# 非構造化データの構造化における情報抽出

川崎拳人 [1]

**概要**：近年，メールや文書などの非構造化データは増加の一途をたどっているが，それらを分析し，価値のあるデータとして利用するためには，あらかじめ定められたフォーマットに整えるといったデータの構造化が必要となる．そして，構造化にあたって，的確に有益な情報を抽出することは非常に重要である．本研究では，求人情報を含むメール文書を構造化することを目的とし，その中でも情報抽出について効果的な手法の検討を行った．具体的には，メール文書から求人情報に関わる 20 の事項について質問応答データを作成し，ALBERT による機械読解を行った．また，近年研究が進められている特定ドメイン適用のための再事前学習なども取り入れつつ，実用に則した情報抽出の手法について検証を行った．

**キーワード**：自然言語処理，情報抽出，深層学習

# Information extraction in structuring non-structured data

KENTO KAWASAKI[†1]

***Abstract***: In recent years, the amount of unstructured data, such as emails and documents, has been steadily increasing, but in order to analyze them and use them as valuable data, it is necessary to structure the data, such as arranging them into a predetermined format. It is very important to extract useful information precisely when structuring. The purpose of this study is to structure mail documents containing job information, and we examined effective methods for information extraction. Specifically, question-and-answer data for 20 items related to job information were generated from e-mails and read by ALBERT. In addition, we have verified the method of information extraction for practical use, including relearning for the application of specific domains, which has been recently studied.

***Keywords***: NLP, Information Extraction, Deep Learning

## 1. はじめに

今日の AI・データサイエンスに対する社会需要の高まりの中で，円滑に質の高いデータを収集，管理することは重要になっている．しかしながら，その多くはデータベースなどの構造化データに限られ，文書やメールなどの非構造化データは，収集や管理の難しさから，AI・データサイエンスの現場で生かしきれていない．2017 年に総務省が企業を対象に調査した「安心・安全なデータ流通・利活用に関する調査研究」によると，「産業データの取り扱いや利活用において，現在または今後想定される課題や障壁」として約 50%の企業が「データの収集・管理に価格コストの増大（データのフォーマット等が共通化されていない，データ品質の管理等）」を挙げており，列挙された障壁の中で一番高い結果となった[1]．また，アメリカ，イギリス，ドイツにおいても同様の結果が出ており，非構造化データの活用に対する期待が高いものの，具体的な活用方法は見出されていないと考えられる．

本稿では，非構造化データとして，メール文書を対象とし，機械読解による手法を用いて非構造化データからの情報抽出を行った．

1 株式会社リーディング・エッジ社
Leading Edge Co.,Ltd
https://www.leadinge.co.jp/

## 2. 非構造化データ

### 2.1 非構造化データと構造化データとの違い

非構造化データとは，特定の型やフォーマットが定められていないデータである．例えば文書，メール，画像，音声などが非構造データに挙げられる．一方で，構造化データは CSV やデータベースなど，フォーマットが定められたデータである．

### 2.2 非構造化データの構造化における課題

非構造化データの構造化にあたって，下記のような課題が挙げられる．

(1) **必要情報の取捨選択**

非構造化データにおける情報の重要度や関連性は，すべて均一である．助詞であっても固有名詞であっても，その重要度に優劣をつけるためには，人間が文書をすべて読んで解釈するような高度な作業が必要となる．

(2) **文章表現の統一**

非構造化データは構造化データと異なり，あらかじめ定められた型や表記などがないため，その表現方法は多種多

様である．同一の意味を表す文章であっても，文や文字一つの順序によって表現は大きく異なるため，表現の統一が必要となる．

## 3. 機械翻訳による構造化

本研究の前段階として機械翻訳による構造化をおこなった．これは表 1 に示すメール文書から表 2 に示す JSON に構造化することを目的とし，自然言語処理モデル Transformer を用いた．メール文書には求人情報が書かれており, 案件名や内容, 勤務地などの文章が記載されている．対して JSON では,「”job_name”:[ AI Web アプリ開発案件]」などのように Key : Value の形で，メール文書の内容が書かれている．Key がメール文書から抽出したい項目名，Value がその内容となっている．次に具体例を交え，構造化タスクを紹介する．

