KWJA：汎用言語モデルに基づく日本語解析器

植田 暢大[1,a)]　大村 和正[1,b)]　児玉 貴志[1,c)]　清丸 寛一[1,d)]
村脇 有吾[1,e)]　河原 大輔[2,f)]　黒橋 禎夫[1,g)]

概要：現在の自然言語処理では end-to-end のニューラルモデルが主流となっている．しかし，本質的に試行錯誤を必要とする情報分析や因果関係分析などを支援するためには，エンティティやイベント，および，それらの関係の認識が必須であり，形態素解析や述語項構造解析などの言語解析が依然として重要である．これまで，解析器はそれぞれのタスクごとに開発されており，それらを繋いで利用するにはコストが高いという問題があった．本研究では，汎用言語モデルに基づく高精度な統合的日本語解析器 KWJA を設計・構築する．本解析器は，入力誤り訂正，分かち書き，単語正規化，形態素解析，固有表現認識，言語素性付与，構文解析，述語項構造解析，橋渡し照応解析，共参照解析，談話関係解析を統一的なアーキテクチャで実現する．構築した解析器は `https://github.com/ku-nlp/kwja` で公開している．

1. はじめに

自然言語処理においてニューラルネットが席巻して以降，機械翻訳や対話など多くのタスクで end-to-end モデルが主流となっている．しかし，情報分析や因果関係分析は本質的に試行錯誤を必要とするため，その支援の手段として end-to-end モデルが最適とは限らない．このような自然言語処理アプリケーションを構築するうえで，形態素解析や述語項構造解析などのテキスト解析技術は依然として重要である．

テキスト解析は，形態素解析，構文解析，述語項構造解析，談話関係解析などのタスクを含み，そのそれぞれもニューラル化の波を受けている．特に大規模テキストから事前学習した Transformer アーキテクチャ [33] に基づく汎用言語モデル [3] を用いることで，構文解析 [25]，述語項構造解析 [31], [32]，談話関係解析 [9], [10] などのタスクにおいて，従来のモデルを大幅に上回る精度が報告されている．しかも，個別のタスクに応じた漸進的な改善よりも，事前学習手法の改善，モデルやテキストの大規模化の方が精度向上に大きく寄与する傾向が見られる．

本研究では，言語解析タスクの近年の高精度モデルがいずれも汎用言語モデルに基づくという共通性に着目し，統合的日本語解析器 KWJA（クージャ）[*1] を設計・構築する．統合解析では，従来は個別に取り組まれてきたテキスト解析の諸タスクを統合的に処理する．具体的には，入力誤り訂正，分かち書き，単語正規化，形態素解析，固有表現認識，言語素性付与，構文解析，述語項構造解析，橋渡し照応解析，共参照解析，談話関係解析を取り扱う（図 1）．

本研究の目標は，高い精度の達成に加えて，テキスト解析の利用者の利便性の向上である．従来はタスクごと別々の解析器が提供されており，それらすべてをインストールして繋ぐことは利用者にとって敷居が高かった．これに対し，統合的解析器の提供は高度なテキスト解析へのアクセスを容易にする．統合解析器の利用者は，ひとつのコマンドで入力文書に対する高度な解析の結果を得られる．この際，入力誤り訂正から談話関係解析までのどの範囲の解析を行うかをオプションで指定できる．

KWJA の設計方針は，汎用言語モデルの能力を積極的に活用し，人手によるコード・ルールの記述を最小化することである．形態素解析器 JUMAN [15][*2]，Juman++ [29][*3] や構文解析器 KNP [13][*4] などの特に低層の日本語解析器は，研究の歴史的な経緯から，辞書形式の語彙知識や人手で記

[1] 京都大学
Kyoto University, Yoshida-honmachi, Sakyo-ku, Kyoto 606–8501, Japan
[2] 早稲田大学
Waseda University, 3-4-1 Okubo, Shinjuku-ku, Tokyo 169–8555, Japan
[a)] ueda@nlp.ist.i.kyoto-u.ac.jp
[b)] omura@nlp.ist.i.kyoto-u.ac.jp
[c)] kodama@nlp.ist.i.kyoto-u.ac.jp
[d)] kiyomaru@i.kyoto-u.ac.jp
[e)] murawaki@i.kyoto-u.ac.jp
[f)] dkw@waseda.jp
[g)] kuro@i.kyoto-u.ac.jp

[*1] Kyoto-Waseda Japanese Analyzer.
[*2] `https://github.com/ku-nlp/juman/`
[*3] `https://github.com/ku-nlp/jumanpp/`
[*4] `https://github.com/ku-nlp/knp/`

© 1959 Information Processing Society of Japan

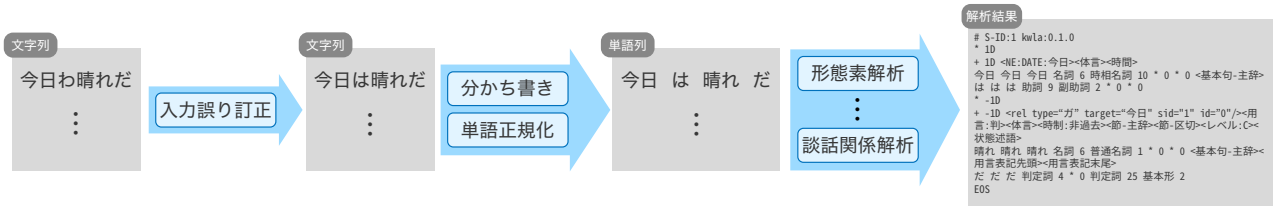

図 1 解析フロー

解析コンポーネント	入力トークン	出力	用いるリソース・素性
入力誤り訂正	文字	訂正後の文字	
分かち書き 単語正規化	（訂正後の）文字	単語 正規形	
形態素解析	単語（サブワード）	単語: 品詞・活用, 原形, 読み, 意味情報	基本語彙・準基本語彙辞書
固有表現認識		単語列: 固有表現	
言語素性付与		単語: 単語素性（基本句区切・主辞, 文節区切など） 基本句: 基本句素性（用言, 体言, 節区切・主辞など）	
構文解析		単語: 係り先, 係り受けタイプ 基本句: 係り先, 係り受けタイプ	基本句区切・主辞
述語項構造解析		基本句: 項（外界照応含む）	基本句区切・主辞, 用言, 非用言格解析
橋渡し照応解析		基本句: 先行詞（外界照応含む）	基本句区切・主辞, 体言, 非用言格解析
共参照解析		基本句: 共参照関係の句（外界照応含む）	基本句区切・主辞, 体言
談話関係解析		節: 談話関係	節区切

表 1 各解析コンポーネントの入出力，用いるリソース・言語素性

述したルールや特徴量に強く依存してきた．しかし，語彙知識は文脈に基づく曖昧性解消がともなわず，ルールや特徴量は周辺的な現象の網羅に限界があった．本研究では，大規模テキストからの事前学習による汎化知識の獲得を前提として，相対的に小規模な注釈付きコーパスからの教師あり学習を基本とし，辞書の利用は補助的な位置づけに留める．新しい解析タスクの導入は，適切な注釈付きコーパスを構築することで簡単に行える．

KWJA は，タスクの独立性や入力トークンの単位に基づき，解析タスクを表 1 に示す解析コンポーネントに再編成する．従来の日本語形態素解析は，分かち書き，品詞タグ付け，単語への原形等の語彙情報の付与からなっていたが，本研究では，分かち書きは文字ベースの Transformer を，品詞タグ付けや語彙情報付与 （本研究では便宜的に形態素解析と総称する）は単語（サブワード）ベースの Transformer を用いて解く．

各解析コンポーネントは，事前学習された Transformer を fine-tuning することで実現する．Transformer は巨大モデルであり，実行時の計算資源の制約を考慮すべきであることから，コンポーネントごとに独立に fine-tuning することを避け，可能な限りマルチタスク学習によりモデルパラメータを共有する．本統合解析器は，分かち書きだけが欲しいのであれば高計算資源を要し低速だが，高度なテキスト解析の利用を想定するのであれば実用的な選択肢を提供する．構築した解析器は `https://github.com/ku-nlp/kwja` で公開している．

