IBM Watson

WatsonとOSSのAI/Deep Learning

日本アイ・ビー・エム株式会社
ワトソン事業部

赤石　雅典

IBM

おことわり

この資料の内容は講演者自身の個人的見解であり、所属会社の立場、戦略、意見を代表するものではありません。
この資料は執筆時点の情報を元に書いているため、必ずしも最新情報であるとはかぎりません。
この資料の内容の正確性には責任を負いません。

講演者略歴

赤石雅典

1987年日本アイ・ビー・エムに入社。入社当時は、東京基礎研究所研究員として APL2 を利用した数式処理システム、数学教育支援システムの研究開発に従事する。
1993年にSE部門に異動し、ITスペシャリストとして主にオープン系システムのインフラ設計・構築及びアプリケーションデザインを担当。
2013年よりスマーターシティ事業に転属し、2016年8月にワトソン事業部に異動、今に至る。

いろいろな領域を幅広くやっているので、基盤系からアプリ開発・プログラム言語・SQLチューニングまで一通り語れるのが自慢。

mail: akaishi@jp.ibm.com

第一部　IBM Watson紹介

Watsonとは

IBM Watson 人工知能研究とコグニティブ・システム

IBM

IBMリサーチの
グランドチャレンジ
プロジェクト

チェス対戦DeepBlue

質問応答システムWatson

- がん診断支援
- 保険査定支援
- 顧客接点変革

- 画像認識
- 音声認識

コグニティブ
システム
（コグニティブビジネス基盤）

人間の知的活動を拡張

記号処理

自然言語処理

知識表現

専門知識のルール化

脳のはたらきを再現

機械学習

ニューラルネットワーク

Deep Learning

“Artificial Intelligence”
1960

エキスパート・システム
1980

ビッグデータ活用AI基盤
2000

IBM Watson 人工知能研究とコグニティブ・システム

IBM

人間の知的活動を拡張

IBMリサーチの
グランドチャレンジ
プロジェクト

チェス対戦DeepBlue

質問応答システムWatson

- がん診断支援
- 保険査定支援
- 顧客接点変革
- 画像認識
- 音声認識

コグニティブ
システム
（コグニティブビジネス基盤）

記号処理

自然言語処理

知識表現

“Augmented Intelligence”

専門知識のルール化

脳のはたらきを再現

機械学習

ニューラルネットワーク

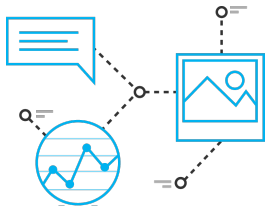
Deep Learning

“Artificial Intelligence”
1960

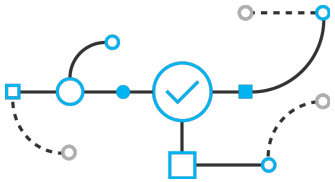
エキスパート・システム
1980

ビッグデータ活用AI基盤
2000

コグニティブ・システムの特長
～従来のコンピューター・システムとの違い



理解
（Understanding）



推論
（Reasoning）

学習
（Learning）

… 人と対話をし、必要な情報の探索や高度な意思決定を支援する
(Interacting)

IBM Watson 人間の知識を活用するコグニティブ・システム

ソリューション

知識を活用しビジネス課題を解決

Watson Solution Framework

照会応答 Engagement | 探索・発見 Discovery | 意思決定支援 Decision | …

Watsonの提供する機能

Watsonコグニティブ・サービス

質問応答 | テキスト解析 | 探索 | 発見 | … | 音声認識合成 | 画像認識 | …

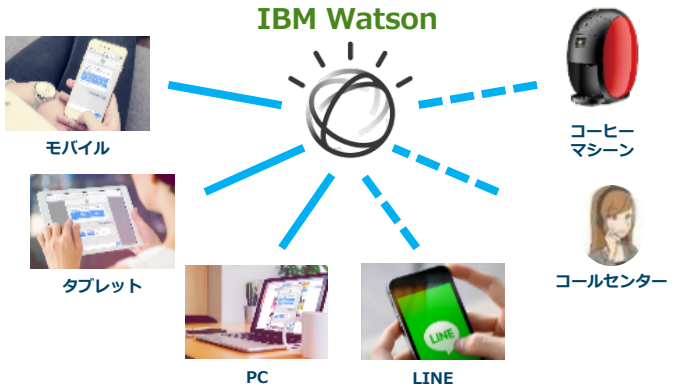
自然言語処理・知識表現・機械学習・Deep Learning

個別領域文献 企業内データ | 概念体系・辞書 | 文脈情報 | 専門家の知見

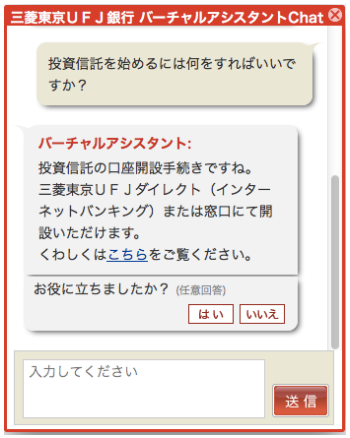
顧客接点

Watsonの顧客接点への適用 - 対話型自動応答

三菱東京UFJ銀行は10月7日より、
Webチャット形式の自動対話応答機能
“バーチャルアシスタントChatサービス”を開始。



ネスレ・ジャパンはIBM Watsonを活用した
マルチ・チャネル対応の対話型自動応答による
お客様サポート**“ネスレ・チャット・アシスタント”**を
11月21日より開始。

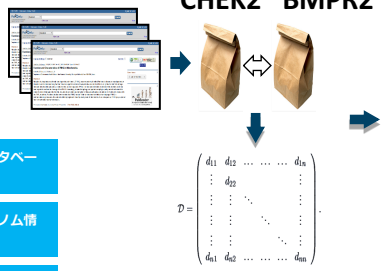


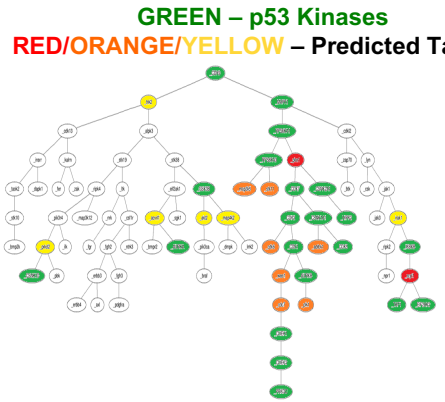
JALは12月5日 IBM Watsonを活用し
“JALバーチャルアシスタントサービス”開始。
赤ちゃん同伴のハワイ旅行の不安を解消 。

