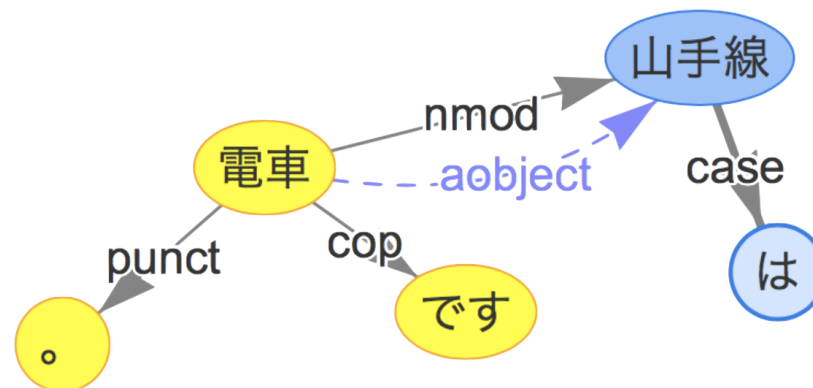


N: 名詞
V: 動詞
A: 形容詞
X: 品詞問わず

$[N1, aobject, N2] \ \& \ [N1, adjectivals, A] \ ==\ ==\ ==\ ==\ > \ (N2, A)$

ルールによる構文情報からペアを抽出

- ・ クラスーインスタンス
「山手線は電車です。」
電車ー山手線



N: 名詞
V: 動詞
A: 形容詞
X: 品詞問わず

$[N1, aobject, N2] \ \& \ [N1, cop, X] \implies (N2, N1)$

ルールによる構文情報からペアを抽出

- 他にも以下のルールを採用

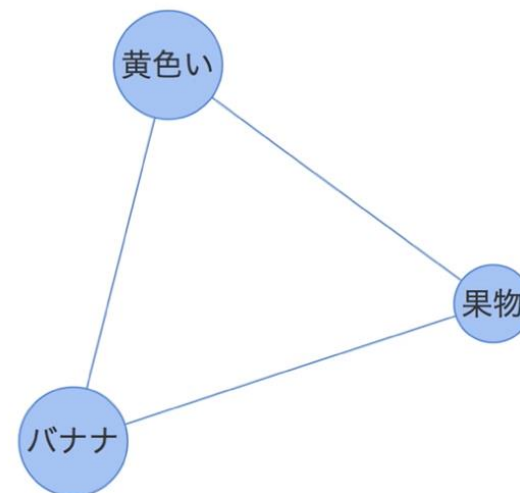
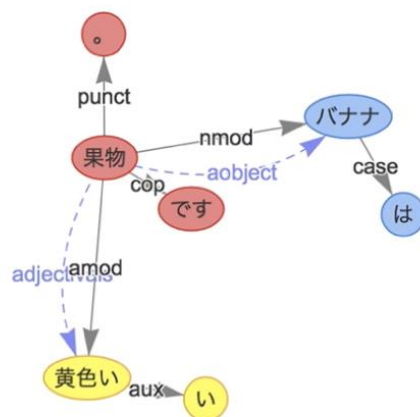
[N1,aobject,N2] =====> (N2,N1)
[N1,agent,N2] =====> (N2,N1)
[N1,aobject,N2] & [N1,adjectivals,A] =====> (N2,A)
[N1,aobject,N2] & [N1,cop,X] =====> (N2,N1)
[X,agent,N1] & [X,cause,N2] =====> (N1,N2)
[X,agent,N1] & [X,adjectivals,N2] =====> (N1,N2)
[N,adjectivals,A] =====> (N,A)
[N,adjectivals,N2] =====> (N,N2)
[A,aobject,N] =====> (N,A)
[N1,conj,N2] & [N1,other,N2] =====> (N1,N2)
[N1,nmod,N2] & [N1,other,N2] =====> (N1,N2)
[N1,adjectivals,X] & [X,agent,N2] =====> (N2,N1)
[N1,adjectivals,N2] & [N1,nmod,N2] & [N2,case,X] =====> (N1,N2)
[N1,aobject,N2] & [N1,amod,X] =====> (N2,X)
[N1,aobject,N2] & [N1,amod,A] =====> (N1,A)
[N1,amod,A] =====> (N1,A)
[V,agent,N] =====> (N,V)

