東京理科大学 大学院講義

第一回 情報検索特論（イントロダクション）

東京理科大学　創域理工学部　情報計算科学科
植松 幸生

第一回のアジェンダ

1. シラバスの確認
2. 教員紹介
3. 講義の特徴
4. 情報検索とは？
5. この講義で何を学べるか？(具体的な説明)
6. 情報検索基礎
   Boolean検索
7. 次週に向けて

本授業の目標とシラバスの確認

目標：情報検索/LLMの概念を理解し，実装レベルまで習得する

1 講義概要　講義概要の説明
2 特別講義　Federated Learning: Mei Kobayashi(5限なので，出れない方は応相談)
3 情報検索基礎　スコアリング1　TFIDF等のscoring技術の解説
4 情報検索基礎　インデックス1　全文検索の仕組みを理解する
5 情報検索基礎　インデックス2　インデックスの圧縮技術を学ぶ
6 情報検索応用　LLMを用いた情報検索　LLMを用いた情報検索の仕組みを理解する
7 情報検索応用　LLMを用いた情報検索２　第６回の授業の実装を学ぶ/RAGの実装を学ぶ
8 情報検索応用　LLMを用いた情報検索4　LLMのファインチューニングについて学ぶ
9 情報検索応用　応用例　実データを用いたEDAを学ぶ((Exploratory Data Analysis)
10 情報検索応用　応用例 実データを用いた情報検索技術の応用例
11 情報検索実践　情報検索とLLMを応用したシステムの開発１　自分で考えたシステムを実装するための方法を理解する
12 情報検索実践　情報検索とLLMを応用したシステムの開発2　システムを実装する/プレゼンテーションやエレベータピッチの方法を学ぶ
13 情報検索実践　プレゼンテーション1　実装したシステムをデモを通じたプレゼンテーションを行う
14 情報検索実践　プレゼンテーション2　引き続きプレゼンテーションを行う．また，他の学生の発表を評価する
15 情報検索実践　情報検索最新動向（講演）

来週の講義について

来週は通常の授業(2限)は実施しません.

代わりに5限に右記の先生にご講演頂きます.

連合学習(Federated Learningに関して知る良い機会になりますのでご参加ください)

時間が変わるため，出れない都合が悪いという学生は個別に植松まで連絡してください.

2025 年 9 月 11 日

情報科学コロキューム 2 のご案内

下記日程で開催します。

9 月 24 日（水） 5 限

場所：K203 教室

講師　Mei Kobayashi 先生（Eaglys 株式会社）

表題　Federated Learning—An Overview Tutorial

概要：

Federated learning is the study of methods to enable multiple parties to collaboratively train machine learning or AI models, while each party retains its own raw data on-premise, never sharing it with others. The talk begins with the motivation for federated learning and the simplest type of neural network. Next, we introduce multiparty computation (MPC) and why enhancements are needed to provide security and privacy. We follow with a tutorial on edge computing, a distributed computing model in which data processing takes place on local devices, closer to where it is being generated. Advances in hardware and economies of scale have made it possible for edge computing devices to be embedded in everyday consumer products to process large volumes of data quickly and produce results in near real-time. Finally, we present federated learning, a framework that enables multiple parties to collaboratively train AI models, while each party retains control of its own raw data, never sharing it with others. Time permitting, we close with a discussion on attacks that target weaknesses of federated learning systems, e.g., data leakage (inferring raw data used to train an AI model by unauthorized parties) and data poisoning (a cyberattack that compromises data used to train an AI model to manipulate its output).

自己紹介：簡単なキャリア

NTT研究所→NTTコム→スタンフォード大学→NTTコム→ノキア→デジタル庁

Affiliations

Uni/Grad — NTT — 3rd — NTT — 6th — NTT Communications — 9th — Stanford — 11th — NTT Communications / NOKIA — 16th — デジタル庁 Digital Agency — 22nd

Professional experience

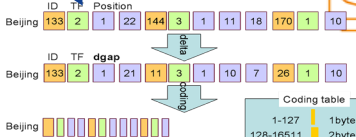
- Web image search
- Search engine R&D
- Recommender system / Distributed data processing
- Crowdsourcing Human computation
- Launched data science team
- Mobile network operation automation
- AI利活用/政府データ/ガイドライン策定

Current

goo 検索エンジン (画像検索)

東京理科大学 TOKYO UNIVERSITY OF SCIENCE

スマートレコメンド Smart Recommend



docomo d-menu検索エンジン

Stanford Lab

hadoop

Data Science Lab

各キャリアで，学術，ビジネスそして日本政府にも貢献してきています

幸運なことに，ほぼ全てのキャリアで**研究レベルのものを500万人以上に使ってもらえるソフトウェア**にして提供

要素技術
学術成果
ビジネス成果

Affiliations

Uni/Grad — NTT 3rd — NTT 6th — NTT Communications 9th — Stanford 11th — NTT Communications / NOKIA 16th — デジタル庁 Digital Agency 22nd