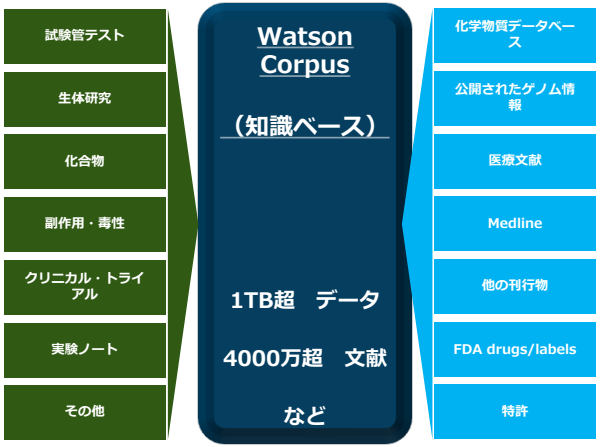
Watsonの知識探索・活用ソリューション

～ ベイラー医科大学との共同研究

膨大な生化学研究文献の分析をとおして癌治療の研究を加速







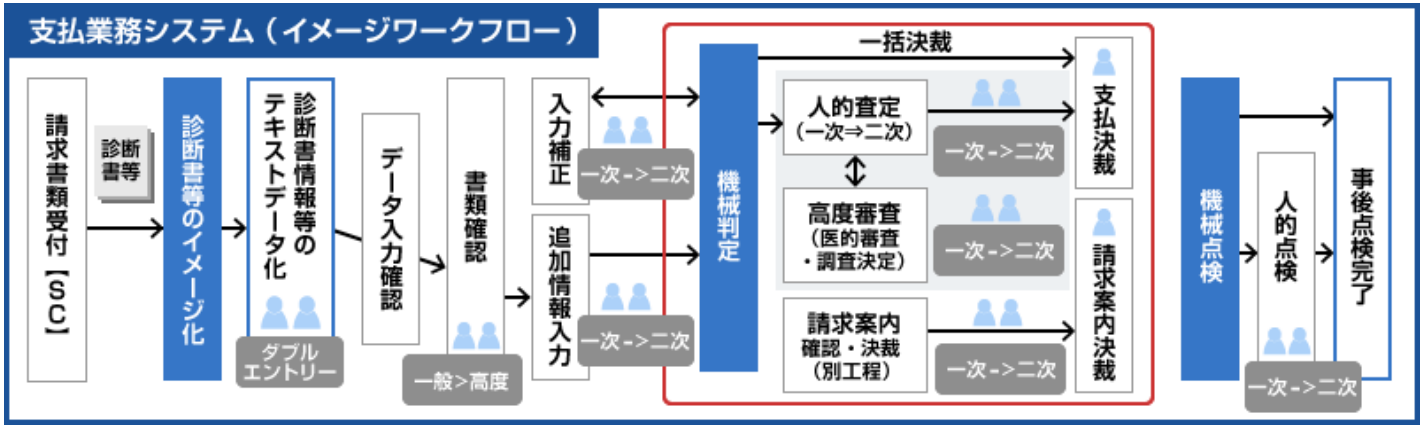
Watsonの適用：

- p53の活性化と不活性化を導くタンパク質を予測するため、p53に関する7万もの科学論文を分析
- この自動分析によって、ベイラー医科大学のがん研究者は、新たな研究対象となり得る6つのタンパク質を特定

Watsonが学習する専門家の判断

～ Watson技術を活用した保険金支払審査業務

かんぽ生命は、コグニティブ・コンピューティング「IBM Watson」を導入、国内保険会社としては初めて、保険金支払審査業務で高度な判断が必要とされる人的査定の支援に役立てようとしている。過去の査定事例に含まれるビッグデータをWatsonに学習させることで、10年近い経験が必要だった難易度の高い査定業務を比較的経験の浅い社員でも実施可能にするとともに、査定品質の向上や生産性の向上を図るのが目的だ。



日本IBM提供日経PR記事より
http://ps.nikkei.co.jp/ibmwatson1603/p2.html

Watson とは

ところで、Watsonとはなんなのか?
実はいろいろな意味のWatsonがあります。

クイズ番組に勝ったWatson:
数百人の研究者チームが一つの目的のため何年もかけて開発した特注システム。

「先進的な」Watson:
"Jeopardy!"ほどではないが、新規ソリューション開拓を目的に戦略的に構築されたシステムが多い。事例紹介にでてくるものは今のところこのパターンが多い。

「普通の」Watson:
商用の製品・サービスを組み合わせてつくるソリューション。
上記に比べて簡単に構築できるが、機能的な制約は多い。
現段階で本番運用されているのは、ほとんどが顧客接点系。

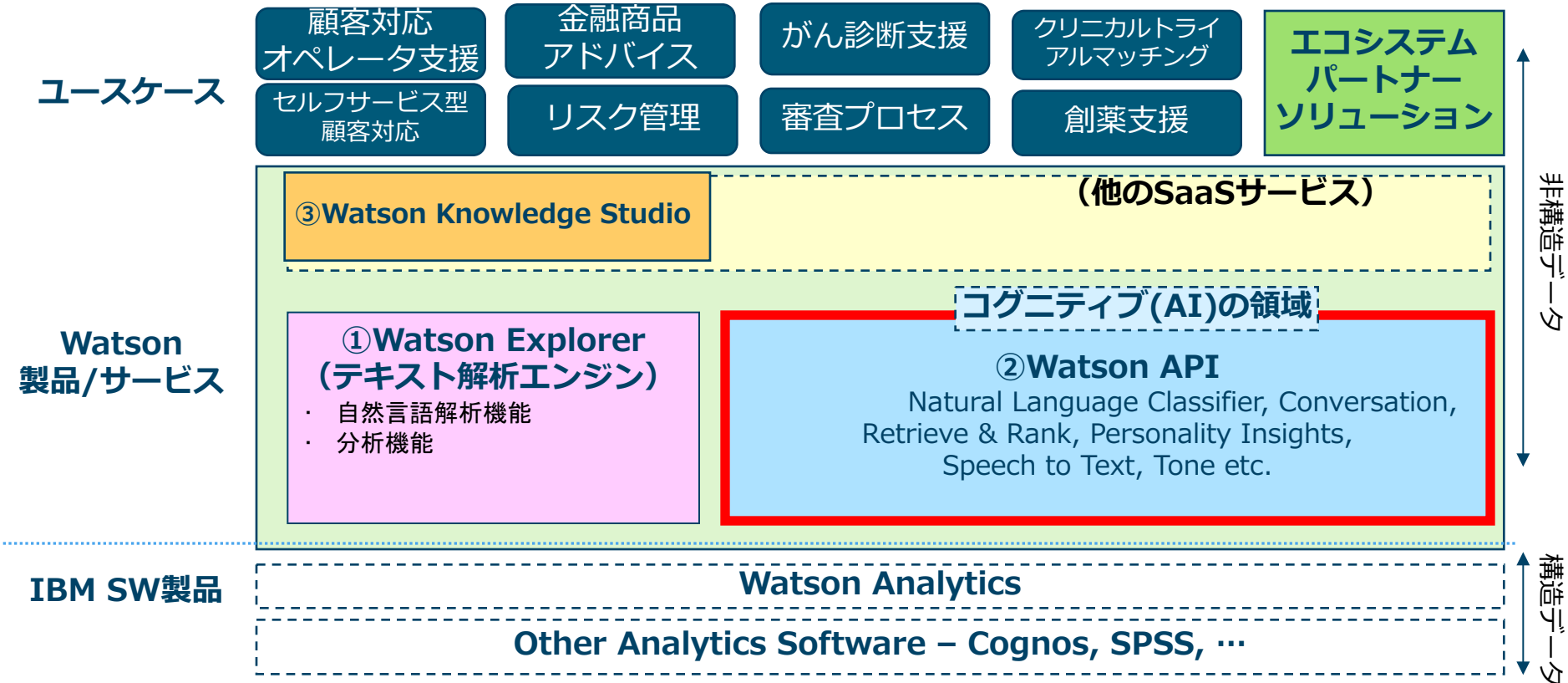
次ページ以降で紹介するのは「普通の」Watsonを構築するためのサービス、製品ということになります。

Watsonの構成要素と製品ファミリー

Watson製品ファミリー

製品・サービスの形態として大きく次の３つに分類される。本日はこのうち②のAPIサービスを中心に紹介します。

① オンプレミス製品 (Watson Explorer)
②クラウド上のAPI サービス(Natural Language Classifier等)
③クラウド上のSaaSサービス (Watson Knowledge Studio)



Watson API紹介

- NLC (Natural Language Classifier)
- R&R (Retrieve and Rank)
- VR (Visual Recognition)
- Discovery

IBM Watson Watson Developer Cloudで提供のAPIサービス

Watson Developer Cloudで提供しているAPIサービスは日々進化しています。
2017年3月18日現在の提供サービスは以下の通りとなります。

言語系

Natural Language Classifier
テキスト文章の分類を行う(質問の意図推定など)

Retrieve and Rank
自然言語の質問に対して、回答の候補を返す

Conversation
アプリケーションに自然言語インターフェースを追加してエンドユーザとのやり取りを自動化

Document Conversion
文書を新しい形式に変換する

Personality Insights
テキストから筆者の性格を推定する

Natural Language Understanding(日本語未対応)
自然言語処理を通じてキーワード抽出、エンティティー抽出、心情分析、感情分析、概念タグ付け、関係抽出、分類法種別、作成者抽出などを行う

Tone Analyzer(日本語未対応)
テキストの感情、社交性、文体を解析する

Language Translator(一部日本語未対応) ※1
自然言語テキストについて翻訳対象の言語へ翻訳を行う

画像系

Visual Recognition
画像コンテンツに含まれる意味を検出する

音声系

Speech to Text
音声をテキスト文章に変換する

Text to Speech
テキスト文章を音声に変換する

分析系

Discovery(日本語未対応)
認知検索およびコンテンツ分析エンジンをアプリケーションに追加して、優れた意思決定を行うのに役立つパターン、傾向、およびアクション可能な洞察を識別する。

Tradeoff Analytics(日本語未対応)
複数の競合する選択肢の中から、選択を行う過程を支援する

本日紹介するAPI

https://www.ibm.com/watson/developercloud/services-catalog.html
※1 Language Translatorに関してはニュースドメインのみ日本語対応をしています。