自然言語処理による連想ペアネットワーク作成デモ

Semantic Network DEMO

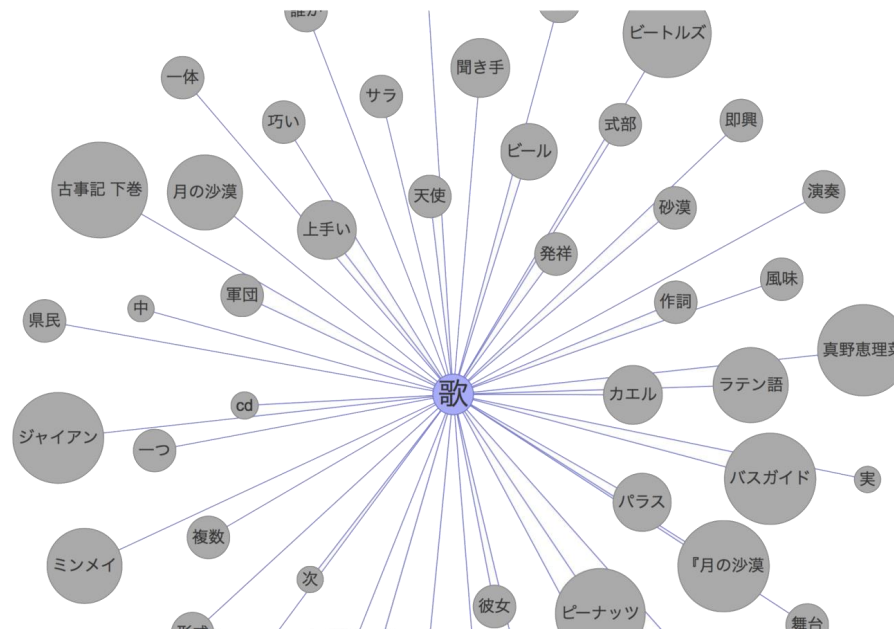
バナナは黄色い果物です。

parse



連想ペアネットワーク

- ・ 隣接するノードのいずれかを返答する



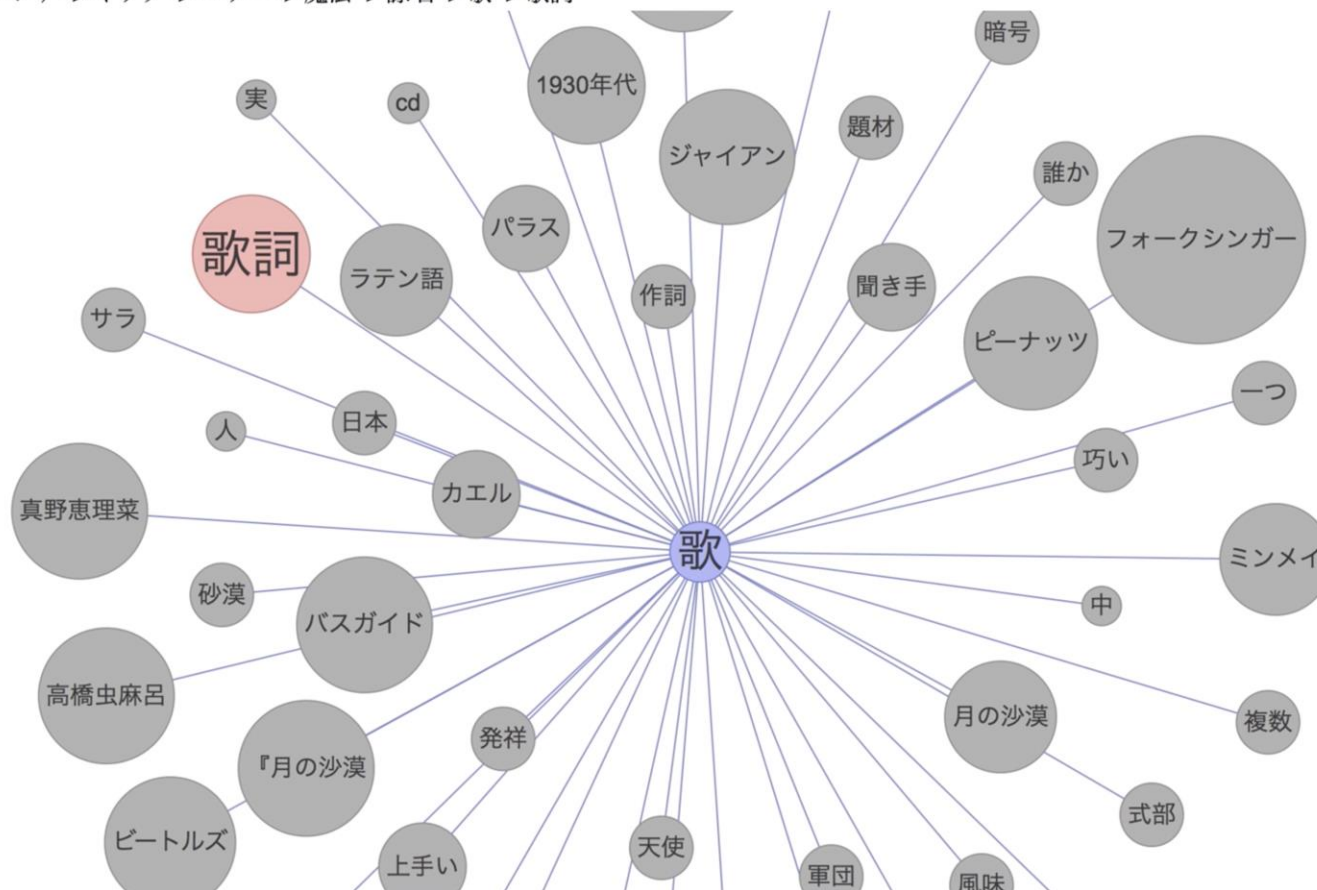
- ・ 変な解答をすることがあるので、単語間の類似度で足切りをする

マジカルバナナ デモ

MAGICAL BANANA DEMO

歌

バナナ -> 黄色 -> 赤色 -> 赤 -> 青 -> 空 -> 地面 -> 土 -> 水 -> 海 -> 彼方 -> 遠い -> 距離 -> 長さ -> 配列 -> リスト -> カテゴリー -> 種類 -> 組み合わせ -> コンビネーション -> パンチ -> キック -> パワー -> 魔法 -> 詠唱 -> 歌 -> 歌詞



アジェンダ

- ・ オープニング
- ・ 自己紹介
- ・ 自然言語処理ってなに
- ・ 自然言語処理ってどうやってるの
- ・ 自然言語処理を使ってみる
- ・ **COTOHA APIの紹介**
- ・ クロージング

COTOHA APIとは

様々自然言語処理技術を扱うAPI
無料で試せるのでぜひ以下のリンクから

COTOHA API








検索



<https://api.ce-cotoha.com>








COTOHA API

・API 一覧

1	 構文解析	日本語テキストの構造と意味を解析します。
2	 固有表現抽出	人名や地名などの固有表現を抽出します。
3	 照応省略解析	「あれ」「これ」「それ」「そこ」等の指示語・省略語を検知し、指し示す対象を特定します。
4	 キーワード抽出	文章からキーワードを抽出します。
5	 文タイプ判定	叙述文、命令文、質問文等の文タイプを出力します。
6	 ユーザ属性推定	文章からユーザの年代、職業等の属性を推定します。
7	 類似度算出	2つの文章の類似性を数値化し出力します。