### 3.1 特定個所の抽出

抽出すべき項目名とその内容の関連性を判断して，特定の個所を抽出する．例えば，メール文書（表 1）から「job_name」（表 2）に対応する内容である「AI Web アプリ開発案件」という文を探し出し，job_name の Value に格納する．

### 3.2 表記変換

同一の意味を持つ様々な表記を統一する．メール文書内（表 1.1）の「【単価】」に記載されている「50～60 万前後」を JSON（表 2）の「"price_min": [500000], "price_max": [600000]」にあるような数値に変換する．また,「【推奨スキル】javascript, django Web アプリ開発の経験」内の「javascript」（表 1）などの単語を「"skill_recommended_web": ["JavaScript", "Django"]」（表 1.2）のようなパスカルケースに変換することなどもこれに含まれる．

### 3.3 表現の統一

非構造化データ，特に人間が作成したデータには曖昧な表現がみられる．例えば，メール文書内（表 1）の「【年齢】40 歳はじめまでを希望」は，何歳までのことを指しているのだろうか．41 歳までだろうか，43 歳までだろうか，それとも 45 歳までだろうか，非常に判断が難しい．こうした抽象的な表現を定量的な表現に統一する．

Transformer による構造化では,「3.2 表記変換」,「3.3 表現の統一」において一定の効果があったが,「3.1 特定個所の抽出」においては，機械翻訳を用いているが故に，時として文章をそのまま抜き出さず，一部を別の単語や文に翻訳して出力するということがあった．

表 1 メール文書の例
Table 1 Example of email document.

【案件名】 AI Web アプリ開発案件
【内容】 チャットボットを用いた Web アプリの設計・開発に携わっていただきます
【勤務地】 茅場町 （東京メトロ東西線・日比谷線）
【期間】 10 月～12 月 ※延長の可能性あり
【必須スキル】 Python，C#を使用した開発経験 自然言語処理に関する知識
【推奨スキル】 javascript, django Web アプリ開発の経験
【単価】 50～60 万前後
【年齢】 40 歳はじめまでを希望
【人数】 1 名
【面談回数】 2 回（弊社同席）
【その他】 勤怠、コミュニケーションに問題のない方

表 2 JSON の例
Table 2 Example of json.

{"job_name": ["AI Web アプリ開発案件"],
"contents": ["チャットボットを用いた Web アプリの設計・開発に携わっていただきます"],
"sites_station": ["茅場町"],
"durings": ["10 月～12 月"],
"price_min": [500000],
"price_max": [600000],
"age_min": [-1],
"age_max": [43],
"skill_required_os": [],
"skill_required_lang": ["Python", "C#"],
…,
"skill_recommended_web": ["JavaScript", "Django"],
"skill_recommended_tech": [],
"required_numbers": ["1 名"],
"counts_for_interview": ["2 回（弊社同席）"],
"etc": ["勤怠、コミュニケーションに問題のない方"]}

## 4. 構造化手法の再検討

そこで本研究では，機械読解による手法を導入し,「3.1 特定個所の抽出」をした上で，その他の構造化タスクを行うこととした．図 1 のように，非構造化データであるメール文書に対して機械読解を行い，必要とする情報の抽出を

行う．その後，抽出した個所を対象に機械翻訳を用いて，「3.2 表記変換」や「3.3 表現の統一」などが必要な項目についてはそれを行う．このように，構造化タスクを細分化し，タスク特化の自然言語処理モデルを用いることで，より構造化の精度を高めることを目指した．