NLC (Natural Language Classifier)

NLCとは:

質問やテキストに含まれるひとつまたは複数の意図を判別する機能

内部の実装としてdeep learningを利用

意図とは

質問や発言から言語に含まれるノイズを取り除いて、それが何を言おうとしているか、あるいは何を聞こうとしているのかということ。

例:

「銀行口座はどうやって開けますか？」
「銀行で口座を開くために必要なものを教えてください。」

はどちらも**「銀行口座開設方法」**という**「意図」**を聞くための質問文のバリエーション

NLCでの学習とは

「銀行口座開設方法」を意図とする異なる複数の質問文(最低10個程度)を同じ意図のバリエーションとして学習させる。

-> 学習の結果、**質問文と似た表現の質問文に対しても同じ「意図」を持つ質問文であると解釈**できるようになる。

-> FAQと呼ばれるよくある質問に対して適切な回答を見つける仕組みを実現するのに適した仕組み

IBM Watson NLC (Natural Language Classifier)

(学習フェーズ)

質問文 **学習** **意図**

銀行口座はどうやって開けますか

銀行で口座を開くために必要なものを教えてください。

銀行口座開設方法

一つの「意図」に対して、表現の異なる複数の「質問文」(最低5つ、20程度が望ましい)を用意し、学習させる

住所変更はどうやって行いますか

転居したのですが、どうしたらいいですか

住所変更手続き

(運用フェーズ)

質問(入力)

学習させた質問文と似た表現の質問は同じ「意図」と判断するようになる

銀行口座を開きたいのですが

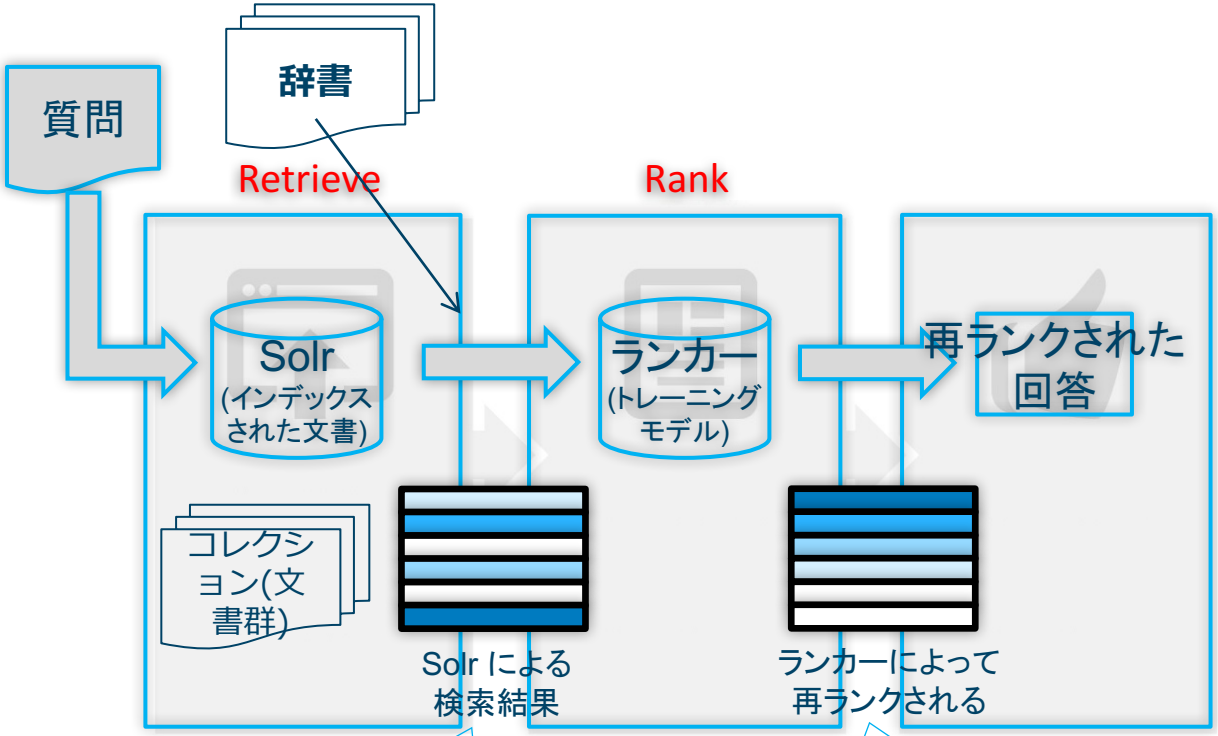
NLC問い合わせ結果(出力)

意図候補	確信度
銀行口座開設方法	0.95
住所変更手続き	0.10
:	
:	

R&R (Retrieve and Rank)

R&R とは:

複数ある検索結果の優先順位を学習によって変更するしくみ。



この時点では、SolrというOSSの検索エンジンによる検索結果が単純に返る。
表示順は、検索キーワードの頻度に基づくもので、重要性の順とはなっていない。

ランカーと呼ばれる機械学習モデルに検索結果の優先順位を学習させる。学習を続けると、重要な順番に表示されるようになる。

IBM Watson R&R (Retrieve and Rank)

実行サンプル

質問文:**明治維新の志士たちにちなんだ名所を教えてください**

回答結果例:

ID	Title	Body
692	親鸞聖人御旧跡きらら坂	京都の観光名所の一つです。修学院離宮の脇より比叡山の山頂に至る古道で、親鸞上人も参拝に利用したと伝えられています。(中略)
760	新島旧邸	同志社の創立し、幕末は志士としても活躍した新島襄の旧邸宅です。（中略）
897	維新の道	坂本龍馬をはじめ、維新の志士たち549人が霊山に祀られています。（中略）

学習方法:

「グランドトゥルース」と呼ばれる、質問文と回答の関連度をCSVで表現したファイル(関連度ファイル)を用意し、APIでサーバーにアップロードすることで学習させます。

(例)

明治維新の志士たちにちなんだ名所を教えてください, 897, 3, 760, 2, 692, 0

IBM Watson Visual Recognitionの提供機能

- Visual Recognitionは画像コンテンツに含まれる意味を検出します。
- このAPIは下記の詳細機能を提供しています。

Image Tagging

一般種別
画像の特徴を検知し、タグとして抽出しクラス・キーワード(犬、山などの一般名詞)を生成します。この機能は事前学習済みであり、生成されるキーワードは他ユーザーと共通のものとなります。

Facial Detection

顔検出
イメージ内の人物の顔を検出します。 また、顔の一般的な年齢層と性別も示されます。有名人に関してはその名称も検出することが可能です。
この機能は事前学習済みであり、検出結果は他のユーザーと共通のものとなります。

Visual Learning

画像トレーニング
カスタム画像種別を行います。ユーザーが識別を行いたいクラスのイメージを事前学習させます。識別結果は確信度と共に返されます。

Similarity Search

類似イメージ検索 (ベータ)
事前準備としてイメージのコレクションをアップロードした後、検索をかけると視覚的に類似したイメージを検出します。

Link Extraction

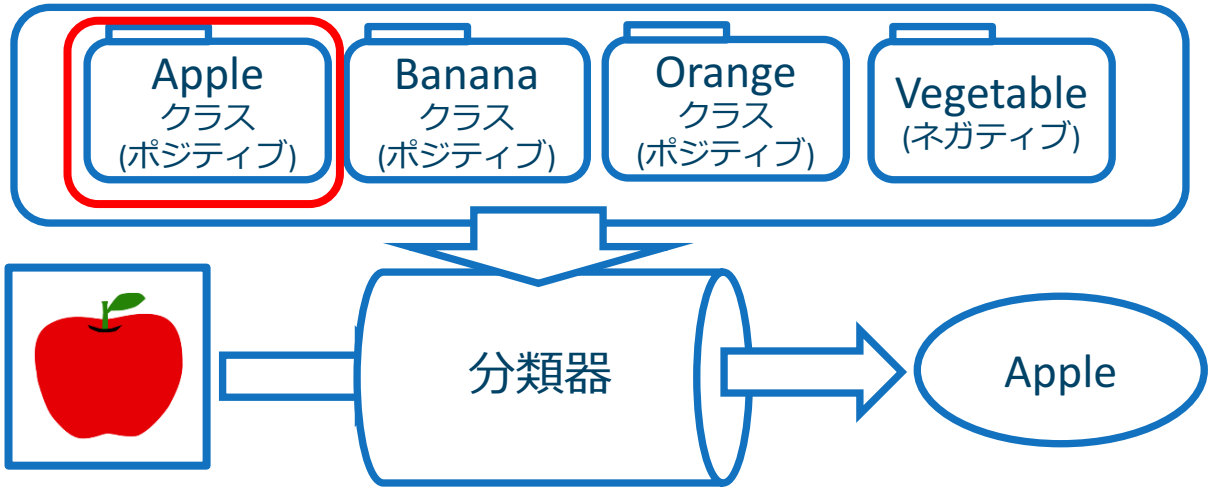
Web画像抽出
指定されたURL先のWebページの内容を分析し、ページ内容に一番関連性の高い画像を抽出します。