Professional experience

Web image search | Search engine R&D | Recommender system / Distributed data processing | Crowdsourcing Human computation | Launched data science team | Mobile network operation automation | AI利活用/政府データ/ガイドライン策定

単語の近接性に着目したインデックス

博士論文，ジャーナル，国際会議

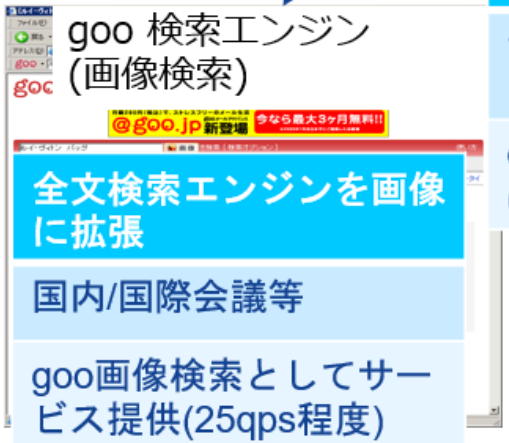
d-menu検索に適用(500qps以上!)

マルチモーダル異常検知

トップカンファレンス(AAAI)

社内向けシステムとして運用(副社長も利用)

goo 検索エンジン(画像検索)



全文検索エンジンを画像に拡張

国内/国際会議等

goo画像検索としてサービス提供(25qps程度)

パーソナライズレコメンド/大規模分散処理

研究会報告，オープンソーストップカンファレンス

551蓬莱，edion，gooニュース等に導入

異常検知と機械学習モデル管理

社内(Bell研)発表，国際会議等

国内外のオペレータに適用

本講義の特徴

- 講義資料
  - LETUS で配布（StanfordのCS276を踏襲しつつ，LLM時代に合わせて授業します）
  - 指定教科書：なしだが以下の2つの本を参考に最新の論文を参照しながら実施
    - https://mitmecsept.wordpress.com/wp-content/uploads/2018/05/stefan-bc3bcttcher-charles-l-a-clarke-gordon-v-cormack-information-retrieval-implementing-and-evaluating-search-engines-2010-mit.pdf
    - https://nlp.stanford.edu/IR-book/information-retrieval-book.html
- 成績
  - 第2回の特別講義のレポート（5限なので出れない人は応相談）
  - 第13,14回の発表
  - （加点） 講義の演習時に作成したプログラムを回収
  - （加点） 出席は，成績が悪い時の救済に利用しますが，必須ではありません
- 推奨事項
  - PC持参，プログラミング出来る環境を整える
  - Collaboratoryで可
- 講義形態
  - 講義＋ハンズオン

第一回のアジェンダ

1. シラバスの確認
2. 教員紹介
3. 講義の特徴
4. 情報検索とは？
5. この講義で何を学べるか？(具体的な説明)
6. 情報検索基礎
   Boolean検索
7. 次週に向けて

情報検索とは？

Stanford　NLPより

https://nlp.stanford.edu/IR-book/newslides.html

- **情報検索（Information Retrieval, IR）** とは、大規模なコレクション（通常はコンピュータに保存されているもの）の中から、情報要求を満たす非構造的な資料（通常はテキスト）を見つけ出すことである。
  今日では最初にウェブ検索を思い浮かべることが多いが、他にも以下のような多様な応用が存在する：
  - 電子メール検索
  - 自分のノートPC内の検索
  - 企業のナレッジベース検索
  - 法令検索

情報検索とは？(今風)

情報検索（Information Retrieval, IR） とは、大規模なコレクション（通常はコンピュータに保存されたテキストやマルチモーダルデータ）の中から、利用者の情報要求を満たす資料を見つけ出すプロセスである。

近年では、大規模言語モデル（LLM）の登場によって「検索」は単なる文書の取得にとどまらず、**検索結果をもとに生成・要約・推論を行う対話的な情報アクセス** へと拡張されつつある。

典型的な利用例としては以下が挙げられる：

- ウェブ検索（クエリに応じた文書の取得と要約提示）
- 電子メール検索やPC内検索（生成AIによる検索意図の解釈や自然文クエリ対応）
- 企業や組織のナレッジベース検索（RAG: Retrieval-Augmented Generation による回答生成）
- 法情報や医療情報の検索（検索結果の検証や根拠提示を伴うQA）

従来のIRは「文書を見つけること」が主眼だったが、現在のLLM時代のIRは「必要な情報を文書から取り出し、生成的に提示すること」まで含む

情報検索って必要なの？

- 情報検索ってLLMに聞けば良いからいらなくね？，そもそも検索なんてもうしなくね？

はい．あなたが情報検索/LLMの利用者であれば，この講義は正直必要ありません/とる必要もありません

情報検索はなぜ必要なのか？（7つの理由）

1. **知識の鮮度保証**
   LLMは静的に学習されるため、最新情報を扱うには検索による動的参照が不可欠。
   （例：ニュース、法令改正、社内最新資料）
2. **根拠付き応答の実現**
   ユーザに安心して提示できるのは「出典つきの回答」。
   → 検索で取得した一次情報を裏付けにすることで信頼性が向上。
3. **専門領域対応**
   医療・法律・工学などの専門分野では汎用LLMだけでは不十分。
   → 検索でドメイン特化文書を組み合わせる必要。
4. **運用コスト削減**
   すべての知識をLLMに再学習させるのは非効率・高コスト。
   → 検索で必要な部分だけ取り込む方が現実的。