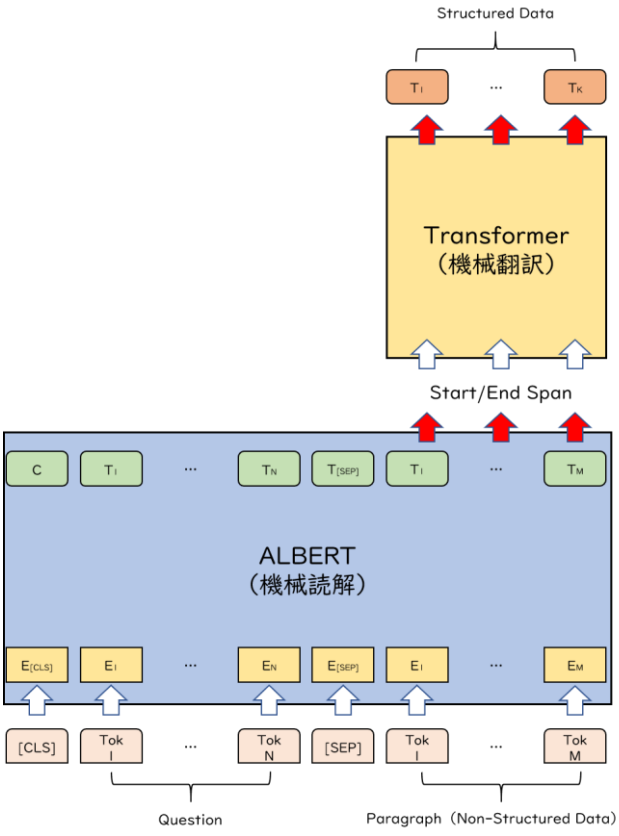


図 1　構造化イメージ

Figure 1　Image of Structuring.

**Article:** Endangered Species Act
**Paragraph:** "*. . . Other legislation followed, including the Migratory Bird Conservation Act of 1929, a 1937 treaty prohibiting the hunting of right and gray whales, and the Bald Eagle Protection Act of 1940. These later laws had a low cost to society—the species were relatively rare—and little opposition was raised.*"

**Question 1:** "*Which laws faced significant opposition?*"
**Plausible Answer:** *later laws*

**Question 2:** "*What was the name of the 1937 treaty?*"
**Plausible Answer:** *Bald Eagle Protection Act*

図 2　SQuAD

Figure 2　Example of SQuAD.

## 5. 関連研究

### 5.1 機械読解

機械読解とは，ある文章とその文章にかかわる質問が与えられた時に，機械が文章の内容を読み解き，質問の回答にあたる部分を抽出するタスクである．質問応答のデータセットとしては，スタンフォード大学が SQuAD を公開しており，機械読解のベンチマークとして一般的に使用されている．SQuAD は図 2 のように，読解対象であるパラグラフとそれに対する質問，回答で構成されている[2].

### 5.2 ALBERT

機械読解による情報抽出にあたって，ALBERT を用いた．ALBERT は BERT を計量化したモデルであり，BERT と同等の精度を保ちつつ学習の高速化に成功している[3]．BERT からの改良点は二つである．

(1) Cross-layer parameter sharing

BERT は Multi Head Attention と Feed Forward Network を含む 12 層の Transformer から構成されているが，ALBERT では 12 層の Transformer 間のパラメータを共有することで，BERT ベースモデルと比較してパラメータ数を大幅に削減している[3].

(2) Sentence Order Prediction

BERT における Next Sentence Prediction における，Negative Sampling はランダムな組み合わせとなるため，容易に解くことができ，学習に寄与しないことが問題視されていた．そこで ALBERT では，Negative Sampling に正例の文の順序を逆にしたものを使用している．これにより，文章のトピックから判断するのではなく，文章の一貫性から判断するようになる[3].


図 3　再事前学習

Figure 3　Training flow in re-pretraining.

### 5.3 再事前学習

ALBERT では，基本的に転移学習による学習が行われる．転移学習とは，大規模コーパスで事前学習を行った後に，ターゲットコーパスでファインチューニングを行う手法である．事前学習では，教師ラベルのないコーパスを用いて，文中の欠落した単語の予測やある文章と文章に関係があるかどうかを予測するなど汎用的な言語モデルを構築する．一方，ファインチューニングでは，教師ラベルのついたコーパスを用いて，タスクに特化学習を行う．近年の自然言

語処理分野における学習は転移学習が主流であり，精度向上に大きく寄与している．また，事前学習モデルからターゲットドメインのコーパス用いて再び事前学習を行う再事前学習は，医療，金融など特定の専門分野における言語知識を獲得する上で有効であることが知られている[4].