IBM Watson 画像トレーニング (Visual Learning)

「画像トレーニング」について

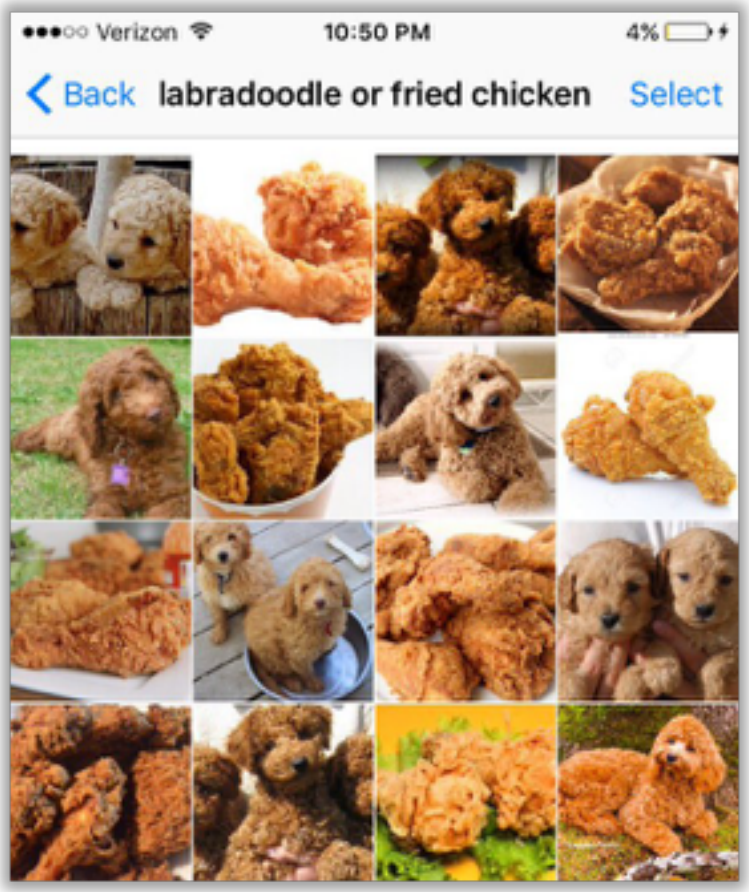
内部の実装としてdeep learningを利用

- **目的**
  - Visual Recognition による事前分類やフィルタリングをすることで、業務における様々な確認や分析作業を行う人間の負荷低減を行います。
    - 不適切な画像のフィルタリング
    - 画像選択での事前分類
    - 商品や製作物の品質確認
- **トレーニングデータ**
  - 分類すべきクラスごとに集められた最低10枚の画像
    （適切な分類品質を得るには50枚以上、理想的には200枚）
  - どのクラスにも該当しないネガティブな画像
    (2つ以上のクラスがある場合はなくてもいい)

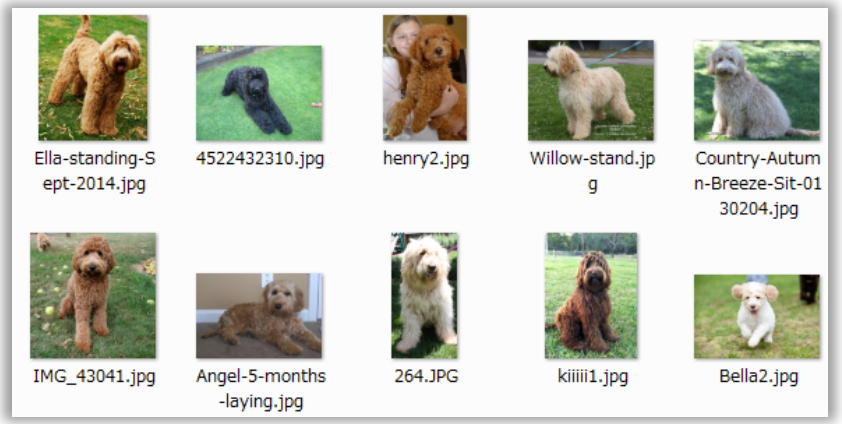


IBM Watson 画像トレーニング (Visual Learning)

画像トレーニングの利用サンプル
ネットで有名になった**「ラブラドールと見分けの付かないフライドチキン」**をVRで見分ける。



学習データ
ネット上で集めたラブラドールとフライドチキンの写真約200枚ずつ。



IBM Watson 画像トレーニング (Visual Learning)

検証結果　全問正解

	犬1	犬2	犬3	犬4	犬5	犬6	犬7	犬8
								
「犬」確信度	0.5549	0.4164	0.5309	0.4468	0.5064	0.5337	0.4296	0.3918
「チキン」確信度	0.0493	0.0833	0.0536	0.0737	0.0594	0.0536	0.0783	0.0909

	チキン1	チキン2	チキン3	チキン4	チキン5	チキン6	チキン7	チキン8
								
「犬」確信度	0.0463	0.0410	0.0474	0.0461	0.0449	0.0443	0.0527	0.0601
「チキン」確信度	0.5690	0.6002	0.5633	0.5703	0.5775	0.5801	0.5343	0.5019

参考リンク: http://qiita.com/VegaSato/items/863b614cd5ab88cf2d44

IBM Watson Discovery

Discoveryはテキスト分析のSaaS/APIプラットフォームとして、これからIBMが推進しようとしているサービスです。
下記の機能を一つのAPIですべて持っていることが最大の特徴です。

データ取り込み機能: HTML / PDF / WORD / JSONに対応。
機能的に従来のDocument Conversionと同等です。

エンリッチ機能: 取り込んだ文書に対してNLU(Natural Language Understanding)によるタグ付けを行います。追加可能項目は、Entity, Conceptsなど最大7項目です。

ストレージ機能：データはクラウド上にINDEXとして保存されます。

検索機能：エンリッチ機能で付加された情報を含め、データの検索を行うことができます。
内部で動いている検索エンジンは**OSSのElasticsearch**です。

IBM Watson Discovery Querysサンプル(1)

(ユースケース)
「IBMがXXX社を買収する」というNews記事中の文章をすべて収集する
(検索パラメータ)
aggregation=nested(relations).filter(relations.action.text:acquire,relations.subject.text:IBM)
.term(relations.sentence)


```
                    "key": " IBM Security,
  headquartered in Cambridge, Mass., also announced
  its intent to acquire Ravy Technologies, which
  OEMs dashboards used in the Agile 3 product
  line.",
                    "matching_results": 6
                  },
                  {
                    "key": " IBM previously
  splashed out about $2 billion in 2013 to acquire
  SoftLayer for its cloud infrastructure business,
  and today the company highlights growth in cloud
  services as a counterpoint to financial declines
  in its more traditional business units.",
                    "matching_results": 6
                  },
                  {
                    "key": " IBM has revealed plans
  to acquire Agile 3 Solutions to give clients
  tools and a platform to understand how
  cyberattacks can damage their businesses and to
  better protect their companies through risk
  management.",
                    "matching_results": 4
                  },
                  {
                    "key": " In addition, IBM
  intends to acquire Ravy Technologies, a
  subcontractor of Agile 3 Solutions.",
                    "matching_results": 4
```


Discoveryは検索用のQuery言語を持っており、これを使うことで複雑な検索も一回のAPI呼び出しで行うことが可能です。

この例では、検索対象コーパスに、IBMが標準で提供しているDiscoveryのコレクションである**「Watson News」**を利用しています

Discovery Queryサンプル(2)

(ユースケース)
2017年1月1日から2017年1月31日までの間のAirbnbに関する記事という条件で検索し、
その記事の評判分析を日単位で行う。

(検索パラメータ)
query= text:Airbnb
count=0
aggregation=term(docSentiment.type).timeslice(blekko.chrondate,1day)
filter=blekko.chrondate>1483196400,blekko.chrondate<1485788400