情報検索はなぜ必要なのか？（7つの理由）

5. 内部知識・限定情報の活用

社内文書や非公開データは事前学習に含められない。

→ 検索基盤と連携して初めて活用可能。

6.説明責任・監査対応

行政・法務・医療システムでは「なぜその回答か」を説明する責任がある。

→ 検索結果を提示できる仕組みが必須。

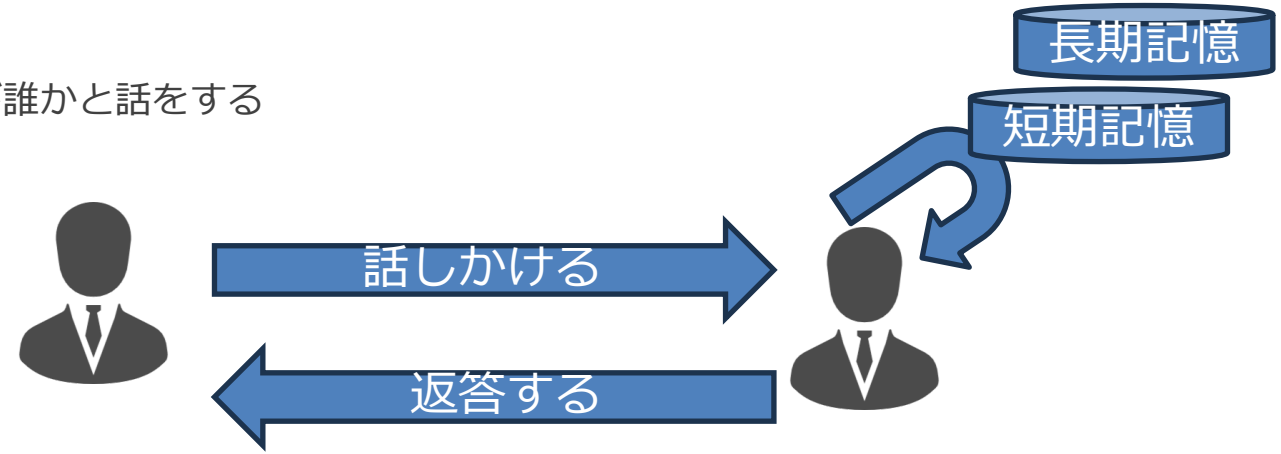
7.長文処理・効率的な情報アクセス

LLMのコンテキストはトークン上限があり、大規模文書を直接扱えない。

→ 検索で適切に切り出し、効率的に入力する必要。