2. 関連研究

Universal Dependencies (UD) プロジェクト [20] は依存構造に基づく構文解析の注釈付けを言語間で共通化することを目的としている．UD のガイドラインは，依存構造に加えて，単語分割基準や品詞タグも定めている．UD に準拠して設計された解析ツールは多言語を扱えることが大きな強みだが，それに加えて，代表的なツールである UDPipe [27] は，文分割からトークン化，品詞タグ付け，見出し語化，依存構造解析までを扱えるという特徴を持つ．UD の対象は基本的には構文木であり，拡張 UD グラフとよばれるグラフ構造に拡張する動きも見られるが，そこで扱われる省略や共参照は関係代名詞節等の構文的に明らかな関係に留まっている [23]．また，UD が採用する世界共通基準は各言語の実態を無視しがちである [5]．そのため，先駆的なタスク設計は，まずは個別言語に対象を絞って行うのが実際的だと考えている．

日本語解析ツール GiNZA [19] は UD に基づき構文解析までを扱う．GiNZA v5 は Transformer に基づく汎用言語モデル ELECTRA [1] を採用することで構文解析の精度向上を報告している．また，安岡[*5]は汎用言語モデルに基づく日本語や古典漢文の解析を精力的に進めており，事前学習モデルと構文解析等の fine-tuning モデルをあわせて 100

[*5] `https://huggingface.co/KoichiYasuoka`

© 1959 Information Processing Society of Japan

個近く（2022 年 8 月末時点）公開している．

既存の解析ツールを組み合わせてパイプラインを構成するためのフレームワークとして，Kachako [6] や Jigg [21] が提案されている．Kachako は UIMA とよばれる非構造化情報管理アーキテクチャ，Jigg は Scala XML オブジェクトを用いて各解析ツールを接続する仕組みを提供している．これらは，既存解析ツールの利用性を向上させることを目的としており，我々が目的とする統合解析器自体の設計と構築とは異なる．

3. リソース

3.1 事前学習モデル

汎用言語モデルとして，日本語言語理解ベンチマーク [12][*6]で高い性能を示している RoBERTa [17] を採用する．入力トークンの単位に基づき，文字ベースと単語ベースの 2 種類の事前学習モデルを使う．文字ベースの事前学習モデルとしては `ku-nlp/roberta-large-japanese-char-wwm`[*7]を用いる．これは，日本語 Wikipedia と日本語 CC-100[34] を用いて事前学習した RoBERTa Large モデルである．マスク戦略として whole word masking [2] を使用する．

単語ベースの事前学習モデルとしては，文字ベースの場合と同じテキストで事前学習した RoBERTa Large モデルである `nlp-waseda/roberta-large-japanese-seq512`[*8]を用いる．トークン化は，まず形態素解析器 Juman++で分かち書きした後に unigram 言語モデルでサブワード化するという 2 段階処理で行う．

3.2 注釈付きコーパス

入力誤り訂正の学習には日本語 Wikipedia 入力誤りデータセット（JWTD v2） [28] を用いる．JWTD は日本語 Wikipedia の編集履歴から入力誤りをマイニングしたデータセットである．

分かち書きから述語項構造解析・橋渡し照応解析・共参照解析までの学習には以下の 3 種類のコーパスを用いる．

- 京都大学テキストコーパス　(KC)[*9] [14]
- 京都大学ウェブ文書リードコーパス (KWDLC)[*10] [4]
- 注釈付き不満買取センターコーパス (Fuman)[*11]

ただし，KC については，述語項構造などの注釈は一部の文書にのみ付与されている．固有表現の注釈については，IREX CRL 固有表現データ [24] を統合することで注釈付きの文書数を拡大する．

談話関係解析の学習には KWDLC に付与された注釈を用いる [8]．KWDLC に対しては，専門家による小規模な注釈付けと，クラウドソーシングによる大規模な注釈付けが行われている．

いずれのコーパスについても，基本的には自動解析の結果を人手で修正することで注釈付与を行っている．ただし，単語の読みに対する人手の修正は徹底しておらず，今後の見直しが必要である．

単語や基本句[*12]に対する各種言語素性は，従来はコーパスには付与されておらず，構文解析器 KNP がルールに基づいて自動的に付与していた．本研究ではコーパスに自動付与したものから学習を行い，今後は必要に応じてコーパス中の言語素性を人手で修正する．

3.3 辞書

語彙知識は従来の言語処理において重要な役割を果たしてきた．従来の日本語形態素解析は，単語辞書を解候補の効果的な絞り込みに用いてきただけでなく，単語の原形や読みは辞書項目の記述をそのままコピーすることで出力してきた．また，辞書項目に付与された様々な意味情報が構文解析や言語解析以外のアプリケーションにも利用されてきた．例えば，動詞が付属語候補か否かは文節まとめあげに，代表表記とよばれる一種の単語 ID は日本語の表記ゆれの吸収 [26] に，同義語・反義語等の単語同士の関係は含意関係の認識等に用いられている．

近年は機械学習の進展，特に汎用言語モデルによる大幅な精度向上により，辞書が果たす役割は限定的になりつつある．とはいえ，すべての知識処理を end-to-end で解くのが最適とは限らず，単語の意味情報に対する需要は残ると考えている．そこで，形態素解析の結果出力時に辞書を参照し，各種の情報を付加した結果を利用者に提供することにする．このために，これまで整備してきた単語辞書を再編する．

従来の JUMAN 辞書は，人手で徹底的に整備した約 3 万項目の基本語彙辞書と，Wikipedia，Wiktionary，Web コーパスの自動処理によって構築した自動獲得辞書の 2 本建てであった [16]．しかし，自動獲得辞書には意味情報はほとんど付与されておらず，付加情報の提供という観点からはあまり役に立たない．そこで，自動獲得辞書の項目のうち，基本語彙辞書に関連項目がある（例えば「蒼い」は「青い」の異表記）場合は基本語彙辞書に統合した．残りの項目のうち，重要度が高いと判断した約 2 千項目を準基本語彙と名づけ，読みと代表表記を人手で付与した．統合解

[*6] https://github.com/yahoojapan/JGLUE
[*7] https://huggingface.co/ku-nlp/roberta-large-japanese-char-wwm
[*8] https://huggingface.co/nlp-waseda/roberta-large-japanese-seq512
[*9] https://github.com/ku-nlp/KyotoCorpus
[*10] https://github.com/ku-nlp/KWDLC
[*11] https://github.com/ku-nlp/AnnotatedFKCCorpus/

[*12] 自立語 1 単語を核として前後の付属語をまとめた単位．1 つ以上の基本句で文節を構成する．古いアノテーションマニュアルではタグ単位とよばれている．

© 1959 Information Processing Society of Japan

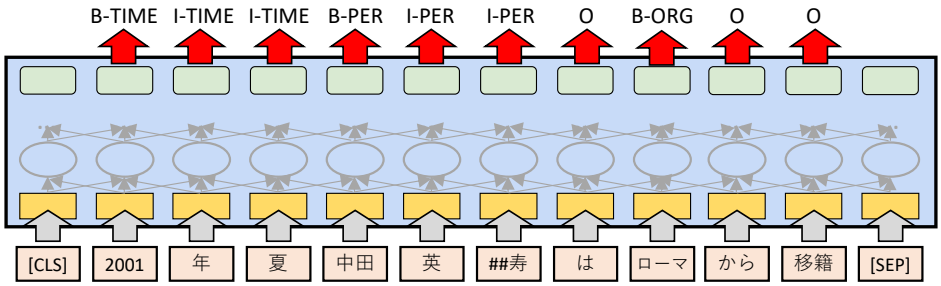

図 2 系列ラベリングタスクの解析例（例：固有表現認識）

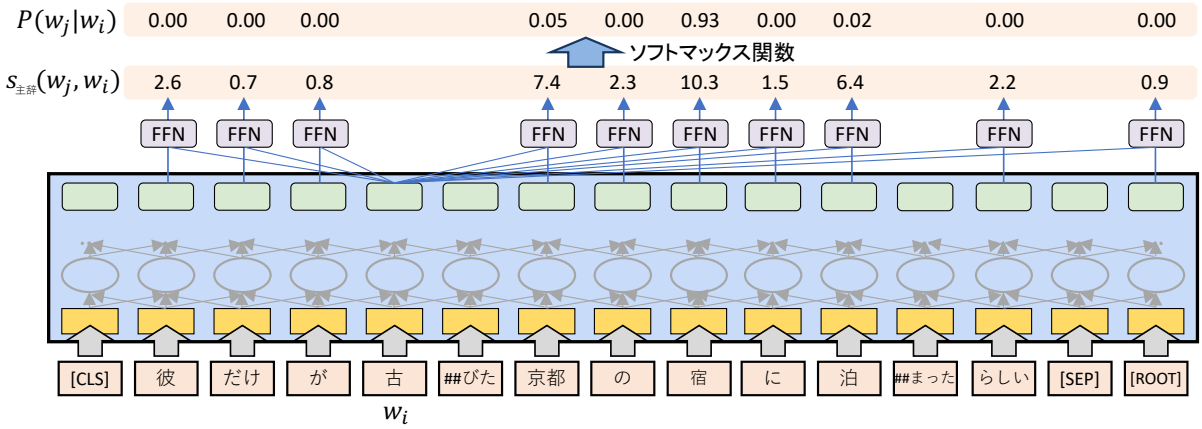

図 3 単語選択タスクの解析例（例：構文解析）

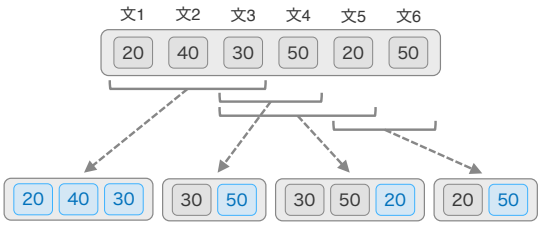

図 4 文書の分割例．数字の書かれたセルは 1 文を表し，数字は文に含まれるサブワードトークン数を表す．図では文書あたりのトークン数が 100 以下になるよう分割している．