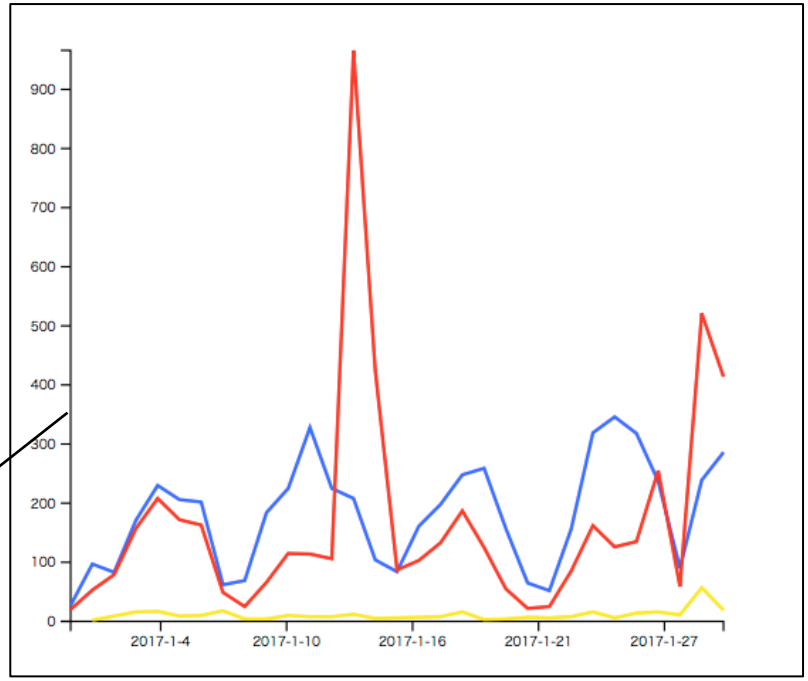
aggregation機能は複数の関数を連続して利用可能

期間をFROM-TOで絞り込む場合、chrondateを利用する


```
{
  "matching_results": 12474,
  "results": [],
  "aggregations": [
    {
      "type": "term",
      "field": "docSentiment.type",
      "results": [
        {
          "key": "positive",
          "matching_results": 5636,
          "aggregations": [
            {
              "type": "timeslice",
              "field":
"blekko.chrondate",
              "interval": "1d",
              "results": [
                {
                  "key_as_string":
"1483142400",
                  "key": 1483142400000,
                  "matching_results": 27
                },
                {
                  "key_as_string":
"1483228800",
                  "key": 1483228800000,
                  "matching_results": 97
```


Queryの結果

グラフ化の結果
(D3.jsを利用)



第二部 WatsonとOSSの関わり

Watson開発環境としてのOSS

Watson API構成要素としてのOSS
- R&RとSolrの関係
- DiscoveryとElasticSearchの関係

WatsonとOSS　機能レベルの比較例
- VRとTensorflowサンプルアプリの比較

Watson開発環境としてのOSS

- GitHub
- Watson Developer Cloud
- (おまけ)Kibanaの利用

Watson開発環境としてのOSS

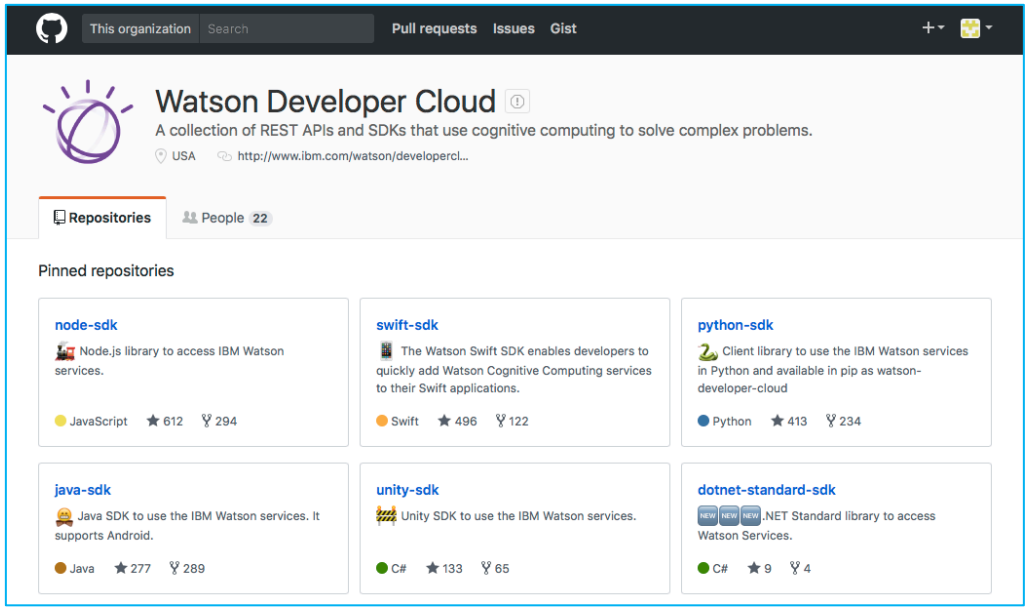
GitHub

GitHubはソースコード管理 + チーム開発支援機能を持った統合環境です。
OSSのリポジトリとして非常に有名です。
IBMは2016年にGithubとの戦略的提携を発表しています。

Watson Developers Cloud

Watson APIを使ったアプリ開発を支援する環境です。
ライブラリ・サンプルアプリリポジトリはGitHub上に構築されています。
Bluemixと連携することで、効率のいい、開発・テスト・実行環境を構築可能です。

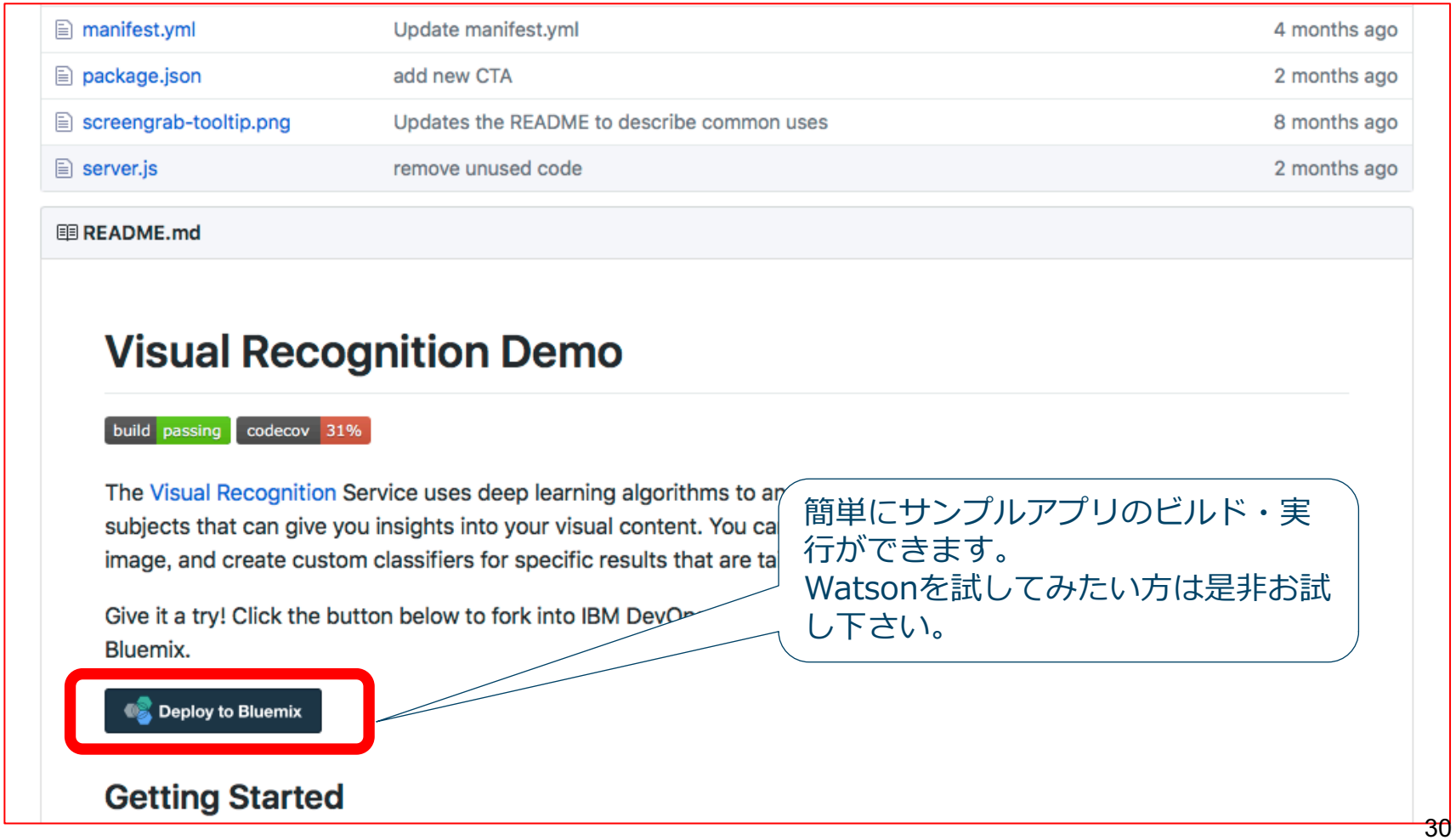
https://github.com/watson-developer-cloud/