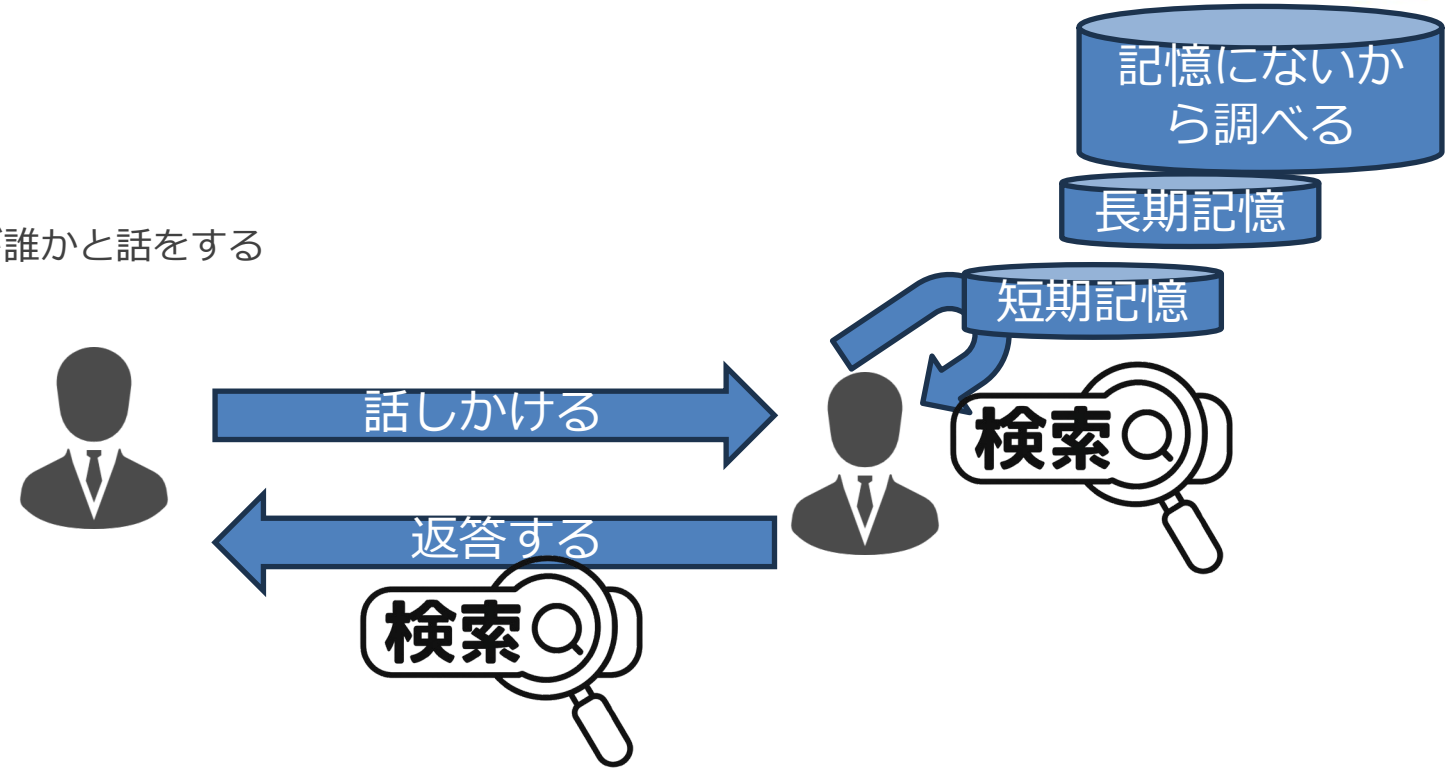
そもそも植松の認識ではすべての”やりとり”は情報検索である

例えば，人間が誰かと話をする



そもそも植松の認識ではすべての”やりとり”は情報検索である

例えば，人間が誰かと話をする



そもそも植松の認識ではすべての”やりとり”は情報検索である

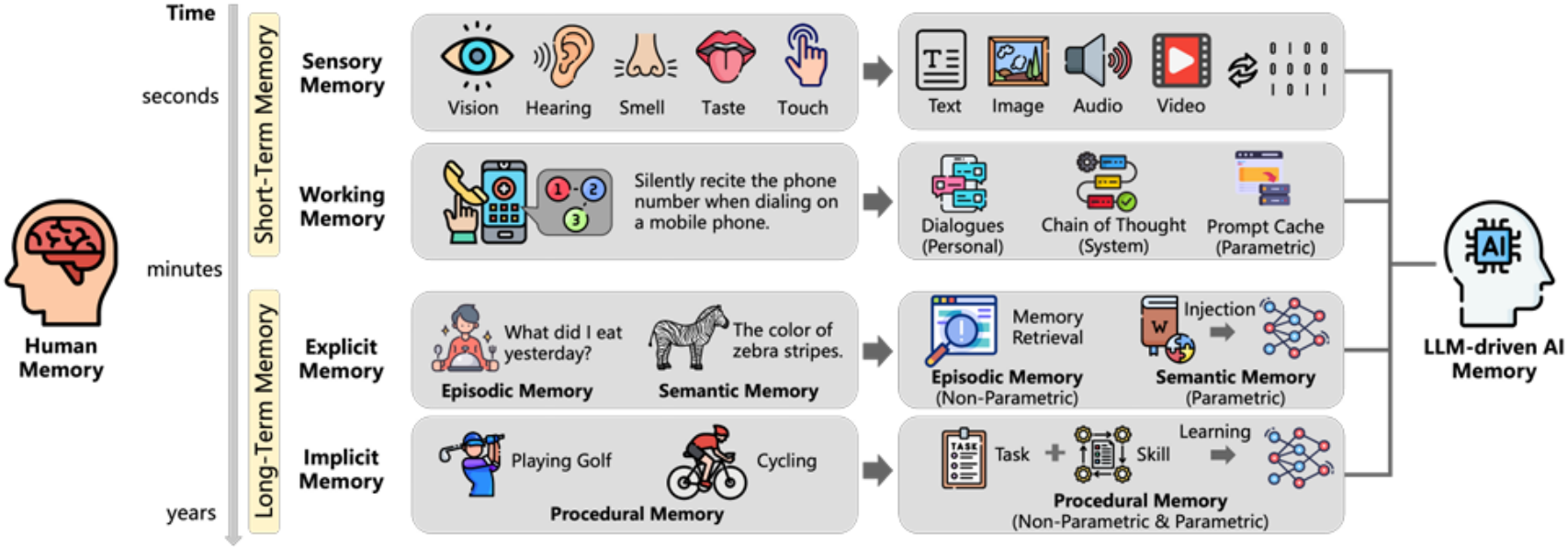


Figure 1: Illustrating the parallels between human and AI memory.