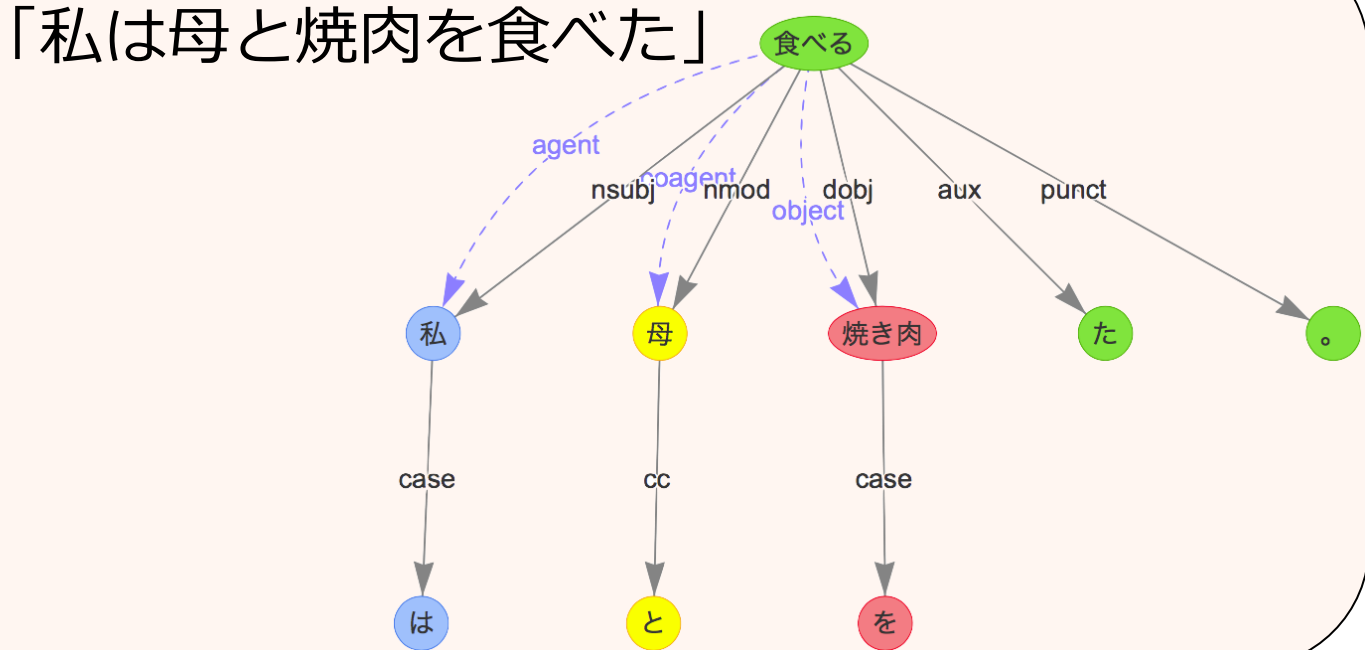




COTOHA API

・ API 一覧

1	 構文解析	日本語テキストの構造と意味を解析します。
2	 固有表現抽出	人名や地名などの固有表現を抽出します。
3	 照応省略解析	「あれ」「これ」「それ」「そこ」等の指示語・省略語を検知し、指し示す対象を特定します。
4	 キーワード抽出	文章からキーワードを抽出します。
5	 文タイプ判定	叙述文、命令文、質問文等の文タイプを出力します。
6	 ユーザ属性推定	文章からユーザの年代、職業等の属性を推定します。
7	 類似度算出	2つの文章の類似性を数値化し出力します。