## 6. 実験

本研究では，これまでに作成したメール文と JSON 形式のデータから SQuAD 形式のデータを作成し，情報抽出に関して評価実験を行った．JSON データのうち，表記変換や表現の統一がなされている項目については，それらを施す前の文章を教師データとした．SQuAD 形式のデータは JSON の項目数 20 に合わせ，各メール文に対する質問応答を作成した．先行研究をもとに，次に示す様々な条件下で実験を行い，箇条書き文章の情報抽出における各モデルの有効性について検証した．

### 6.1 大規模コーパスを用いた事前学習モデルによる転移学習

Wikipedia の日本語全記事をもとに事前学習した albert-japanese を用いて転移学習を行った[5]. 事前学習モデル albert-japanese は，語彙数を 32,000，入力文の最大長を 512，バッチサイズを 256 として 140 万 step の学習によって作成された．また，単語分割に使用したトーカナイザーは SentencePiece である．ファインチューニングでは，この学習モデルの重みを初期値とし，ターゲットドメインのコーパスを用いて QA タスクにおける学習を行った．

### 6.2 小規模コーパスによる再事前学習

Wikipedia の中でも秀逸な記事・良質な記事（1,647 記事）を用いて事前学習を行い，その後，ターゲットドメインのコーパスを用いて再事前学習を行った[4][6]．まず，事前学習にあたっては, 語彙数を 32,000, 入力文の最大長を 512, バッチサイズを 64 として 48 万 step の学習を行った. また，学習率は 1e-4 である．再事前学習では，事前学習モデルを重みの初期値とし, 語彙数を 32,000, 入力文の最大長を 512, バッチサイズを 64 として 24step の学習を行った．また，この時の学習率は 2e-5 である．最後に，こうして得られた事前学習モデル・再事前学習モデルを用いて，ターゲットドメインのコーパスによるファインチューニングを行った．

### 6.3 MeCab を用いた単語分割

ALBERT の実装では，SentencePiece がデフォルトで用いられているが，サブワード化によって未知語に対する欠損がなくなる一方，日本語らしい文章を考慮した単語分割がされないことによる弊害もある．そこで，実験 2 の文章の単語分割において MeCab を使用し, SentencePiece との精度を比較した．入力文章を MeCab（NEologd[a]）で事前に単語分割し，さらに SentencePiece でサブワード化した．ファインチューニングにあたって利用した事前学習モデルは，「6.2 小規模コーパスによる再事前学習」で使用した事前学習モデルと再事前学習モデルである．

## 7. 結果

### 7.1 評価指標

評価指標には Exact Match と F 値を用いて各質問に対応する箇所を抽出できているかを評価した．大規模コーパスによって事前学習された albert-japanese からファインチューニングを行ったモデルの精度が最も高い結果となったが，小規模コーパスによる事前学習ののちメール文書による再事前学習を行った Wikitext-JA（再事前学習モデル）も大差ない精度となった．加えて，Wikitext-JA において，トーカナイザーを MeCab に変更, もしくは再事前学習を行うことで, 精度が向上した. F 値では, 事前学習モデルでも MeCab を使用した場合や再事前学習をした場合に比べて，あまり離れた数値ではないため，部分的には正解コーパスと一致していると思われる．

表 3　事前学習モデル及び単語分割における評価結果
Table 3　Evaluation results of pre-training model and tokenizer.

| Pretrain model | Tokenizer | | | |
|---|---|---|---|---|
| | SentencePiece | | MeCab | |
| | EM 値 | F 値 | EM 値 | F 値 |
| albert-japanese | 96.95 | 97.79 | - | - |
| Wikitext-JA（事前学習モデル） | 92.23 | 95.39 | 95.03 | 96.13 |
| Wikitext-JA（再事前学習モデル） | 95.47 | 97.00 | 95.49 | 96.45 |

### 7.2 項目別の誤答率

次に，Wikitext-JA（事前学習モデル）の EM 値を算出した場合における，質問応答項目別の誤答率と正解コーパスにおける文字数の平均長の関係を図 4 に示す．誤答率が高い項目ほど教師データの文字数の平均長ながくなり，長い文章ほど，抽出精度が低いことがわかる．

[a] 2020-08-20 時点の辞書を使用している
https://github.com/neologd/mecab-ipadic-neologd/releases/tag/v0.0.7