（参考）From Human Memory to AI Memory: A Survey on Memory Mechanisms in the Era of LLMs

https://arxiv.org/abs/2504.15965

第一回のアジェンダ

1. シラバスの確認
2. 教員紹介
3. 講義の特徴
4. 情報検索とは？
5. この講義で何を学べるか？(具体的な説明)
6. 情報検索基礎
   Boolean検索
7. 次週に向けて

この授業で何をやるのか？

本授業では，テキストデータを扱うLLM時代の情報検索を**自分で実装出来る力を付ける内容**になります

1 講義概要　講義概要の説明
2 特別講義　Federated Learning: Mei Kobayashi(5限なので，出れない方は応相談)
3 情報検索基礎　スコアリング1　TFIDF等のscoring技術の解説
4 情報検索基礎　インデックス1　全文検索の仕組みを理解する
5 情報検索基礎　インデックス2　インデックスの圧縮技術を学ぶ
6 情報検索応用　LLMを用いた情報検索　LLMを用いた情報検
7 情報検索応用　LLMを用いた情報検索2　第6回の授業の実
8 情報検索応用　LLMを用いた情報検索4　LLMのファインチュ
9 情報検索応用　応用例　実データを用いたEDAを学ぶ((Explo
10 情報検索応用　応用例 実データを用いた情報検索技術の応
11 情報検索実践　情報検索とLLMを応用したシステムの開発1
12 情報検索実践　情報検索とLLMを応用したシステムの開発2
方法を学ぶ
13 情報検索実践　プレゼンテーション1　実装したシステムをデ
14 情報検索実践　プレゼンテーション2　引き続きプレゼンテー
15 情報検索実践　情報検索最新動向(講演)

検索エンジンの仕組みを実装レベルで理解

LLMのtools等検索エンジン等と外部連携する仕組みを実装レベルで理解

実データとの苦悩を理解w

自分で何かを作って理解を深める(PBL)

参加者が作ったものを評価/理解することで知識を増やす

最新動向を知る

第一回のアジェンダ

1. シラバスの確認
2. 教員紹介
3. 講義の特徴
4. 情報検索とは？
5. この講義で何を学べるか？(具体的な説明)
6. 情報検索基礎
   Boolean検索
7. 次週に向けて

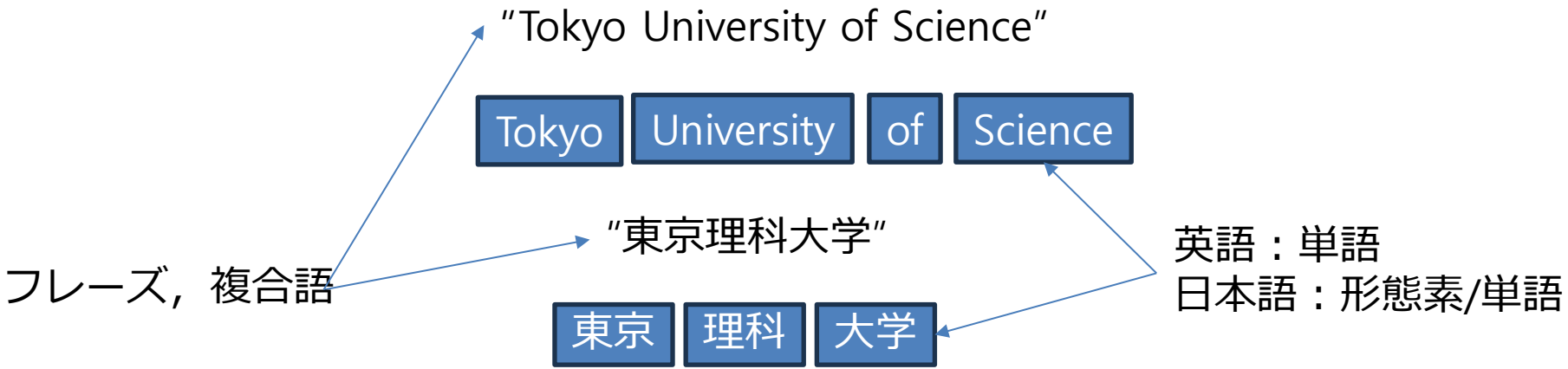
Boolean検索モデル

単語の有無を 0/1 で扱い、**集合演算**で文書を絞り込む検索手法

- 基本演算
  - AND（かつ）：両方の語を含む文書を取得
  - OR（または）：いずれかの語を含む文書を取得
  - NOT（除外）：特定の語を含まない文書を取得
- 例：
  - 「東京 AND 先生」 → 東京と先生を両方含む文書
  - 「猫 OR 学校」 → 猫または学校を含む文書
- 利用例
  - 図書館の端末での検索にいまだに使われている