析においては，基本語彙・準基本語彙辞書を参照する．

4. アーキテクチャ

KWJA の各解析コンポーネントは，事前学習された Transformer [33] アーキテクチャに基づく汎用言語モデル [3] を利用する．タスクごとに数層の feed-forward neural network (FFN) を Transformer 上に構築し，全体を fine-tuning する．

KWJA では，解析タスクを系列ラベリングタスク，単語選択タスク，単語間関係分類タスクのいずれかの形で定式化する．系列ラベリングタスクではテキスト中の各文字あるいはサブワードに対し，事前に定義されたラベル集合から適切なラベルを選択する．図 2 に固有表現認識における例を示す．単語選択タスクではテキスト中の各サブワードに対し，テキスト中から適切な 1 サブワードを選択する．図 3 に構文解析における例を示す．単語間関係分類タスクでは，テキスト中の各サブワードペアに対し，事前に定義されたラベル集合から適切なラベルを選択する．単語間関係分類タスクは複数選択が許容される単語選択タスクと解釈できる．本研究では，談話関係解析を単語間関係分類タスクとして定式化する．

本解析器は多くのタスクを扱うため，各々を独立に実行した場合，多くの時間的あるいは空間的計算資源が必要となる．そこで，マルチタスク学習により可能な限りモデルパラメータを共有する．表 1 の横線で区切られたブロック内のタスクはマルチタスク学習が可能である．一方，分かち書きと形態素解析などは後者の入力が前者の出力に依存するためマルチタスク学習ができない．本研究では，ブロック内のタスクについてマルチタスク学習を実行する．

形態素解析や構文解析が文を最小の解析単位とする一方，述語項構造解析や談話関係解析は文書単位の処理が必要である．本研究では マルチタスク学習を行うため，すべてのタスクにおいて入力を文書単位に統一する．なお，それにともない解析において文境界を得る重要度は低くなる．本研究では，文区切りに関しては正規表現に基づく単純なルールを使用するに留め，評価実験は行わない．

入力を文書単位としたことで Transformer モデルの最大系列長に収まらない学習事例が一部発生する．それら事例については複数文書に分割することで対処する．具体的には，解析対象の文ができるだけ多くの先行する文脈を持つ

© 1959 Information Processing Society of Japan

よう文単位で分割する．図 4 にこの分割手法を簡略化して示す．図はサブワード数がそれぞれ 20, 40, 30, 50, 20, 50 である 6 つの文を含む文書を，各文書のサブワード数が 100 以下になるよう分割する場合を表す．図の青色で示した文が解析対象の文である．対象でない文は分割後の他の文書で解析対象となるため，文脈としてのみ使用される．

5. 各解析コンポーネント

5.1 文字単位の解析タスク

5.1.1 入力誤り訂正

入力誤り訂正はテキスト中の入力誤りを検知・修正するタスクである．田中ら [28] は事前学習 seq2seq モデルを用い，入力誤りを含み得る入力テキストに対して，入力誤りを訂正したテキストを出力していた．seq2seq モデルは入力誤りを柔軟に訂正できる一方で，入力テキストに忠実でないテキストが出力される危険性をともなう．

本研究では，より保守的に，編集操作の系列ラベリング [18] を文字単位で行うタスクとして定式化する．具体的には，入力中の各文字に対して以下のいずれかの操作を予測する．

- K　そのまま
- D　削除
- R:x　文字 x で置換

この編集操作列に加えて，各文字の直前に挿入を行う操作 I:x の系列を予測する．ただし，x が NULL の場合は何も挿入しない．また，末尾への挿入に対応するため，入力文の末尾にダミートークンを追加する．末尾に挿入するには，そのダミートークンに対して挿入操作を予測すれば良い．

例えば，「固体発生」を「個体発生」に訂正するには，R:個, K, K, K, K と I:NULL, I:NULL, I:NULL, I:NULL, I:NULL を予測すればよい．同様に，「有力ある」から「有力である」への訂正は，K, K, K, K, K と I:NULL, I:NULL, I:で, I:NULL, I:NULL によって実現できる．ここで I:x の x を文字とした場合，複数文字の挿入（例えば「それは原則と不可能だ」から「それは原則として不可能だ」への訂正）に対応できない．そこで訓練データ内に存在する複数文字の挿入については対応できるように挿入操作語彙に追加する．

編集操作列は，学習データの入力誤り訂正前後の文をアラインメントすることで生成する．入力誤り訂正モデルは，入力文の各文字に対して編集操作を予測する．推論後は，各文字に対して編集操作を適用する後処理を実行し，訂正後のテキストを出力する．

評価には田中ら [28] の手法にならい，入力誤り訂正の適合率，再現率，F1 スコアを用いる．

5.1.2 分かち書き

分かち書きはテキストを単語列に分割するタスクである．本研究では文字単位の系列ラベリングタスクとして解く．系列ラベリングタスクでは B (Begin), I (Inside), O (Outside) のラベルを使用した BIO 方式を採用するのが一般的であるが，分かち書きにおいて O ラベルは存在しない（＝単語に含まれない文字は存在しない）ため，B ラベルと I ラベルのみを使用する．評価時はスパン単位の F 値を評価指標とする．

5.1.3 単語正規化

単語正規化は，「びみょーーー」のような非標準表記を「びみょう」に正規化する処理と，「上海ガニ」の「ガニ」のように連濁により語頭のカナが濁音化した単語を「カニ」に戻す処理からなる．形態素解析器 JUMAN や Juman++ は，非標準表記（JUMAN は濁音化も）への対応をラティス構築時に行っていた [7]．具体的には，入力文字列に対して辞書引きを行う際，非標準表記に対応する正規化見出し語候補をルールで動的に生成し，辞書にマッチした場合にラティスのノードとして追加していた．この手法は煩雑であり，また副作用による精度悪化を防ぐために，ある程度保守的に候補生成を行っていた．

本研究では，入力誤り訂正と同様に，単語正規化を各文字に対する正規化操作の系列ラベリングによって定式化する．推論後は，分かち書き結果と正規化操作列を参照し，後処理によって見出し語（分かち書きされた単語）を正規化する．なお，入力誤り訂正と単語正規化の境界は必ずしも明確ではないが，入力誤り訂正を適用せずに解析を行う場合を想定して，非標準表記の正規化を別タスクとして維持している．

正規化操作として以下の 6 種類を定義する．

- K　そのまま
- D　削除
- V　濁音，半濁音を清音に置換（語頭のみ）
- S　カナの小文字を大文字に置換（例えば「なぁ」を「なあ」に正規化）
- P　長音符を前接のカナに応じてカナに置換（例えば「もーれつ」を「もうれつ」に）
- E　長音符を「え」に置換（「ねー」を「ねえ」に）

例えば，「びみょーーー」を「びみょう」に正規化する正規化操作列は K, K, K, P, D, D である．

正規化見出し語はコーパスには直接記述されていないが，ルールにより得られる．具体的には，活用しない単語は原形が正規化見出し語と同じである．活用する単語は活用型・活用形をもとに活用変化表を参照し，原形を活用変化させることで正規化見出し語が得られる．正規化操作列は見出し語と正規化見出し語をアラインメントすることで生成できる．