COTOHA API

・ API 一覧

1	 構文解析	日本語テキストの構造と意味を解析します。
2	 固有表現抽出	人名や地名などの固有表現を抽出します。
3		<p>「私は母と焼肉を食べた」</p> 
4		
5		
6		
7		








COTOHA API

・ API 一覧

1	 構文解析	日本語テキストの構造と意味を解析します。
2	 固有表現抽出	人名や地名などの固有表現を抽出します。
3	 照応省略解析	「あれ」「これ」「それ」「そこ」等の指示語・省略語を検知し、指し示す対象を特定します。
4	 キーワード抽出	文章からキーワードを抽出します。
5	 文タイプ判定	叙述文、命令文、質問文等の文タイプを出力します。
6	 ユーザ属性推定	文章からユーザの年代、職業等の属性を推定します。
7	 類似度算出	2つの文章の類似性を数値化し出力します。

COTOHA API

・API 一覧

1	 構文解析	日本語テキストの構造と意味を解析します。
2	 固有表現抽出	人名や地名などの固有表現を抽出します。
3	 照応省略解析	「これ」「それ」「そこ」等の指示語・省略語を 指し示す対象を特定します。
4	 キ	
5	 文	
6	 ユ	
7	 類	

田中は昨日富士山に登った。








「田中」
人名

「昨日」
時間

「富士山」
地名








COTOHA API

・ API 一覧

1	 構文解析	日本語テキストの構造と意味を解析します。
2	 固有表現抽出	人名や地名などの固有表現を抽出します。
3	 照応省略解析	「あれ」「これ」「それ」「そこ」等の指示語・省略語を検知し、指し示す対象を特定します。
4	 キーワード抽出	文章からキーワードを抽出します。
5	 文タイプ判定	叙述文、命令文、質問文等の文タイプを出力します。
6	 ユーザ属性推定	文章からユーザの年代、職業等の属性を推定します。
7	 類似度算出	2つの文章の類似性を数値化し出力します。

COTOHA API

・ API 一覧








1	 構文解析	日本語テキストの構造と意味を解析します。
2	 固有表現抽出	人名や地名などの固有表現を抽出します。
3	 照応省略解析	「あれ」「これ」「それ」「そこ」等の指示語・省略語を検知し、指し示す対象を特定します。
4	 キーワード抽出	キーワードを抽出します。
5	 文	
6	 ユーザ	
7	 種類	

「太郎は花子と公園で遊んだ。」

「彼はそこでで彼女と縄跳びをした」








COTOHA API

・ API 一覧

1	 構文解析	日本語テキストの構造と意味を解析します。
2	 固有表現抽出	人名や地名などの固有表現を抽出します。
3	 照応省略解析	「あれ」「これ」「それ」「そこ」等の指示語・省略語を検知し、指し示す対象を特定します。
4	 キーワード抽出	文章からキーワードを抽出します。
5	 文タイプ判定	叙述文、命令文、質問文等の文タイプを出力します。
6	 ユーザ属性推定	文章からユーザの年代、職業等の属性を推定します。
7	 類似度算出	2つの文章の類似性を数値化し出力します。

COTOHA API

・ API 一覧

1	 構	
2	 固	
3	 照応省略解析	「それ」「そこ」等の指示語・省略語を、指し示す対象を特定します。
4	 キーワード抽出	文章からキーワードを抽出します。
5	 文タイプ判定	叙述文、命令文、質問文等の文タイプを出力します。
6	 ユーザ属性推定	文章からユーザの年代、職業等の属性を推定します。
7	 類似度算出	2つの文章の類似性を数値化し出力します。