ここで用語の確認

ここからの講義は英語と日本語で若干の差異が出ますので，基本用語を定義します．



Boolean検索モデル

なんでgrepじゃだめなの？

- いくつかの観点で難しいです.
  - 文書数nが大きくなるとその数だけ時間が掛かる(O(n)だからOKとではない)
- OR/NOTのオペレータは難しい

Boolean検索を実現するまで

文書集合（例：夏目漱石の作品）

- 文書1: 吾輩は猫である
- 文書2: 坊っちゃんは学校に勤める
- 文書3: こころの先生は東京に住む

Boolean検索を実現するまで

単語に分割（Tokenize）

- 文書1 → {吾輩, 猫}
- 文書2 → {坊っちゃん, 学校}
- 文書3 → {こころ, 先生, 東京}

※日本語では形態素解析（例：MeCab）を使用

{吾輩, 猫, 坊っちゃん, 学校, こころ, 先生, 東京}

ちなみに英語の場合は？

StemmingやStop Wordと呼ばれる処理を行います

文書1: The distance is 200 miles.
文書2: My name is Miles Davis.
文書3: Distance “You Are My Friend”

文書1

Tokenize: ["the", "distance", "is", "200", "miles"]
Stopword削除: ["distance", "200", "miles"]
Stemmed: ["distanc", "200", "mile"]

文書2

Tokenize: ["my", "name", "is", "miles", "davis"]
Stopword削除: ["name", "miles", "davis"]
Stemmed: ["name", "mile", "davi"]

文書3

Tokenize: ["distance", "you", "are", "my", "friend"]
Stopword削除: ["distance", "friend"]
Stemmed: ["distanc", "friend"]

コラム：　　問題

音楽のSNSの検索を担当していた時の話です

- 　　のコミュニティから検索のヒット数が極端に少ない．なぜだ？　という苦情が来ました．
- ヒント

Boolean検索を実現するまで

文書をベクトルに変換（Bag-of-Words）

- 文書1 → {吾輩, 猫}
- 文書2 → {坊っちゃん, 学校}
- 文書3 → {こころ, 先生, 東京}

{吾輩, 猫, 坊っちゃん, 学校, こころ, 先生, 東京}

上記をベクトルに変換する

- 文書1: [1, 1, 0, 0, 0, 0, 0]
- 文書2: [0, 0, 1, 1, 0, 0, 0]
- 文書3: [0, 0, 0, 0, 1, 1, 1]

（語彙リスト順に 0/1 を並べる）

Boolean検索を実現するまで

各文書に単語が存在することを１存在しないことを0であらわす

単語＼作品名	吾輩は猫である（文書1）	坊っちゃん（文書２）	こころ（文書３）	三四郎（文書４）	草枕（文書５）
猫	1	0	0	0	0
坊っちゃん	0	1	0	0	0
先生	1	0	1	0	0
東京	1	1	1	1	0
学校	1	1	0	0	0
旅	0	0	0	1	1
芸術	0	0	0	0	1

Boolean検索を実現するまで

転置インデックスを作成する

単語＼作品名	吾輩は猫である（文書1）	坊っちゃん（文書２）	こころ（文書３）	三四郎（文書４）	草枕（文書５）
猫	1	0	0	0	0
坊っちゃん	0	1	0	0	0
先生	1	0	1	0	0
東京	1	1	1	1	0
学校	1	1	0	0	0
旅	0	0	0	1	1
芸術	0	0	0	0	1

・・・

「先生」: [1, 3, 7, 20, ...]

「東京」: [1, 2, 3, 4, 10, 25, ...]

このように，単語ごとに出現する文書を保存したものを**転置インデックス**と呼ぶ
また，このリストのことを**posting list**と呼ぶ

Boolean検索の計算量

単純に1単語の検索の場合

N = 1,000,000 文書

df: document frequency
df(東京) = 10,000, df(先生) = 500

Grepの場合: O(n)
転置Index: 最速でO(1)
→ 100万倍以上の差！

Boolean検索の計算量

単純に２単語のAND検索の場合

N = 1,000,000 文書

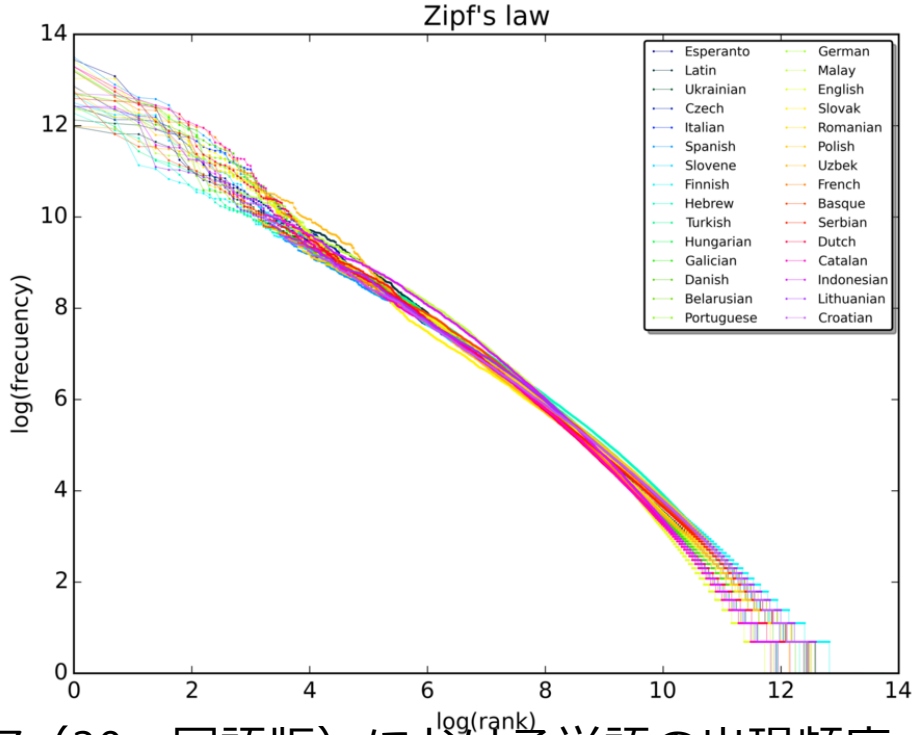
df: document frequency
df(東京) = 10,000, df(先生) = 500

