2W-05

ファイル操作に対する自然言語の指示文の学習

矢嶋　梨穂†　　藤田　悟†

法政大学　情報科学部†

1. はじめに

近年の先進的な技術の裏側には，ニューラルネットワークが用いられていることが多い．ニューラルネットワークを用いた自然言語の研究では，seq2seq モデルや Transformer モデルなどの技術が，より複雑な文脈の理解を可能にし，大きな進歩をもたらしている．そして，web 検索や機械翻訳，音声アシスタントなど様々な領域でその有効性が示されている．しかし，これらは，出力結果に曖昧性が残っても問題が生じにくい．一方，言語を理解し，外界に対して行動を起こすためには，理解した内容に曖昧性があってはならない．本研究では，「プログラムを編集する」という知的行動に着目し，自然言語による指示から，入力や削除などの必要なコマンドの生成と実行をすることで，ファイルを自動編集できるような知能システムの開発を目指す.

2. 関連研究

GNMT は、seq2seq モデルに Attention を加えた機械翻訳システムである[1]．GNMT では，エンコーダとデコーダの LSTM 層を 8 層とし，エンコーダの一番下の層に双方向 LSTM を用いる．GNMT の精度は従来の機械翻訳システム(PBMT)よりも優れている．エンコーダの最下層で用いられている双方向 LSTM は，入力データを双方向から処理を行い，よりバランスのとれた情報をエンコードすることができ，精度の向上に繋がる.

3. 提案手法

本研究では，実際のファイル編集を vim で行う環境を想定している．学習システムは，seq2seq を用いた学習モデルで自然言語の指示文に対応する vim の編集コマンドを生成し，それを vim のソケット通信機能を用いて実行することで求めるファイル編集結果を得ることとした.

学習モデルでは，自然言語の指示文から編集コマンドを生成させるため，学習に必要な自然言語の指示文と編集コマンドからなるコーパスを作成する．例えば，「”●”を□行目に入力して．」に対応する編集コマンドを「{□G} {O●}」，「”○”の次の行に”●”がない．」に対応する編集コマンドを「</○/> {o●}」とする．””内は入力文字や置換

Editing files from directive statements in natural languages

† Riho Yajima and Satoru Fujita, Computer and Information Sciences, Hosei University

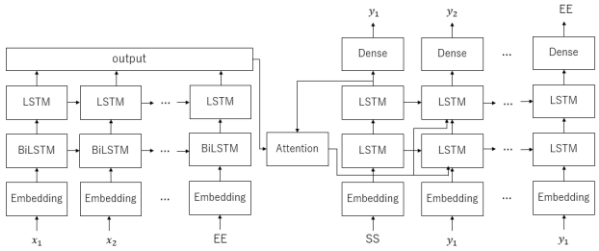

図 1 学習モデル

文字などの具体的な文字，●と□は任意の文字と数字，{}は vi コマンド，<>は ex コマンドを表している．また，他の編集コマンドについても，同様にコーパスを作成する.

今回扱う自然言語の指示文と編集コマンドは非構造化データであり，前処理が必要である．seq2seq モデルの入力と出力は単語レベルで扱っているため，自然言語の指示文は MeCab を用いて形態素解析を行い，形態素を入力とする．ただし，””内で表す具体的な文字は分割せず，別の文字で置き換えて学習させる．具体的な文字は，指示文から編集コマンドの生成の際に変化がないため，予め別の文字で置き換えても問題はなく，編集コマンド生成後に具体的な文字へ復元を行う．編集コマンドは，コマンドの役割毎に分割したものを出力とする．例えば，編集コマンドが「{□G}」の場合は「normal, □, G」と分割し，編集コマンドが「</○/>」の場合は「ex, /, ○, /」のように分割する．また，vi・ex コマンドを区別するために，vi コマンドなら normal，ex コマンドなら ex を先頭に追加する． Vim へコマンドを送信する際，vi コマンドなら[“normal”, {vi コマンド}]，ex コマンドなら[“ex”, {ex コマンド}]の形で送信することで，編集が実現する.