非標準表記のコーパスにおける出現頻度は低く，正規化

© 1959 Information Processing Society of Japan

操作列としても K 以外の頻度は極端に低い．そこで正規化とは反対の非正規化処理をルールによって実装し，コーパスに適用することで疑似データを生成し，モデルの訓練・評価に用いる．評価指標は，文字のほとんどに対して K のラベルが与えられることを考慮し，K 以外のラベルに対する F 値 のマイクロ平均とする．

5.2 単語単位の解析タスク

5.2.1 形態素解析

本研究における形態素解析とは，単語（サブワード）を入力とし，その単語に対する品詞，品詞細分類，活用型，活用形，原形，読み，意味情報を付与するタスクである．

品詞，品詞細分類，活用型，活用形の付与については系列ラベリングの問題として捉え，各単語の先頭トークンの隠れ状態から分類を行う．原形は，活用型，活用形をもとに活用変化表を参照し，後処理で正規化見出し語から生成する．ただし，語幹を破壊する活用形[*13]があり，その場合は活用変化表だけでは原形が復元できない．そこで，基本語彙・準基本語彙辞書内の当該項目については，見出し語から原形への変換表をあらかじめ用意しておき，これを参照することで原形を生成する[*14]．意味情報は，原形，品詞，品詞細分類をキーとして基本語彙・準基本語彙辞書を引き，マッチした辞書項目の意味情報をコピーすることで出力する．

なお，JUMAN や Juman++は辞書引きによって解候補を生成しており，辞書にマッチしない場合のためのフォールバック処理として，未知語ノードを動的に生成していた．未知語に対しては特殊な品詞「未定義語」を付与し，KNP がルールによって適当な品詞（主に名詞）を付与しなおしていた．本研究では品詞付与を辞書によらずに分類問題として解くため，特殊品詞「未定義語」は使わない．

読みの付与も系列ラベリングとして解くが，サブワード分割への対処法が品詞等とは異なる．単語が複数のサブワードに分割されるとき，品詞等は先頭トークンを対象として分類を行うが，読みは各トークンに対して推測する．例えば，単語「初詣」がサブワード「初」と「詣」に分割される場合，それぞれに対して読み「はつ」と「もうで」を推測する．

コーパスには単語単位で読みが付与されているため，訓練データを生成するには，見出し語のサブワード分割にあわせて読みを分割する必要がある．そのために，図 5 のように，見出し語の文字列と読みの文字列のアラインメントを行う．まず，見出し語部分文字列と読みの部分文字列を被覆するノード候補を生成し，それらのノード候補にルールによってコストを与える．ただし，ノード候補は見出し語のサブワード境界をまたがないという制約を満たすとする．次に見出し語・読みの先頭から見出し語・読みの末尾に至るまでを重複なく被覆するノード列候補のうち，コスト最小のものを動的計画法を用いて選ぶ．最後に選択されたノード列を見出し語のサブワードごとにまとめあげることで，各サブワードに対応する読みを得る．ノード候補の生成には，ひらがな・カタカナ変換規則や漢字辞書の読み情報[*15]などを用いる．

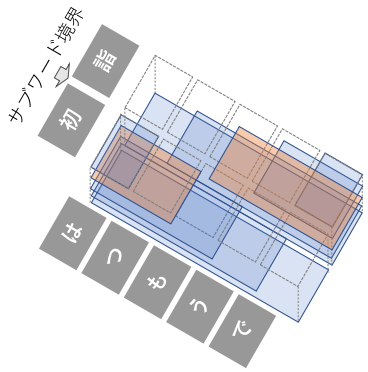

図 5 見出し語と読みのアラインメントに基づく読みのサブワード分割． 見出し語「初詣」の部分列と読み「はつもうで」の部分列を被覆するノード候補を構築し，見出し語・読み全体を被覆するコスト最小のノード列を選択する．

本研究では，各サブワードのベクトル表現から対応する読みを予測する分類器を訓練する．読み分類器の出力候補は，訓練コーパス中で 2 回以上出現する読みとする．これに，いずれの読みとも異なることを表す特殊記号 UNK，読みが表層文字列と同じであることを表す特殊記号 ID を加え，読み分類器の最終的な出力候補とする[*16]．

5.2.2 固有表現認識

固有表現認識は文中の固有表現を識別するタスクである．本研究では単語単位の系列ラベリングタスクとして解く．認識する固有表現のカテゴリは IREX CRL 固有表現データ [24] に従い，ORGANIZATION, PERSON, LOCATION, ARTIFACT, DATE, TIME, MONEY, PERCENT の 8 種類とする．これらに対応する BIO タグを各トークンに付与する．評価時はスパン単位のマイクロ F を評価指標とする．

5.2.3 言語素性付与

言語素性付与は各単語および基本句に言語素性を付与するタスクである．本研究では単語単位の系列ラベリングタスクとして解く．ただし，各単語に対して事前に定義されたラベル集合から複数のラベルを選択できるように実装する．具体的には，まずラベル集合は 69 個の言語素性からなるため，各単語の隠れ状態を 69 次元のベクトル表現に

[*13] 現在はイ形容詞のエ基本形のみ．
[*14] ほとんどの場合は，辞書によって曖昧性が解消されるが，「くれえ」が「くらい」と「くろい」に対応するように，本質的に曖昧な場合も存在する．そうした場合は，同形という出力形式を用い，両者を解析結果として出力する．
[*15] http://www.edrdg.org/wiki/index.php/KANJIDIC_Project
[*16] 読みと表層文字列が同じ場合は，その読みが出力候補の読みに含まれていても ID を予測するよう訓練する．

© 1959 Information Processing Society of Japan

変換する．続いてベクトル表現を sigmoid 関数に通すことで，各言語素性が付与されるか否かを表す確率的な値を算出する．最後に値が 0.5 以上である次元に対応する言語素性を選択することでマルチラベリングに対応する．

訓練時は, ルールベースの言語素性付与器である KNP [13] を用いて種々のコーパスに言語素性を自動付与し，それらをシルバーデータとして利用する．また，§3.2 で述べたとおり，コーパスに付与した素性の一部は人手で修正し，ゴールドデータとして利用できるように整備する．付与する言語素性[*17]の一覧は以下のとおりである．なお，† は今後人手で整備する予定の素性であることを表す．

- 単語素性
  - 基本句主辞 †
  - 基本句-区切 †，文節-区切 †
  - 用言表記先頭・末尾
- 基本句素性
  - 用言（動，形，判）†, 体言 †
  - 状態述語, 動態述語
  - 非用言格解析（動，形）†
  - モダリティ †, 時制 †, 否定表現 †, 可能表現, 敬語, 時間
  - 修飾
  - SM-主体
  - レベル（用言の強さ）
  - 係: ノ格
  - 節-主辞 †, 節-区切 †, 節-区切: 連体修飾, 節-区切: 補文, 節-機能,

基本句素性の予測は基本句主辞の単語を対象として行う．単語列の基本句，文節へのまとめあげは，それぞれ「基本句-区切」,「文節-区切」の言語素性を使って後処理で行う．節は基本句をまとめあげた単位であり，§5.2.6 で述べる談話関係解析で用いる．「節-区切」素性は節の末尾の基本句に付与する．

評価時は,「基本句-区切」および「文節-区切」についてはスパン単位の F 値を，その他の言語素性は単語単位の F 値を計算し，それらのマクロ平均を評価指標とする．なお，データに 1 つも付与されなかった言語素性は除外した上でマクロ平均を計算する．

5.2.4 構文解析

本研究における構文解析とは，単語間の係り受けおよび係り受けタイプを解析するタスクである．本研究では先行研究 [25], [36] にならい，単語選択タスクとして解く．

学習に利用するコーパスには基本句単位の係り受けのみが付与されているため，訓練時は単語単位の係り受けに変換して学習する．具体的には，基本句間の係り受けを基本句主辞の単語間の係り受けとみなし，基本句主辞でない単語は属する基本句の基本句主辞を係り先とすることで単語単位に変換する．一方で推論時は，基本句主辞の単語の係り先を基本句の係り先とみなすことで，基本句単位の係り受けを出力する．

係り受けタイプは D（通常の係り受け），P（並列），A（同格），I（部分並列）の 4 種類がある．係り元と予測した係り先の単語の隠れ状態の結合を入力とする 2 層 FFN を係り受けタイプの分類器として使用する．

構文解析は文単位の解析タスクであるが，§4 で述べたとおり，他のタスクと並列で実行するために文書単位で解析を行う．このために，文外の単語を係り先として予測しないようにマスクを施し，各文の根に当たる単語はすべて入力トークン列の末尾に追加した特殊トークン ([ROOT]) を指すように訓練・評価を行う．