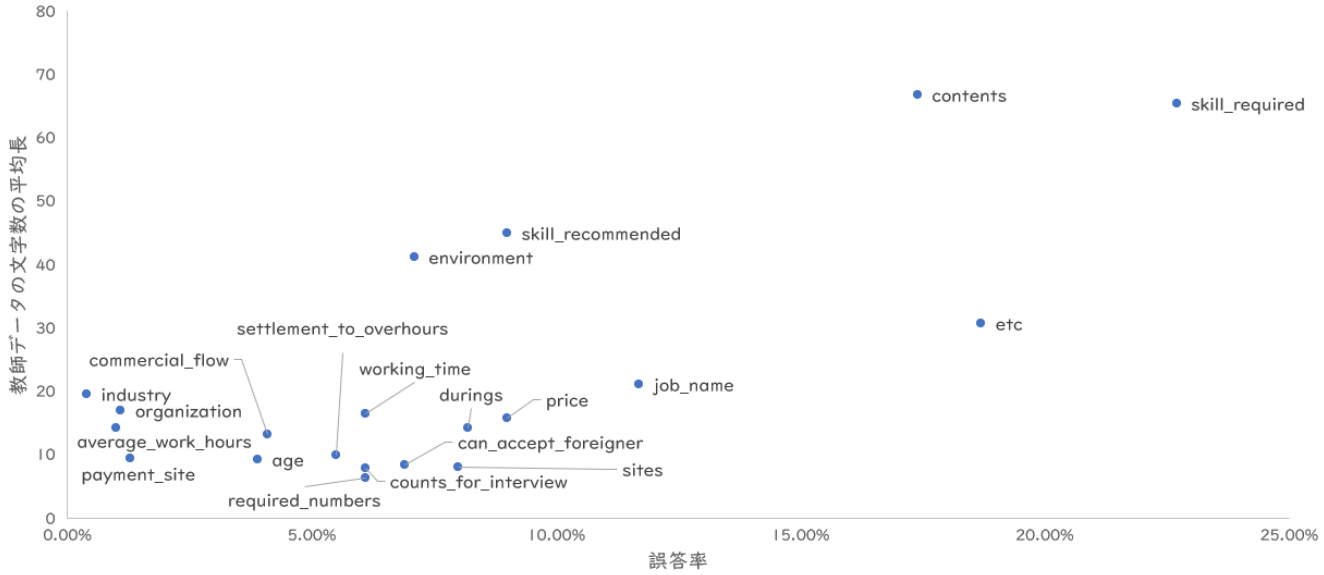


図 4　質問応答項目別の教師データの文字数の平均長と誤答率

Figure 4　Average length of characters whose teacher data for each items and error rate.