Grepの場合: O(n)
転置Index: O(df(東京)+df(先生))
→ 100倍以上の差！

単語の出現頻度

単語の出現頻度はZipf則に準ずることが知られている

- 要するに低頻度語が圧倒的に多いので，転置インデックスの効果は大きい



ウィキペディア（30ヶ国語版）における単語の出現頻度

転置インデックスを使ったAND検索

Posting List の例

「東京」: [1, 2, 3, 4, 10, 25, ...]
「先生」: [1, 3, 7, 20, ...]

→ AND 検索 = posting list の交差(intersection)

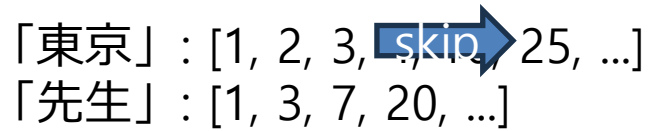
[1, 2, 3, 4,...] ∩ [1, 3, 7, 20,...] = [1, 3]

転置インデックスを使ったAND検索

Posting List の高速化：Skip Pointer

例: df(東京) = 10,000, df(先生) = 500

→ 東京の posting list は長いが、先生は短い



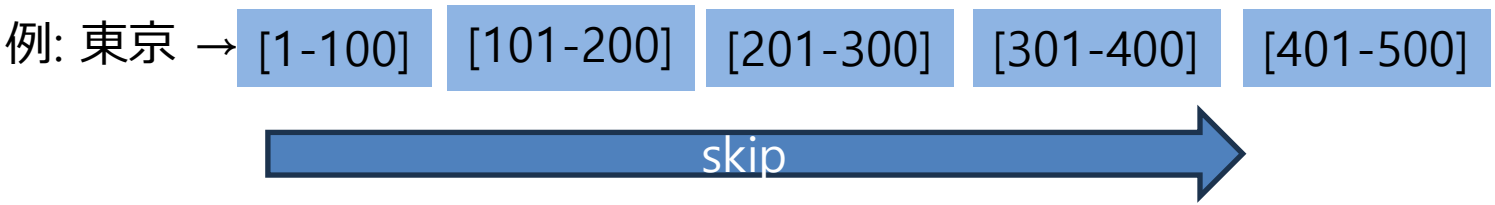
工夫: 東京のリストに Skip Pointer を置き、まとめてジャンプできるようにする

→ 無駄な比較を減らして高速化！

転置インデックスを使ったAND検索

Posting List の高速化：Blocked Inverted Index

- posting list をブロックに分割し、各ブロックに最大文書IDを保持



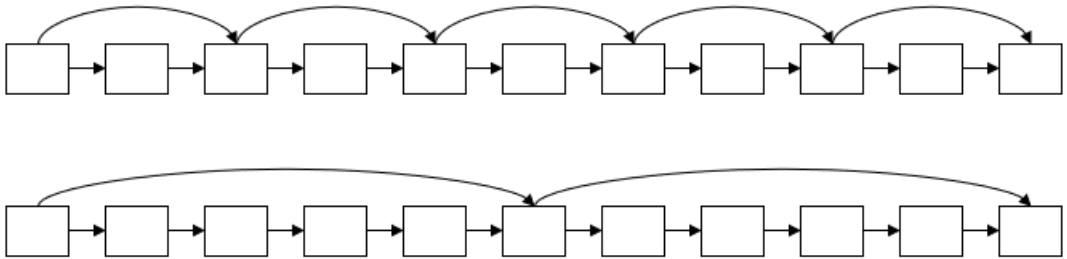
- 東京=450 を探すとき、上限<450 のブロックを一気にスキップ

  → 無駄な比較を減らせる

転置インデックスを使ったAND検索

Posting List の高速化：Blocked Inverted Index

- Skipするスパンを大きくするとSkipし過ぎるため，結局全てをなめる必要がある
- 逆にSkipするスパンを小さくすると効果が小さい



RandomにSkipポインタを置くことで，理論値でO(log(n))にするなど様々な研究がある

Intersecting two postings lists (a “merge” algorithm)