評価時は CoNLL 2018 Shared Task の公式評価スクリプト[*18]を使用して基本句間の係り受けの Labeled Attachment Score (LAS) を算出し，これを評価指標とする．なお，LAS を算出するために単語単位の係り受けを基本句単位に変換する際は，言語素性付与モジュール (§5.2.3) が付与する「基本句-区切」および「基本句主辞」の言語素性は使用せず，正解の基本句区切りおよび基本句主辞の情報を使用する．

5.2.5 述語項構造解析・橋渡し照応解析・共参照解析

述語項構造解析・橋渡し照応解析・共参照解析はいずれも基本句間の意味的関係を認識するタスクである．述語項構造解析では述語に対して「誰が」や「何を」に相当する項を特定する．橋渡し照応解析では名詞に対してその名詞の必須的な情報を補う同一でない名詞を特定する．共参照解析では名詞に対して実世界における同一の実体を指す名詞を特定する．

本研究では Ueda ら [31] にならい，いずれも単語選択タスクとして解く．一般に基本句は複数の単語から構成されるが，基本句主辞素性を持つ単語を基本句の代表として使用する．単語選択に使用する FFN はタスクごと（述語項構造解析においては格ごと）に用意し，それぞれ異なる単語を選択する．実験では，述語項構造解析における格としてガ格・ヲ格・ニ格・ガ２格[*19]を対象とした．

本研究では，基本句素性を用いて解析対象とする基本句をフィルタリングする．述語項構造解析においては用言あるいは非用言格解析素性を持つ基本句，橋渡し照応解析においては述語項構造解析の対象外となった体言素性を持つ基本句，共参照解析では体言素性を持つすべての基本句を

[*17] https://github.com/ku-nlp/knp/blob/master/doc/knp_feature.pdf
https://github.com/ku-nlp/KWDLC/blob/master/doc/clause_feature_manual.pdf

[*18] https://universaldependencies.org/conll18/evaluation.html

[*19] ガ２格は「象は鼻が長い」といったときの「長い」に対する「象」のように，ガ格の項が複数存在する場合に使用される．

© 1959 Information Processing Society of Japan

タスク（評価指標）	精度 JWTD	KC	KWDLC	Fuman
入力誤り訂正（適合率/再現率/F1）	92.6±0.1/60.3±1.0/73.0±0.7	—	—	—
分かち書き（F 値）	—	98.3±0.0	98.2±0.0	97.6±0.2
単語正規化（F 値）	—	0.0±0.0	0.8±1.2	66.7±27.2

表 2 文字単位の各解析タスクにおけるコーパスごとの精度．3 つの異なるシード値で fine-tuning を行い，その精度の平均と標準偏差を記載している．“—” はコーパスにアノテーションがなく評価できないことを表す．

タスク（評価指標）		精度 KC	KWDLC	Fuman
形態素解析（読み以外: F 値，読み: 正解率）	品詞	99.6±0.0	99.4±0.0	98.9±0.1
	品詞細分類	98.8±0.1	98.4±0.0	98.2±0.0
	活用型	99.4±0.0	99.3±0.0	98.7±0.1
	活用形	99.6±0.0	99.5±0.0	98.8±0.2
	読み	95.7±0.4	95.5±0.4	97.1±0.4
固有表現認識（マイクロ F，スパン単位）		90.2±0.5	81.1±1.4	76.0±2.4
言語素性付与（マクロ F）	単語素性	98.9±0.1	98.5±0.0	97.8±0.1
	基本句素性	90.1±0.9	89.3±0.3	88.6±0.3
構文解析（LAS，基本句単位）		91.8±0.1	94.7±0.1	94.1±0.7
述語項構造解析（F 値）		67.6±1.1	77.0±0.1	74.3±0.5
橋渡し照応解析（F 値）		58.1±0.5	65.6±0.5	67.2±2.3
共参照解析（F 値）		69.5±1.6	75.5±0.1	65.2±1.0
談話関係解析（F 値）		—	27.4±7.1	—

表 3 単語単位の各解析タスクにおけるコーパスごとの精度．3 つの異なるシード値で fine-tuning を行い，その精度の平均と標準偏差を記載している．“—” はコーパスにアノテーションがなく評価できないことを表す．

対象とする．対象外となった基本句に対しては訓練時に損失を計算しない．このような基本句に関係ラベルが付いていることは稀であり，フィルタリングによってモデルはより難しい事例を学習する．

外界照応に対応するため，入力系列の末尾に照応先に対応する特殊トークンを挿入し，対象基本句として選択可能とする．実験では外界照応の照応先として，著者・読者・不特定: 人・不特定: 物を扱った．

5.2.6 談話関係解析

談話関係解析は節間の談話関係を認識するタスクである．本研究では岸本ら [8] にならい，文書中の各節ペアに対して談話関係解析を行う．例えば 3 つの節からなる以下の文章[*20]に対して，Arg1 と Arg2 の間に談話関係がないこと，Arg1 と Arg3 の間に「原因・理由」の談話関係があること，Arg2 と Arg3 の間に「逆接」の談話関係があることを解析する．

(1) [$_{Arg1}$ 厚労省によると、日本では患者数 6 人と公表している（2008 年）。][$_{Arg2}$ 患者を多く見積もっても][$_{Arg3}$ 販売によって利益水準に達することはない。]

解析対象とする談話関係は「原因・理由」「目的」「条件」「根拠」「対比」「逆接」の 6 タイプである．これらの関係はいずれも岸本ら [8] の注釈付きコーパスにアノテーションが付与されている．談話関係解析では，この 6 タイプにそのいずれにも該当しないことを示す「談話関係なし」を加え，7 値分類問題を解く．

談話関係解析を解くため，まず節のベクトル表現を得る．本研究では，節末の基本句の先頭単語の先頭サブワードの埋め込みを節のベクトル表現とする．これをもとに談話関係解析を解く分類器を構築する．本研究では，2 つの節のベクトル表現の結合を入力とし，談話関係を分類する 3 層 FFN を用いる．

訓練時は注釈付きコーパスに付与された「節-区切」の言

[*20] https://github.com/ku-nlp/KWDLC/blob/master/disc/disc_expert.txt#L360

© 1959 Information Processing Society of Japan

語素性を利用して節のベクトル表現を得る．推論時は言語素性付与モジュール (§5.2.3) が付与する「節-区切」の言語素性を利用して得る．評価指標は F 値のマイクロ平均である．

6. 実験・議論

6.1 実験設定・結果

KC，KWDLC，Fuman のすべての訓練データを用いてモデルを訓練し，それぞれのコーパスのテストデータで性能を評価した．各エポックの終了時に検証データに対して各解析タスクのスコアを評価し，その平均値が最大となる時点のモデルを採用した．単語単位の解析タスクについては，正しい分かち書きを入力として用いた．実験は乱数シードを変えて 3 回行った．今回は計算資源の都合上ハイパーパラメータの探索は行わず，先行研究での値を参考に設定した．表 4 に設定したハイパーパラメータの一覧を示す．最適なハイパーパラメータの探索は今後の課題である．

入力誤り訂正モデルは TITAN X 12GB GPU 8 枚で 64 時間訓練した．分かち書きと単語正規化モデルは Tesla V100 32GB GPU 2 枚で 2 時間訓練した．単語単位の解析モデルは Tesla V100 32GB GPU 1 枚で 30 時間訓練した．実装には transformers パッケージ [35] を使用した．

文字単位の各解析タスクの精度を表 2 に，単位単位の各解析タスクの精度を表 3 に示す．

6.2 文字単位の解析タスクに関する議論

6.2.1 入力誤り訂正

入力誤り訂正では適合率がより重視される．そこで本解析器では解析結果の確信度スコアを利用し，訂正の発現を抑制する．具体的には，モデルの出力に softmax 関数をかけたものを確信度スコアとし，モデルが訂正の操作（削除，置換，挿入）をすべきと予測した場合でも，確信度スコアが所定の閾値より低ければその操作を行わない（「そのまま」が出力される）ようにする．閾値は任意の値（0 以上 1 以下）に設定できるオプションを設ける．これにより，適合率をより重視した設定に変更するなど，利用者の望んだ振る舞いを実現することが可能となる．

表 5 に閾値をそれぞれ 0.0，0.8，0.9 とした時の各スコアを示す．ここでは適合率，再現率， F1 スコアに加えて，F1 スコアより適合率を重視した評価指標である F0.5 スコアについても報告している．閾値を上げることでより保守的な運用が可能となっている．表 2 には最も F0.5 スコアが高い，閾値 0.8 の時のスコアを記載している．