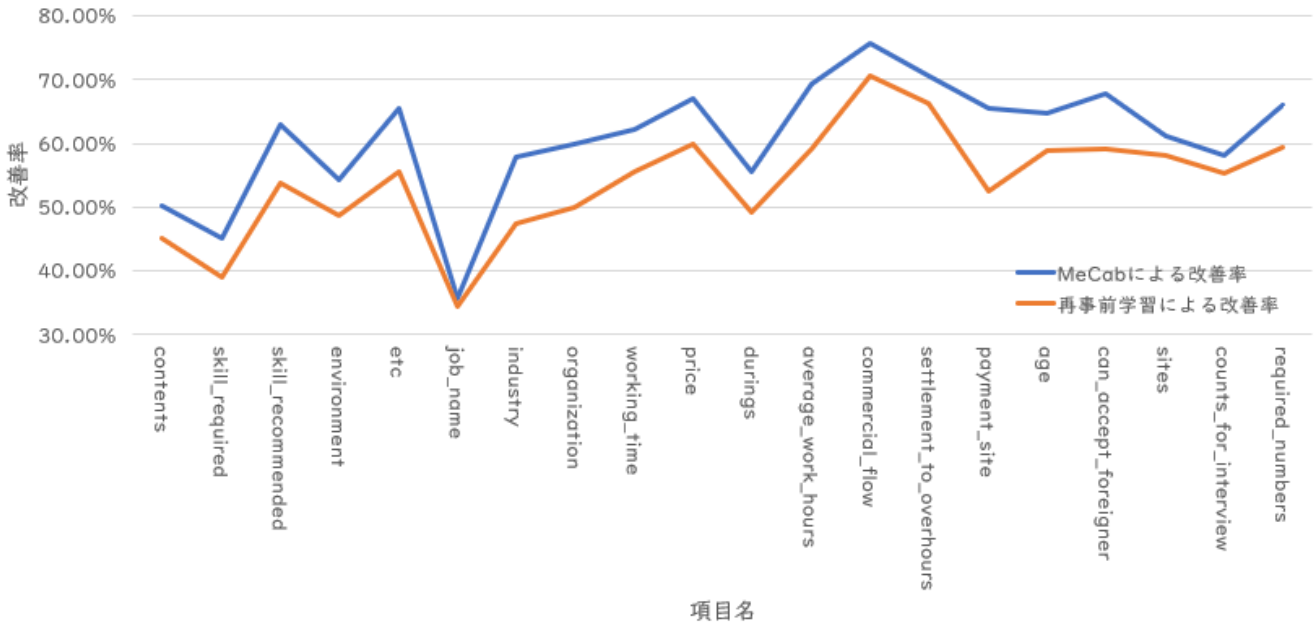


図 5　MeCab，再事前学習による質問応答項目別の改善率

Figure 5　Improvement rate by MeCab and re-pretraining for each items.

### 7.3　MeCab・再事前学習による改善

次に，Wikitext-JA（事前学習モデル）の誤答において，MeCab を用いて学習した場合と，再事前学習を行った場合における改善率を示す．誤答率が低かったもののが，さらに改善されている一方で，contents，skill_required など予測する文字列が長い項目の改善は他の項目と比べて小さかった．したがって，長い文章を抽出するという点に絞ると MeCab や再事前学習では問題の解決には至らなかったと考えられる．また，一部例外として，予測する文字列が比較的短い job_name の改善率が低く，項目特有の問題があると考えられる．

## 8.　おわりに

本研究では，非構造化データの構造化における情報抽出に焦点をあて，その手法について検討を行った．大規模コーパスによる事前学習には多少劣るものの，小規模コーパスでもトーカナイザーに MeCab 用いることや再事前学習を行うことで，より効率的に情報抽出をすることができた．今後の研究としては，膨大な非構造化データに対して，教師なし学習によるテキストセグメンテーションを行うこと

で，有益な情報とそうでない情報区別する手法について検討している．

## 参考文献

[1] 総務省，平成 29 年版情報通信白書，第 1 部 2 章 2 節 データ流通利活用における課題. https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h29/pdf/n2200000.pdf
[2] “SQuAD2.0 The Stanford Question Answering Dataset“. https://rajpurkar.github.io/SQuAD-explorer/
[3] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, Radu Soricut ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. p.4-5, 7-8. 2019
[4] 小川晃，友利涼，亀甲博貴，森信介 WikiText-JA 構築による BERT 事前学習の効率化．言語処理学会第 26 回年次大会発表論文集. p.1174. 2020
[5] “ALBERT with SentencePiece for Japanese text.”. https://github.com/alinear-corp/albert-japanese
[6] “Wikitext-JA“. http://www.lsta.media.kyoto-u.ac.jp/data/wikitext-ja/

## 付録

### 付録 A.1 質問回答項目と詳細

| 項目名 | 詳細 |
|---|---|
| job_name | 案件名 |
| industry | 業界 |
| content | 案件概要 |
| sites | 勤務地 |
| durings | 期間 |
| price | 単価 |
| age | 年齢 |
| skill_required | 必須スキル |
| skill_recommended | 推奨スキル |
| environment | 開発環境 |
| required_number | 要求人数 |
| counts_for_interview | 面談回数 |
| working_time | 始業時間， 終業時間 |
| average_work_hours | 平均労働時間 |
| settlement_to_overhours | 稼働時間 |
| organization | 組織名 |
| payment_site | 支払いサイト |
| commercial_flow | 商流 |
| can_accept_foreigner | 外国人可否 |
| etc | その他 |