「昨日は久しぶりに懐かしい友人
とレストランで昼食を食べた。」



友人 レストラン 昼食








COTOHA API

・ API 一覧

1	 構文解析	日本語テキストの構造と意味を解析します。
2	 固有表現抽出	人名や地名などの固有表現を抽出します。
3	 照応省略解析	「あれ」「これ」「それ」「そこ」等の指示語・省略語を検知し、指し示す対象を特定します。
4	 キーワード抽出	文章からキーワードを抽出します。
5	 文タイプ判定	叙述文、命令文、質問文等の文タイプを出力します。
6	 ユーザ属性推定	文章からユーザの年代、職業等の属性を推定します。
7	 類似度算出	2つの文章の類似性を数値化し出力します。








COTOHA API

・API 一覧

1		構	「食べました」 → 宣言文・情報提供
2		固	「食べましたか」 → 疑問文・情報獲得
3		照	「食べなさい」 → 命令文・命令
4		キーワード抽出	キーワードを抽出します。
5		文タイプ判定	叙述文、命令文、質問文等の文タイプを出力します。
6		ユーザ属性推定	文章からユーザの年代、職業等の属性を推定します。
7		類似度算出	2つの文章の類似性を数値化し出力します。








COTOHA API

・ API 一覧

1	 構文解析	日本語テキストの構造と意味を解析します。
2	 固有表現抽出	人名や地名などの固有表現を抽出します。
3	 照応省略解析	「あれ」「これ」「それ」「そこ」等の指示語・省略語を検知し、指し示す対象を特定します。
4	 キーワード抽出	文章からキーワードを抽出します。
5	 文タイプ判定	叙述文、命令文、質問文等の文タイプを出力します。
6	 ユーザ属性推定	文章からユーザの年代、職業等の属性を推定します。
7	 類似度算出	2つの文章の類似性を数値化し出力します。

COTOHA API

・ API 一覧

1		構	
2		固	
3		照	
4		キ	
5		文タイプ判定	文、質問文等の文タイプを出力します。
6		ユーザ属性推定	文章からユーザの年代、職業等の属性を推定します。
7		類似度算出	2つの文章の類似性を数値化し出力します。

「私は昨日田町駅で飲みに行ったら奥さんに怒られた。」



年齢：40~49歳

既婚：yes

習慣：飲酒








職業：会社員

通勤手段：電車

趣味：動物、料理、釣り、ギャンブル…








COTOHA API

・ API 一覧

1	 構文解析	日本語テキストの構造と意味を解析します。
2	 固有表現抽出	人名や地名などの固有表現を抽出します。
3	 照応省略解析	「あれ」「これ」「それ」「そこ」等の指示語・省略語を検知し、指し示す対象を特定します。
4	 キーワード抽出	文章からキーワードを抽出します。
5	 文タイプ判定	叙述文、命令文、質問文等の文タイプを出力します。
6	 ユーザ属性推定	文章からユーザの年代、職業等の属性を推定します。
7	 類似度算出	2つの文章の類似性を数値化し出力します。

COTOHA API








・ API 一覧

1	 構文解析	日本語テキストの構造と意味を解析します。
2	 固有表現抽出	人名や地名などの固有表現を抽出します。
3	 照	
4	 キ	
5	 文	
6	 ユーザ属性推定	ユーザの年代、職業等の属性を推定します。
7	 類似度算出	2つの文章の類似性を数値化し出力します。

「近くのレストランはどこですか」
「この辺りの定食屋はどこにありますか」
→ 類似度 : 0.91079295

COTOHA API

・API 一覧

1	 構文解析	日本語テキストの構造と意味を解析します。
2	 固有表現抽出	人名や地名などの固有表現を抽出します。
3	 照応省略解析	「あれ」「これ」「それ」「そこ」等の指示語・省略語を検知し、指し示す対象を特定します。
4	 キーワード抽出	文章からキーワードを抽出します。
5	 文タイプ判定	叙述文、命令文、質問文等の文タイプを出力します。
6	 ユーザ属性推定	文章からユーザの年代、職業等の属性を推定します。
7	 類似度算出	2つの文章の類似性を数値化し出力します。