具体例として，「観覧席の人をＴＶに映ることもある。」を入力文とした時，閾値が 0.0 のモデルでは「観覧席の人がＴＶに移ることもある。」となり，0.8 のモデルでは「観覧席の人がＴＶに映ることもある。」となった．閾値を上げることで「TV に移る」といった誤変換を防ぐことができる．ただ，適合率と再現率はトレードオフの関係にあり，適合率を上げるとその分訂正できたはずの誤りを取りこぼすことになる点には留意されたい．

6.2.2 分かち書き

先行研究 [30] では各コーパスに対する分かち書きの精度（F 値）は KC が 99.5，KWDLC が 98.7 と報告されている．本研究でのスコアはこれらの値に近い値を出せており，実利用には問題のない精度である．

社内データなどの独自分野に適応する際，分かち書きの誤りに辞書項目の追加によって対処したいという需要は大きい．ただし，こうした際に利用できる語彙項目は必ずしも単語ではなく，複合語であることが多い．

そこで，複合語辞書（内部の単語構成を記述しない複合語のリスト）を用意し，それを分かち書きモデルに参照させることで分かち書きをアドホックに制御するという解法を考えている．しかし，Wikipedia のエントリを複合語辞書として用いた予備実験では，複合語辞書の利用が一貫して分かち書きの精度を低下させる結果となった．そのため，複合語辞書の利用は今後の課題としたい．

6.2.3 単語正規化

単語正規化の F 値は KC で 0.0, KWDLC で 0.8, Fuman では 66.7 と極めて低い．本研究では，注釈付きコーパス中に K （入力文字を操作しない）以外のラベルとなる事例が極めて少ないことを考慮し，単語を非正規化するルールをコーパス中の単語に適用することによって疑似データを作成し訓練・評価に使用した．しかし，K 以外のラベルの割合を調べたところ，いずれのコーパスでも，作成した疑似データを含めて 0.1%未満であり，依然としてラベル分布に極度な不均衡性が確認された．単語正規化の性能が低いのは，そうした不均衡データからの訓練が困難だったからだと推測される．

単語正規化の精度向上のために，今後，疑似データ生成規則の改良や，低頻度の言語現象に特化した訓練データの拡充を進める．後者については，部分的注釈のサポートすることで，注釈付けを効率化したい．

6.3 単語単位の解析タスクに関する議論

6.3.1 形態素解析

表 3 に示すとおり，品詞，品詞細分類，活用型，活用形の精度は非常に高い．読みについては，訓練データと同様，テストデータも人手修正が十分ではないため参考値である．

日本語形態素解析における品詞は，辞書引きによる候補列挙と連接コストという局所的な手がかりによる曖昧性解消という解析手法に配慮した妥協の産物であった [11]．例えば，「一時」は「一時停止の標識」では名詞的，「一時中断した」では副詞的用法で使われているが，形態素解析では両用法を区別せずに「名詞-時相名詞」という品詞・品詞細分類を付与するだけである．しかし，Transformer の強

© 1959 Information Processing Society of Japan

パラメータ名	入力誤り訂正	分かち書き・単語正規化	単語単位解析
エポック数	50		
Early stopping の patience	3		
バッチサイズ	64	16	16
最大入力トークン長	512		
オプティマイザ	AdamW		
学習率	5e-5	5e-5	2e-5
重み減衰	1e-2		
ウォームアップのステップ数	100		
勾配クリッピングの閾値	0.5		

表 4 各解析モジュールの訓練時のハイパーパラメータ.

閾値	適合率	再現率	F1	F0.5
0.0	82.6±0.6	70.9±1.2	76.3±0.9	79.9±0.7
0.8	92.6±0.1	60.3±1.0	73.0±0.7	83.6±0.4
0.9	95.0±0.2	53.7±1.8	68.6±1.5	82.3±0.9

表 5 入力誤り訂正における閾値ごとの各スコア

カテゴリ	コーパス			
	KWDLC		Fuman	
ORGANIZATION	74.0	(168)	0.0	(1)
PERSON	88.1	(118)	100.0	(1)
LOCATION	88.0	(398)	100.0	(6)
ARTIFACT	60.6	(114)	33.3	(3)
DATE	88.3	(245)	85.7	(3)
TIME	40.0	(16)	100.0	(1)
MONEY	100.0	(20)	100.0	(1)
PERCENT	70.0	(8)	—	(0)

表 6 カテゴリごとの固有表現認識の F 値．() 内の数値はテストデータに含まれる該当カテゴリの固有表現の総数を表す.

KC	1. 体言	99.7	(17,344)
	2. SM-主体	98.0	(4,869)
	3. 節-主辞	98.1	(4,448)
	57. モダリティ-禁止	100.0	(2)
	59. 節-機能-目的	100.0	(1)
	59. 節-前向き機能-目的	0.0	(1)
KWDLC	1. 体言	99.6	(11,739)
	2. 節-主辞	97.6	(3,656)
	3. 用言: 動	99.0	(3,494)
	59. 節-前向き機能-対比	100.0	(1)
	59. 節-前向き機能-条件	0.0	(1)
	59. 節-前向き機能-目的	0.0	(1)
Fuman	1. 体言	99.0	(731)
	2. 時制: 非過去	98.5	(416)
	3. 節-主辞	94.5	(407)
	52. モダリティ-依頼A	100.0	(1)
	52. 節-区切: 補文	66.7	(1)
	52. モダリティ-認識-蓋然性	0.0	(1)

表 7 各コーパスにおける頻度上位下位 3 基本句素性の付与精度（F 値）．() 内の数値はテストデータに付与されている該当素性の総数を表す.

力な文脈処理を考慮すると，両用法の識別はもはや不可能ではない．本研究ではコーパスに既に付与された品詞をそのまま引き継いだが，今後は実用法に応じて品詞を割り当てなおすことも検討している．

6.3.2 固有表現認識

表 3 が示すとおり，KC と比べて KWDLC と Fuman に対する固有表現認識の精度が低い．その原因を分析するため，両コーパスに対するカテゴリごとの固有表現認識の精度を調査した．結果を表 6 に示す．いずれのコーパスに対しても，ARTIFACT （固有物名）カテゴリの認識精度が低い傾向が見られる．固有物名はその多様さに対して訓練事例が少ないことが要因であると考えられるため，固有表現の辞書を組み込むことによる改善を検討する．また，Fuman についてはテストデータに含まれる固有表現の数が全体的に少ないため，F 値が不安定であることも一因であると考えられる．

6.3.3 言語素性付与

基本句素性はルールベースで付与されていることを考慮すると，表 3 における基本句素性付与の精度は若干低いと考えられる．そこで，基本句素性ごとの精度を算出し，要因を分析した．表 7 に各コーパスにおける頻度上位下位 3 基本句素性の付与精度を示す．高頻度の基本句素性に対して，低頻度の基本句素性の付与精度が不安定であることが確認できる．基本句素性付与はマクロ F を評価指標としているため，誤りが少し増えるだけで大きく F 値が変動する低頻度の基本句素性の付与精度に影響されて全体の精度が

© 1959 Information Processing Society of Japan

若干低くなっていると推測される．実際に各コーパスにおける頻度上位 50%の基本句素性に対してマクロ F を算出したところ，それぞれ 97.4 (KC)，97.4 (KWDLC)，95.5 (Fuman) であった．このため，アプリケーションで利用可能な精度は出ていると考えられる．

6.3.4 構文解析

先行研究 [25] では KC に対する係り受け解析の精度が KWDLC より良いと報告されている一方で，KWJA は KC に対する構文解析の精度が KWDLC より低く，これまでの傾向と反している．この一因として，KWJA が文書単位で解析を行っていることが考えられる．KC は解析時に設定される最大入力トークン長を超えるような長い文書が多く含まれ，それらは複数文書に分割した上で解析される (cf. §4)．この時，解析対象の文は基本的に末尾に位置することになるが，末尾は先頭と比較して位置埋め込みが十分に学習されていない可能性があり，単語の前後関係を十分に教えられないことによる構文解析への影響が表れていると推測される．

6.3.5 述語項構造解析・橋渡し照応解析・共参照解析

Ueda らの先行研究 [31] では KWDLC におけるゼロ照応解析[*21]・橋渡し照応解析・共参照解析の F 値はそれぞれ 70.3，60.4，77.0 と報告されている．本研究ではそれぞれ，63.2，65.6，75.5 であり，橋渡し照応解析は向上したものの，ゼロ照応解析の F 値は 7 ポイント程度低かった．述語項構造解析についてはハイパーパラメータの調整が必要と考えられる．