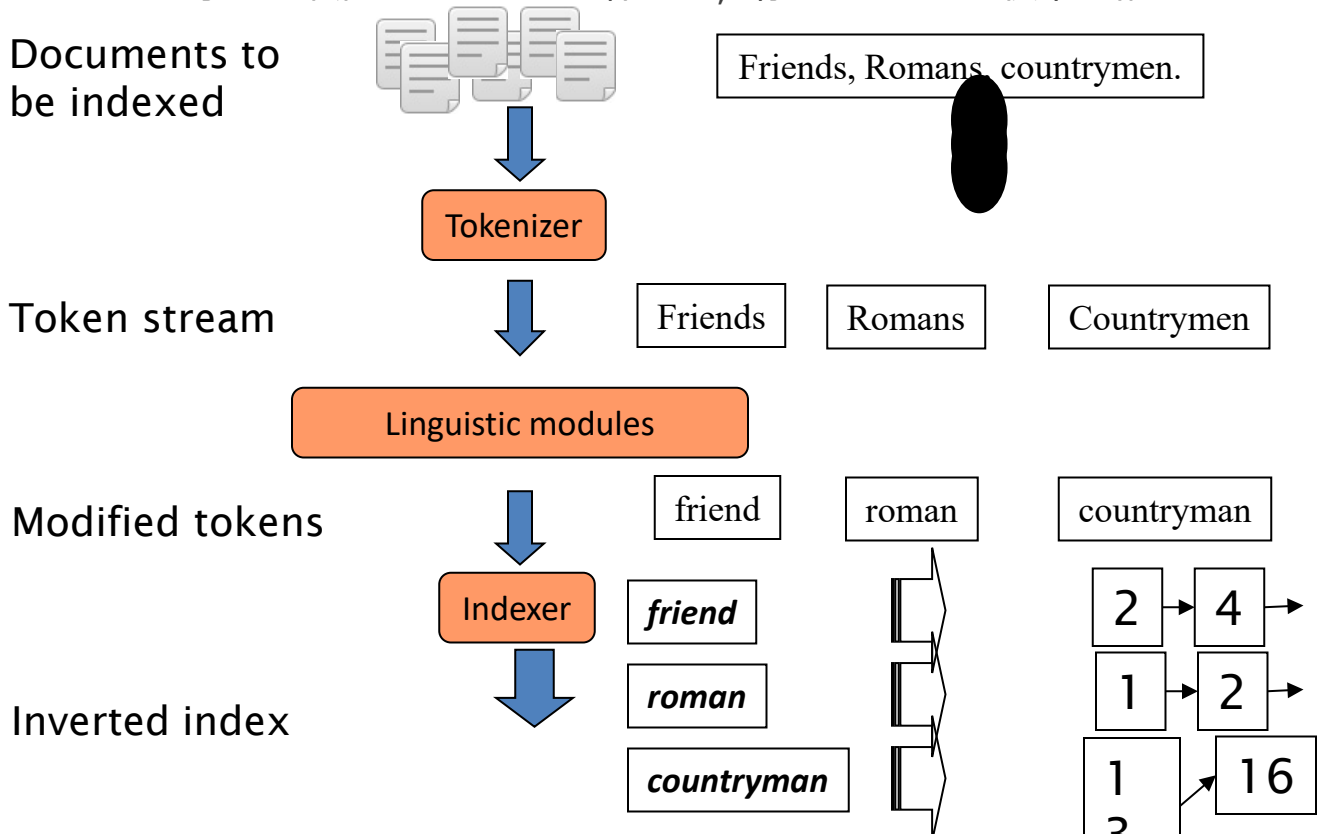
```
INTERSECT(p1, p2)
 1  answer ← ⟨ ⟩
 2  while p1 ≠ NIL and p2 ≠ NIL
 3  do if docID(p1) = docID(p2)
 4        then ADD(answer, docID(p1))
 5             p1 ← next(p1)
 6             p2 ← next(p2)
 7        else if docID(p1) < docID(p2)
 8                then p1 ← next(p1)
 9                else p2 ← next(p2)
10  return answer
```

疑似コード

IIR より抜粋

Inverted index construction

高速に検索を実行するために事前に転置インデクスを作成し，様々なBoolean検索に備えている



まとめ

情報検索の基礎としてBoolean検索を学んだ

- 単語の出現特性(Zipf則)を生かして，転置インデックスによるAND検索の実現
- Posting listをtraverseする際のskipリストによる高速化
- その他前処理等(Stop word, Stemming, 形態素解析)

次回予告：ランキングによる関連度スコア

- Boolean 検索はヒット/非ヒットのみ
  - 結果が多すぎる問題
- ユーザは「どの文書が重要か？」を知りたい
- 次回はランキング（関連度スコア）の導入へ
  -> tf-idf, BM25 (単語依存のスコア)
  -> PageRank (単語非依存のスコア)

I am a THINKER!

Thank you!

Contact:

Yukio Uematsu

yukio.uematsu@nokia.com

yukio@cs.stanford.edu