以前に比べてIBMの開発系のマインドがオープン志向になった一つの象徴

Watson開発環境としてのOSS

サンプルアプリのリポジトリのREADME.mdファイル例を下に示します。
Bluemixのアカウントを持ったユーザーがこのボタンをクリックすると、Bluemixに向けてdeployが始まり、数分後には利用可能なサービスまで起動される状態になります。



Watson開発環境としてのOSS

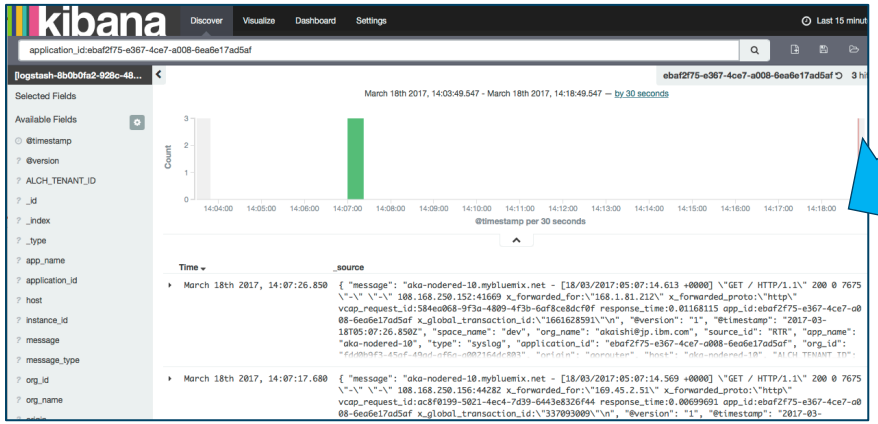
(おまけ)Kibanaの利用

Watson API呼び出し用のアプリケーションサーバーをBluemix上に作った場合、そのサーバーのログ分析にKibanaを利用できる形になっています。

ダッシュボード左のメニューから「ログ」をクリックします

画面右の「詳細ビュー」をクリックします

Kibanaが自動的に起動します



Watson API構成要素としてのOSS

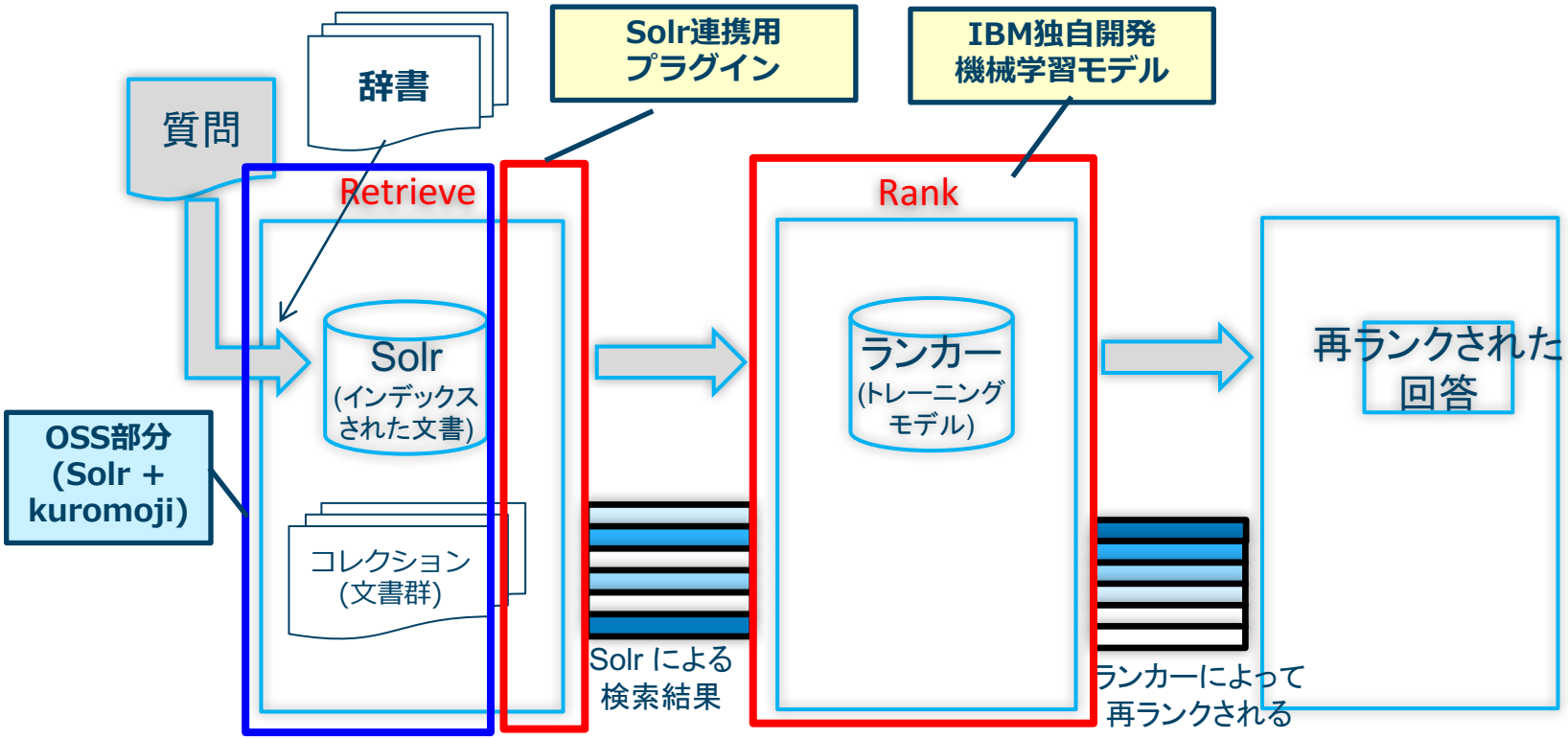
R&RとSolrの関係

DiscoveryとElasticSearchの関係

・R&RとSolrの関係

Watson APIの一つであるR&R (Retrieve and Rank)ではOSSの検索エンジンであるSolrがAPIの構成要素として使われています。

-> 実はR&Rは簡単に構築可能なSolr SaaSとして利用することも可能です。
(基盤的な冗長構成もAPIが裏でやってくれています)



・R&RとSolrの関係

R&RのAPIでSolr Index作成時に必要な構成ファイルの一部を以下に示します。
日本語形態素解析には**Solr**にバンドルされているOSSの**kuromoji**が使われています。

Solr連携用プラグイン

形態素解析実装はkuromoji

```
<fieldType name="watson_text_ja" indexed="true" stored="true"
class="com.ibm.watson.hector.plugins.fieldtype.WatsonTextField">

    <analyzer>

      <tokenizer class="solr.JapaneseTokenizerFactory" mode="search"/>
      <filter class="solr.JapaneseBaseFormFilterFactory"/>
      <filter class="solr.JapanesePartOfSpeechStopFilterFactory" tags="lang/stoptags_ja.txt"/>
      <filter class="solr.CJKWidthFilterFactory"/>
      <filter class="solr.StopFilterFactory" words="lang/stopwords_ja.txt" ignoreCase="true"/>
      <filter class="solr.JapaneseKatakanaStemFilterFactory" minimumLength="4"/>
      <filter class="solr.LowerCaseFilterFactory"/>

    </analyzer>

</fieldType>
```

・R&RとSolrの関係 (APIリファレンス記載の標準検索パターン)

Rankerなしに素のSolrを呼び出す方法と、Ranker付きで呼び出す方法の2通りがあります。
１つのSolr Indexに複数のRankerを持たせることも可能です。