マルチタスク学習による効果を検証するため，各解析コンポーネントについてシングルタスクで学習および評価を行った．結果，述語項構造解析・橋渡し照応解析・共参照解析の KWDLC における F 値はそれぞれ 79.7，66.0，76.0 だった．結果から，マルチタスク学習によってこれらタスクの性能はわずかに低下することがわかる．これは，関連タスクとの知識の共有による利得に比べ，モデルの表現力が他のタスクに割かれたことによる損失が大きかったためと考えられる．

6.3.6 談話関係解析

談話関係解析の F 値は，類似の手法を採用している Kiyomaru ら [10] が報告している F 値 47.67 と比べ，20 ポイント低い結果となった．この原因を調べるため，談話関係解析のみをシングルタスクで学習し，その F 値を評価した．実験の結果，F 値 48.42 が得られ，先行研究の報告とほぼ一致した．このことから，性能差の主たる原因は同時学習とそれにともなう解析タスクの総合スコアに基づくモデル選択にあると考えられる．

[*21] ゼロ照応解析は述語項構造解析のサブタスクであり，述語と項が係り受け関係を持たない場合を指す．Ueda らは述語項構造解析のスコアを報告していないためゼロ照応解析のスコアを比較する．

解析器の利用者にとって 20 ポイントの解析精度の差は大きい．KWJA は実行時の計算資源の制約を考慮して可能な限りマルチタスクで学習・推論を実行するよう設計しているが，談話関係解析については解析精度を重視して，シングルタスクで学習したモデルを追加で実行するオプションを提供することを検討している．

6.4 解析速度の検証

テキストに対して KWJA がサポートする入力誤り訂正から談話関係解析までの全解析を適用し，その解析速度を検証する．今回は入力テキストとして日本語 CC-100[34] から無作為に抽出した 1,000 文を利用した．抽出された 1,000 文の平均文字数は 43.4 文字であった．解析には Tesla V100 32GB GPU を 1 枚利用し，入力誤り訂正・文字単位解析・単語単位解析をそれぞれバッチサイズ 128・512・2 で行った．また，すべての解析は最大入力トークン長を 512 に設定して行った．この結果，解析にかかった時間は 6.4 分（モデルのロード時間を除く）であった．参考として，入力テキストを Juman++ [29] および KNP [13] で解析した場合[*22]にかかった時間は 3.2 分であり，KWJA はより高度な解析を行っていることを考慮すると合理的な解析速度であると考えられる．

6.5 解析結果の利用と可視化

KWJA の解析結果は KNP フォーマットで出力される．したがって，プログラム上で KWJA の解析結果から必要な情報を取り出す処理は，KNP 向けに開発されてきた種々のツールを用いて容易に実装できる．例えば，プログラミング言語 Python で KWJA の解析結果を読み込むには，KNP の解析結果を読み込むために開発された Python パッケージである pyknp[*23]，pyknp-eventgraph[*24]，rhoknp[*25]等を利用できる．図 6 に pyknp-eventgraph を用いて出力した KWJA による解析結果の可視化 [22] を示す．

7. おわりに

本研究では，汎用言語モデルに基づく高精度な統合的日本語解析器 KWJA を設計・構築した．本解析器は，入力誤り訂正，分かち書き，形態素解析，言語素性付与，構文解析，述語項構造解析，橋渡し照応解析，共参照解析，談話関係解析を統一的なアーキテクチャで実現する．利用者は文章を入力し，欲しい解析のレベルを指定するだけで簡単に解析結果を得られる．

KWJA の設計は完成したものの，各解析コンポーネントの精度には改善の余地がある．今後開発されるより高性

[*22] KNP は “-tab -ne-crf” オプションを指定した．省略・照応解析は動作が不安定であったため実行していない．
[*23] https://github.com/ku-nlp/pyknp
[*24] https://github.com/ku-nlp/pyknp-eventgraph
[*25] https://github.com/ku-nlp/rhoknp

© 1959 Information Processing Society of Japan

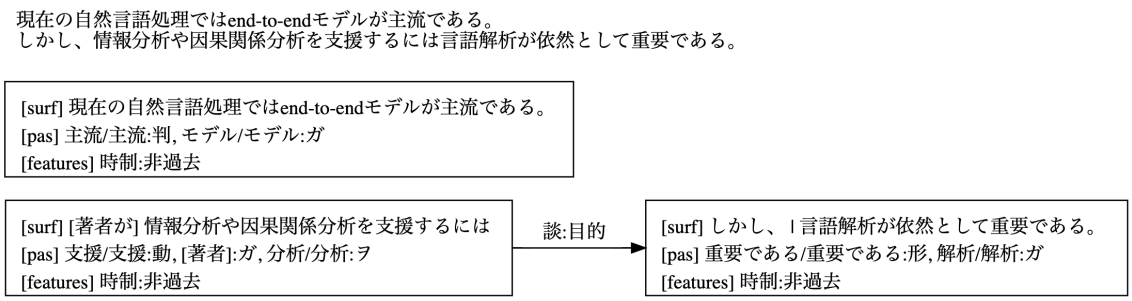

図 6 pyknp-eventgraph を用いた KWJA による解析結果の可視化．各ブロックは節を基本とする言語単位，イベントを表す．ブロック内には述語項構造解析の結果など，当該イベントに関する言語解析結果の一部が記載されている．エッジは統語的関係・談話関係を持つイベント間に与えられる．ここでは下段のイベント間にある「目的」の談話関係を表す．

能な汎用言語モデルを用いることによって，すべての解析の精度向上が期待できる．また，コーパスに自動付与した言語素性の人手修正，低頻度の言語現象に対する注釈付きコーパスの拡充，辞書の利用などによって各解析コンポーネントの精度向上を計画している．さらに，すべての解析コンポーネントを文字単位の汎用言語モデルで解くことによる解析のさらなる統合を検討する予定である．

謝辞

本研究で用いた汎用言語モデルの一部は，学際大規模情報基盤共同利用・共同研究拠点 2022 年度公募型共同研究課題「大規模な日本語モデル構築・共有のためのプラットフォームの形成」 (jh221004) において構築した．

参考文献

[1] Clark, K., Luong, M.-T., Le, Q. V. and Manning, C. D.: ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators, *International Conference on Learning Representations*, (online), available from ⟨https://openreview.net/forum?id=r1xMH1BtvB⟩ (2020).

[2] Cui, Y., Che, W., Liu, T., Qin, B. and Yang, Z.: Pre-Training With Whole Word Masking for Chinese BERT, *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, Vol. 29, p. 3504–3514 (online), DOI: 10.1109/TASLP.2021.3124365 (2021).

[3] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, Association for Computational Linguistics, pp. 4171–4186 (online), DOI: 10.18653/v1/N19-1423 (2019).

[4] 萩行正嗣，河原大輔，黒橋禎夫：多様な文書の書き始めに対する意味関係タグ付きコーパスの構築とその分析，自然言語処理，Vol. 21, No. 2, pp. 213–247（オンライン），DOI: 10.5715/jnlp.21.213 (2014).

[5] Kanayama, H., Han, N.-R., Asahara, M., Hwang, J. D., Miyao, Y., Choi, J. D. and Matsumoto, Y.: Coordinate Structures in Universal Dependencies for Head-final Languages, *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, Brussels, Belgium, Association for Computational Linguistics, pp. 75–84 (online), DOI: 10.18653/v1/W18-6009 (2018).

[6] Kano, Y.: Kachako: A Hybrid-Cloud Unstructured Information Platform for Full Automation of Service Composition, Scalable Deployment and Evaluation - Natural Language Processing as an Example, *Service-Oriented Computing - ICSOC 2012 Workshops - ICSOC 2012, International Workshops ASC, DISA, PAASC, SCEB, SeMaPS, WESOA, and Satellite Events, Shanghai, China, November 12-15, 2012, Revised Selected Papers* (Ghose, A. K., Zhu, H., Yu, Q., Delis, A., Sheng, Q. Z., Perrin, O., Wang, J. and Wang, Y., eds.), Lecture Notes in Computer Science, Vol. 7759, Springer, pp. 72–84 (online), DOI: 10.1007/978-3-642-37804-1_8 (2013).

[7] 勝木健太，笹野遼平，河原大輔，黒橋禎夫：Web 上の多彩な言語表現バリエーションに対応した頑健な形態素解析，言語処理学会第 17 回年次大会 発表論文集，豊橋，pp. 1003–1006 (2011).