アジェンダ

- ・ オープニング
- ・ 自己紹介
- ・ 自然言語処理ってなに
- ・ 自然言語処理ってどうやってるの
- ・ 自然言語処理を使ってみる
- ・ COTOHA APIの紹介
- ・ **クロージング**

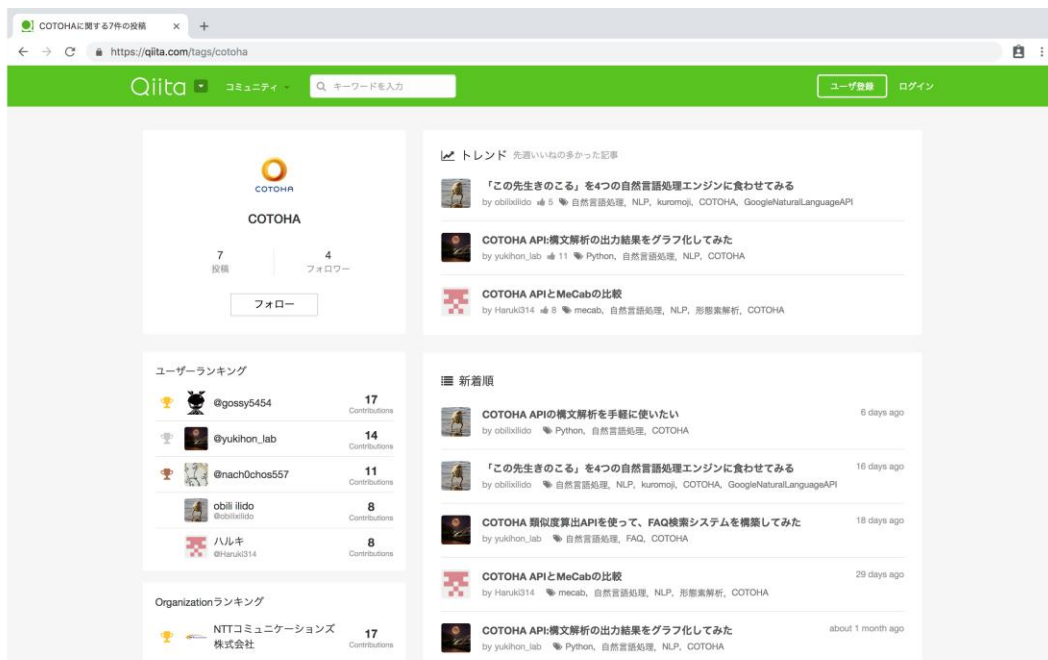
最後に

この講演で自然言語処理について少しでも興味を持ってくれたら嬉しいです。

まだまだ可能性が詰まっている分野なので皆で盛り立てていきましょう。

COTOHA APIはQiitaに記事も投稿されています。

よろしければそちらも読んでみてください。



The screenshot shows the Qiita website interface. At the top, there's a navigation bar with the Qiita logo, a search bar, and links for 'ユーザー登録' (User Registration) and 'ログイン' (Login). The main content area is divided into several sections:

- COTOHA Profile:** Shows the COTOHA logo, 7 articles, and 4 followers. A 'フォロー' (Follow) button is visible.
- トレンド (Trends):** A list of trending articles, including '「この先生きのこる」を4つの自然言語処理エンジンに食わせてみる' by obilixido.
- ユーザーランキング (User Ranking):** A table listing top contributors:

ユーザー名	投稿数 (Contributions)
@gossy5454	17
@yukihon_lab	14
@nach0chos557	11
obilixido	8
ハルキ (@Haruki314)	8
- Organization Ranking (Organization Ranking):** Shows 'NTTコミュニケーションズ株式会社' with 17 contributions.
- 新着順 (Newest):** A list of recent articles, including 'COTOHA APIの構文解析を手軽に使いたい' by obilixido.