Rankerなしに素のSolrを利用(表示順はSolrの結果と同一)

```
${base_url}/solr_clusters/${cluster_id}/solr/${collection_name}/select?¥
q='検索文字列'...
```

Rankerによる順付けを行う場合(機械学習の結果を含めて表示順が決まる)

```
${base_url}/solr_clusters/${cluster_id}/solr/${collection_name}/fselect?¥
ranker_id=${ranker_id},q='検索文字列'...
```

・R&RとSolrの関係 (テストで判明した拡張検索パターン)

Solrで元々持っているMLT(MoreLikeThis 類似検索)も工夫すると使えます。
ただし、この場合Rankerとの連携はできなくなります。
MoreLikeThis Handler呼び出しを行う場合、R&R用標準の構成ファイルに修正が必要です。

MoreLikeThis Componentの呼び出し

```
${base_url}/solr_clusters/${cluster_id}/solr/${collection_name}/select?¥
mlt=true,q='検索文字列'...
```

MoreLikeThis Handlerの呼び出し

```
${base_url}/solr_clusters/${cluster_id}/solr/${collection_name}/mlt?¥
q='検索文字列',mlt.mintf=2,mlt.mindf=5...
```

後者の呼び出しを行う場合に必要な構成ファイルの追加定義。

```
<!-- Added by this project for mlt -->
<requestHandler name="/mlt" class="solr.MoreLikeThisHandler">
</requestHandler>
```

※R&R を Solrとして使う場合のより詳細な情報については下記リンク先を参考とされて下さい。
https://www.ibm.com/watson/developercloud/doc/retrieve-rank/solr_ops.shtml#incompatible

・DiscoveryとElasticsearchの関係

R&RではAPI内部のインデックス・検索機能の実装が**Solr**だったのに対して、新しいAPIである**Discovery**ではその内部で**Elasticsearch**を利用しています。

Discovery APIでqueryにより検索をかけると結果セットはそれぞれscore を含んでいて、これが検索文との適合度を示しています。
結果セットはsocreの高い順に返されるので、結果的に「より検索文に合致した」文書から順に表示されることになります。
この**scoreの実装はOSSであるElasticsearchにより行われています**。
(Index自体もElasticsearchのもの)

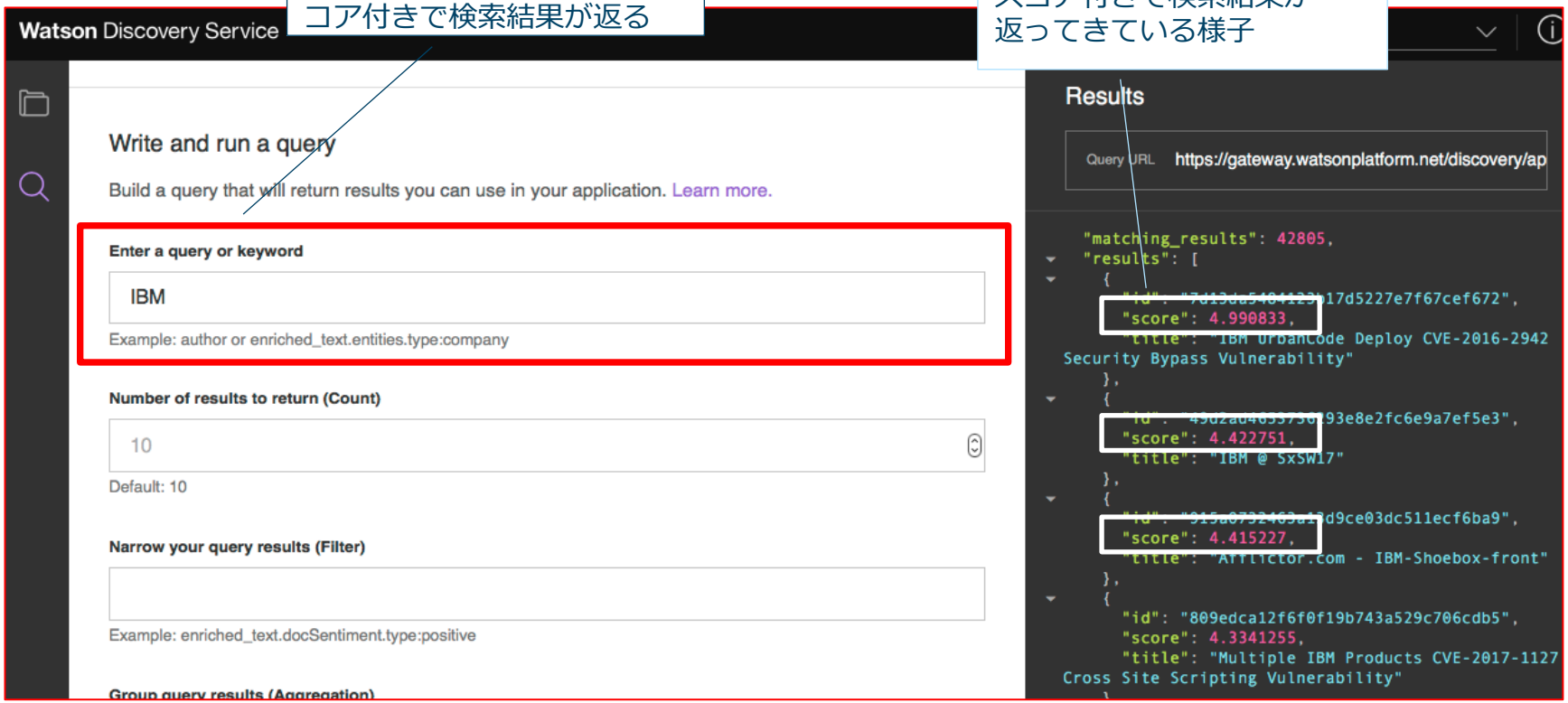
一方、Discoveryの検索のもう一つの大きな特徴であるaggregation関数に関してはDiscovery(IBM)独自の実装となってます。
DiscoveryはOSSのElasticsearchの特徴をうまく活用したAPIであるということができます。

・DiscoveryとElasticsearchの関係

Discoveryでの検索サンプル
下の図はWatson Newsに対してDiscoveryのquery機能で検索をした結果です。
Elasticsearchで求められたscoreの高い順に結果が返されていることがわかります。
同じキーワードをfilter機能で検索すると、結果件数は同じですが、scoreの値はすべて0になります。

query検索引数入力画面
ここに検索条件を入れるとスコア付きで検索結果が返る

スコア付きで検索結果が返ってきている様子



WatsonとOSS　機能レベルの比較例

- VRとTensorflowサンプルアプリの比較

VRとTensorflowサンプルアプリの比較

資料作成のきっかけ

「Watson APIのVRとTensorflowはどちらがいいのか」という質問をするIBM社員・お客様が多いことに気付きました。
そもそも、この２つを比較するのは**「業務パッケージとWEB frameworkのstutus2のどちらがいいか」**を比較するのと同じくらい意味がないことなのですが。。。
ということで、社員向けにVRとTensorflowの立ち位置の違いを説明するとともに、相手がTensorflowでなく**Tensorflowサンプルアプリ**なら比較可能ではあるので、そのような比較テスト結果も付けた資料を作成しました。

IBM Watson VRとTensorflowサンプルプログラムとの比較

VRの画像トレーニング機能は、機能的にはGoogleで提供しているDeep Learning用フレームワークTensorflow付属のサンプルプログラムと似た点がありますが、サービスのレベルは相当異なっています。以下にその違いを整理しました。

	Visual Recognition 画層トレーニング	Tensorlflow サンプル (full_connected_feed.py)
ニューラルネットへの自由度	内部処理はすべて隠蔽されていて、学習データ以外に外部から調整できる余地はない。	ニューラルネットの組み方、重み付け関数のパラメータなどすべて調整可能。(サンプルはあくまで最低限の動作をするプログラムという位置づけ)
学習用の入力データ	一般的なjpegまたはpngファイルを、クラス毎にzipで固めたもの。 イメージサイズは320x320が理想的だがそうでなくてもいい。	機械学習しやすいよう、イメージデータからヘッダなど除いて1次元配列データに展開したもの。サンプルプログラムのため、解像度は28x28で固定。
学習にかかる手間	画像の収集 ->クラスことにzipに固める ->API呼び出し で終わるので非常に簡単。	実際のイメージファイルを入力とする場合、機械モデルの入力となる1次元配列データを作るため事前準備としての加工が必要。
トレーニングに必要なイメージ数※	1クラスあたり最低50枚、理想的には200枚程度。これ以上多くなってもあまり学習には効果がない。	サンプルアプリは1クラスあたり5000枚(全体で50000枚)を利用している。ある程度の認識性能を出すためにはこの程度の枚数が必要と考えられる。