[8] 岸本裕大，村脇有吾，河原大輔，黒橋禎夫：日本語談話関係解析：タスク設計・談話標識の自動認識・コーパスアノテーション，自然言語処理，Vol. 27, No. 4, pp. 889–931（オンライン），DOI: 10.5715/jnlp.27.889 (2020).

[9] Kishimoto, Y., Murawaki, Y. and Kurohashi, S.: Adapting BERT to Implicit Discourse Relation Classification with a Focus on Discourse Connectives, *Proceedings of the 12th Language Resources and Evaluation Conference*, Marseille, France, European Language Resources Association, pp. 1152–1158 (online), available from ⟨https://aclanthology.org/2020.lrec-1.145⟩ (2020).

[10] Kiyomaru, H. and Kurohashi, S.: Contextualized and Generalized Sentence Representations by Contrastive Self-Supervised Learning: A Case Study on Discourse Relation Analysis, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online, Association for Computational Linguistics, pp. 5578–5584 (online), DOI: 10.18653/v1/2021.naacl-main.442 (2021).

[11] 工藤　拓：形態素解析の理論と実装，近代科学社 (2018).

[12] Kurihara, K., Kawahara, D. and Shibata, T.: JGLUE: Japanese General Language Understanding Evaluation, *Proceedings of the Language Resources and Evaluation Conference*, Marseille, France, European Language Resources Association, pp. 2957–2966 (online), available from ⟨https://aclanthology.org/2022.lrec-1.317⟩ (2022).

[13] Kurohashi, S. and Nagao, M.: KN Parser: Japanese De-

© 1959 Information Processing Society of Japan

pendency/Case Structure Analyzer, *Proceedings of the International Workshop on Sharable Natural Language Resources*, pp. 48–55 (1994).

[14] Kurohashi, S. and Nagao, M.: Building a Japanese Parsed Corpus while Improving the Parsing System, *Proceedings of the NLPRS*, pp. 719–724 (1998).

[15] Kurohashi, S., Nakamura, T., Matsumoto, Y. and Nagao, M.: Improvements of Japanese Morphological Analyzer JUMAN, *Proceedings of the International Workshop on Sharable Natural Language Resources*, pp. 22–38 (1994).

[16] 黒橋禎夫，進　義治，柴田知秀，村脇有吾，河原大輔：日本語語彙知識の統一的・整合的管理のデザイン，言語処理学会第 19 回年次大会 発表論文集，名古屋，pp. 26–29 (2013).

[17] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V.: RoBERTa: A Robustly Optimized BERT Pretraining Approach, arXiv:1907.11692 (2019).

[18] Malmi, E., Krause, S., Rothe, S., Mirylenka, D. and Severyn, A.: Encode, Tag, Realize: High-Precision Text Editing, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, Association for Computational Linguistics, pp. 5054–5065 (online), DOI: 10.18653/v1/D19-1510 (2019).

[19] 松田　寛：GiNZA – Universal Dependencies による実用的日本語解析，自然言語処理，Vol. 27, No. 3, pp. 695–701（オンライン），DOI: 10.5715/jnlp.27.695 (2020).

[20] Nivre, J., de Marneffe, M.-C., Ginter, F., Hajič, J., Manning, C. D., Pyysalo, S., Schuster, S., Tyers, F. and Zeman, D.: Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection, *Proceedings of the 12th Language Resources and Evaluation Conference*, Marseille, France, European Language Resources Association, pp. 4034–4043 (online), available from ⟨https://aclanthology.org/2020.lrec-1.497⟩ (2020).

[21] Noji, H. and Miyao, Y.: Jigg: A Framework for an Easy Natural Language Processing Pipeline, *Proceedings of ACL-2016 System Demonstrations*, Berlin, Germany, Association for Computational Linguistics, pp. 103–108 (online), DOI: 10.18653/v1/P16-4018 (2016).

[22] 齋藤　純，坂口智洋，柴田知秀，河原大輔，黒橋禎夫：述語項構造に基づく言語情報の基本単位のデザインと可視化，言語処理学会第 24 回年次大会 発表論文集，岡山，pp. 93–96 (2018).

[23] Schuster, S. and Manning, C. D.: Enhanced English Universal Dependencies: An Improved Representation for Natural Language Understanding Tasks, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, Portorož, Slovenia, European Language Resources Association (ELRA), pp. 2371–2378 (online), available from ⟨https://aclanthology.org/L16-1376⟩ (2016).

[24] Sekine, S. and Isahara, H.: IREX: IR & IE Evaluation Project in Japanese, *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece, European Language Resources Association (ELRA), (online), available from ⟨http://www.lrec-conf.org/proceedings/lrec2000/pdf/27.pdf⟩ (2000).

[25] 柴田知秀，河原大輔，黒橋禎夫：BERT による日本語構文解析の精度向上，言語処理学会第 25 回年次大会 発表論文集，名古屋，pp. 205–208 (2019).

[26] Shibata, T., Odani, M., Harashima, J., Oonishi, T. and Kurohashi, S.: SYNGRAPH: A Flexible Matching Method based on Synonymous Expression Extraction from an Ordinary Dictionary and a Web Corpus, *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*, (online), available from ⟨https://aclanthology.org/I08-2110⟩ (2008).

[27] Straka, M., Hajič, J. and Straková, J.: UDPipe: Trainable Pipeline for Processing CoNLL-U Files Performing Tokenization, Morphological Analysis, POS Tagging and Parsing, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, Portorož, Slovenia, European Language Resources Association (ELRA), pp. 4290–4297 (online), available from ⟨https://aclanthology.org/L16-1680⟩ (2016).

[28] 田中　佑，村脇有吾，河原大輔，黒橋禎夫：日本語 Wikipedia の編集履歴に基づく入力誤りデータセットと訂正システムの構築，自然言語処理，Vol. 28, No. 4, pp. 995–1033（オンライン），DOI: 10.5715/jnlp.28.995 (2021).

[29] Tolmachev, A., Kawahara, D. and Kurohashi, S.: Juman++: A Morphological Analysis Toolkit for Scriptio Continua, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Brussels, Belgium, Association for Computational Linguistics, pp. 54–59 (online), DOI: 10.18653/v1/D18-2010 (2018).

[30] Tolmachev, A., Kawahara, D. and Kurohashi, S.: Design and Structure of The Juman++ Morphological Analyzer Toolkit, *Journal of Natural Language Processing*, Vol. 27, No. 1, pp. 89–132 (online), DOI: 10.5715/jnlp.27.89 (2020).

[31] Ueda, N., Kawahara, D. and Kurohashi, S.: BERT-based Cohesion Analysis of Japanese Texts, *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain (Online), International Committee on Computational Linguistics, pp. 1323–1333 (online), DOI: 10.18653/v1/2020.coling-main.114 (2020).

[32] Umakoshi, M., Murawaki, Y. and Kurohashi, S.: Japanese Zero Anaphora Resolution Can Benefit from Parallel Texts Through Neural Transfer Learning, *Findings of the Association for Computational Linguistics: EMNLP 2021*, Punta Cana, Dominican Republic, Association for Computational Linguistics, pp. 1920–1934 (online), DOI: 10.18653/v1/2021.findings-emnlp.165 (2021).

[33] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I.: Attention is All you Need, *Advances in Neural Information Processing Systems* (Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S. and Garnett, R., eds.), Vol. 30, Curran Associates, Inc. (2017).

[34] Wenzek, G., Lachaux, M.-A., Conneau, A., Chaudhary, V., Guzmán, F., Joulin, A. and Grave, E.: CCNet: Extracting High Quality Monolingual Datasets from Web Crawl Data, *Proc. LREC2020*, Marseille, France, European Language Resources Association, pp. 4003–4012 (online), available from ⟨https://aclanthology.org/2020.lrec-1.494⟩ (2020).

[35] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger,

© 1959 Information Processing Society of Japan

S., Drame, M., Lhoest, Q. and Rush, A.: Transformers: State-of-the-Art Natural Language Processing, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online, Association for Computational Linguistics, pp. 38–45 (online), DOI: 10.18653/v1/2020.emnlp-demos.6 (2020).

[36] Zhang, X., Cheng, J. and Lapata, M.: Dependency Parsing as Head Selection, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, Valencia, Spain, Association for Computational Linguistics, pp. 665–676 (online), available from ⟨https://aclanthology.org/E17-1063⟩ (2017).

© 1959 Information Processing Society of Japan