※「トレーニングに必要なイメージ数」については、２つの方式で実際にテストし結果を比較しました。

VRとTensorflowサンプルプログラムとの比較

ベンチマークテスト

VRおよびTensorflowサンプルアプリでの学習曲線を条件をそろえてテストしました。
VRに関しては最新の2016-05-20版、Tensorflowに関してはv1.0を利用しています。

テスト方法:

教師データ:
http://yann.lecun.com/exdb/mnist/に公開されている手書き文字(0から9の数字)サンプルデータを利用しました。
0から9までの数字はランダムに出てくるので、例えば先頭から500件のデータを取ると1クラスあたりそれぞれ約50枚のイメージが対応する形になります。
全体で60000件分ありますが、先頭からN件のみを教師データとするテストをNを変えて行いました。
VR用には、このデータに逆変換をかけて、pngファイルとしたものを利用しました。

確認データ：
http://yann.lecun.com/exdb/mnist/に公開されている手書き文字(数字)サンプルデータ。データの形式は教師データと同様です。
全体で10000件分ありますが、条件を合わせるため先頭から100件のみを検証データとしました。
Tensorflowに関しては、テストの試行のたびに正解数にばらつきがあったので5回テストを行い平均値を取りました。

ベンチマークテスト結果

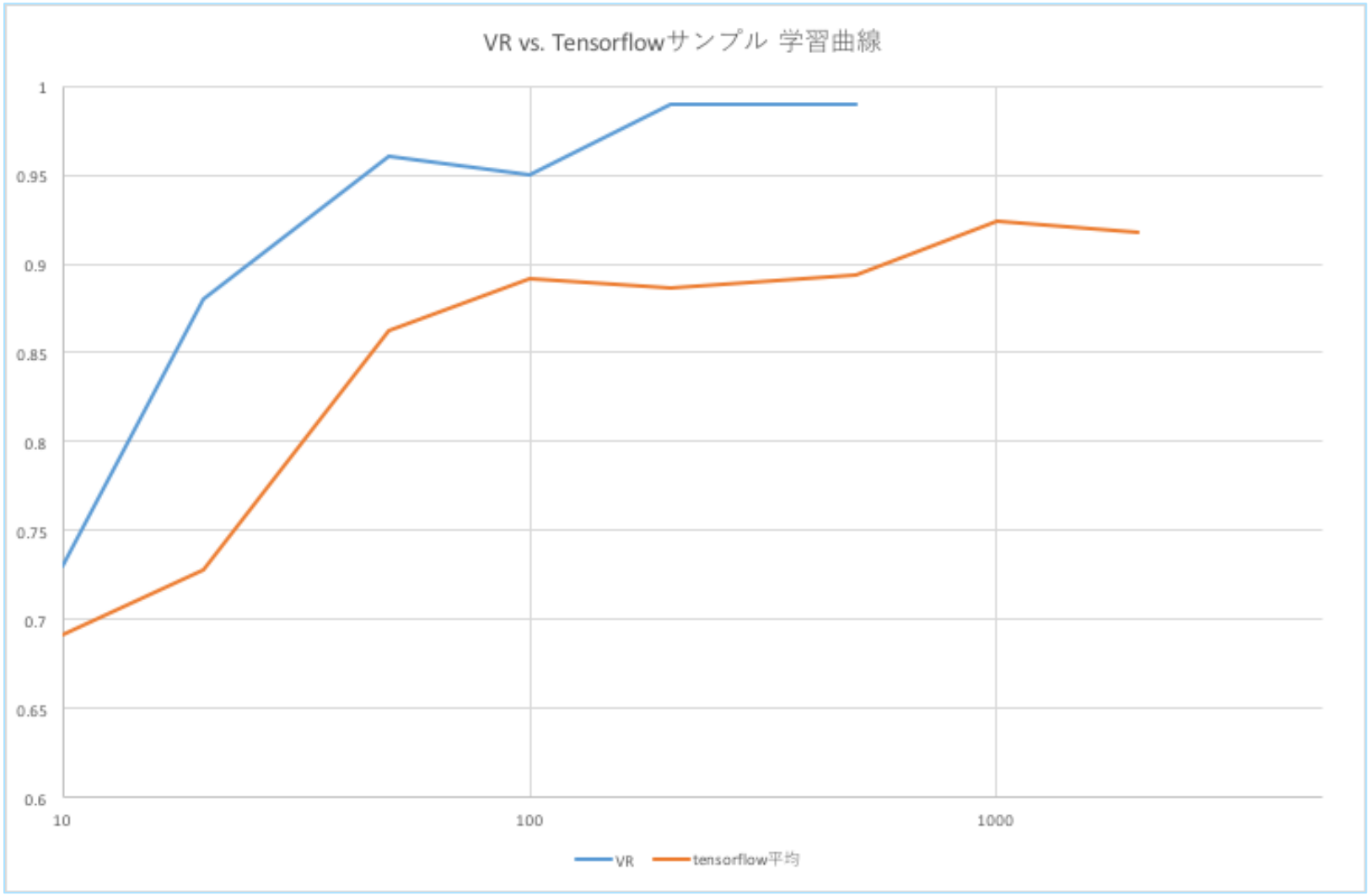
結論としてVisual Recognitionの方が少ない枚数で認識可能であることがわかりました。ある程度の品質を担保するのに1クラスあたり50枚(全体件数に換算すると500枚)、理想的に200枚(全体件数換算で2000枚)というVisual Recognitionの学習ガイドの妥当性も確認できました。

1クラス当たり学習データ数	VR	tensor flow サンプル	1回目	2回目	3回目	4回目	5回目
10	**73%**	**69.2%**	70%	68%	64%	72%	72%
20	**88%**	**72.8%**	74%	74%	72%	71%	73%
50	**96%**	**86.2%**	86%	86%	86%	87%	86%
100	**95%**	**89.2%**	89%	88%	90%	92%	87%
200	**99%**	**88.6%**	88%	89%	89%	88%	89%
500		**89.4%**	91%	88%	90%	88%	90%
1000		**92.4%**	93%	92%	93%	92%	92%

VRとTensorflowサンプルプログラムとの比較

ベンチマークテスト結果

前ページの結果をグラフで示したものです。



VRとTensorflowサンプルプログラムとの比較

Visual Recognitionが少ない学習データで高い認識精度を出せる理由について

(参照元)
https://developer.ibm.com/answers/questions/316140/visual-recognition-capabilities/

(日本語訳抜粋)
Visual Recognitionの画像トレーニング機能はdeep learningの仕組みを使っていますが、**新しい分類器を作った時点で数百万のサンプルデータによる事前学習データが活用されています。**
deep learningの中心的な概念は、特定のクラスを他のクラスと分類するのにどのような特徴量を利用するかです。
このような特徴量は、色、形、質感(texture)などで明示的に表されるものではなく、トレーニングにより学習される機械ネットワークの数千万のパラメータによって暗黙に表現されます。Visual Recognitionは、色、形、質感(texture)などの特徴量を組み合わせた形によりクラス間の識別を行っているということができます。

->事前学習と合っていない種類の画像の識別ではうまくいかない可能性も考えられます。

IBM Watson VRとTensorflowサンプルプログラムとの比較

比較結果まとめ

今回の比較はWatson APIで提供されている機能と、TensorFlowなどのフレームワークを利用したカスタムAIとの比較に一般化することも可能です。
そういう観点で両者の長所・短所を比較すると次のようになります。
それぞれの特徴を理解して適材適所で対応することが重要です。

	Watson API	カスタムAI
コスト	1 Callあたりの従量料金なので高い △	1度開発してしまえば従量料金はかからない ○
適用範囲	限定されている △	機械学習が適用可能な領域に対してはどこでも可能 ○
利用しやすさ	すぐに使える ○	開発工数がかかる △
学習工数	VRのように事前学習されたモデルでは少ない学習工数となる場合がある ○	基本的に多くの学習工数が必要(工夫次第ではそうでないこともある) △
チューニング余地	完全なブラックボックスで一切できない △	ニューラルネットワークの組み方、パラメータなどいくらでも可能 ○

End of File