学習モデルでは，自然言語の指示文を学習し，編集コマンドを生成する自然言語理解を行う．モデルは，GNMT のモデルを参考に，エンコーダとデコーダの LSTM 層を 2 層，エンコーダの最下層を双方向 LSTM とした seq2seq モデルで，エンコーダとデコーダの接続には Attention を用いる(図 2).

エンコーダでは，自然言語の指示文に対して前処理を行い，時系列に並べたものを入力列(x)とする．各入力列は，まず Embedding 層でベクトル化される．Embedding 層では，学習後にコーパスに存在しない未知語に対してもベクトル化を行うことが出来るようするために，word2vec を用いた学習済みのデータを用いる．LSTM 層では，長期の依存性を学習する.

Copyright ©2022 Information Processing Society of Japan.
All Rights Reserved.

エンコーダで双方向 LSTM を用いることで，未来の情報を学習し，予測性能を向上させる．

デコーダでは，編集コマンドに対して前処理を行い，時系列に並べたものを入力列(y)とし，Embedding 層では，入力列の one_hot 表現を行列ベクトル積で，特徴量をより小さい次元に圧縮してベクトル化する．この変換はニューラルネットワークで学習をする．LSTM 層はエンコーダと同様に長期の依存性を学習する．デコーダの全結合層では，LSTM 層の出力の隠れ状態ベクトルを基に，目的の出力列の one_hot 表現に変換する．

Attention では，入力列の各単語の重要度を考慮する．エンコーダの全ステップの隠れ状態と，求める単語の 1 ステップ前のデコーダの LSTM の隠れ状態から，単語の重要度を表す attention weight と求める単語の文脈ベクトルを生成する．本研究では，Bahadanau らが提案した Attention モデルを用いる[2].

Vim にはソケット通信機能が備わっている．入力した ex コマンドや vi コマンドをサーバーが vim へ送信することで，キーボード操作と同じようにファイル編集を行うことができる[3]．本研究では，自然言語の指示文を入力とし，サーバープログラム内で，学習済み seq2seq モデルを呼び出し，生成した編集コマンドを vim へ送信することで，自然言語の指示文からファイルの自動編集を可能にする．

4. 実験

学習モデルのみで編集コマンドが正しく生成されることを評価する．学習に使用するデータは提案手法で作成したコーパスで， 学習させるコマンドの種類は，入力・削除・置換・保存・終了の 5 種類である．コーパス内に重複した指示文は存在せず，(学習用:検証用:テスト用)=(0.784:0.196:0.02)の割合でコーパス内のデータを割り当てる．学習用データと検証用データで学習を行い，テスト用データで評価を行う．評価は，生成された編集コマンドがコーパスに登録されている正解の編集コマンドと完全に一致した割合を計算する．完全一致に設定した理由は，編集コマンドの文字が一つでも異なれば、それは別の命令になってしまうからである．また，完全一致にすることで，出力結果の曖昧性を排除することができる．

同じ学習モデルでも，コーパスサイズやコーパスの中身（編集コマンド毎の比率），LSTM の次元数などが異なると学習結果も変化する．基本機能の実験では，学習モデルの LSTM の次元数を変化させ，編集コマンドの比率とコーパスサイズが異なる 2 つのコーパスで学習させたときのテスト用データの精度を比較する．コーパス 1 は，サイズが 20000（入力，削除，置換: 6000，保存・終了: 1000）であり，これは入力と削除と置換コマンドの数が等しいコーパスである．コーパス 2 は，サイズが 19800（入力: 9600，削除: 5400，置換: 2800，保存・終了: 1000）である．表現できる自然言語の指示文の数が入力，削除，置換の順で多いため，コーパス内の比率もそれに合わせている．エポック数は 30 である．

表 I．テストデータに対する精度

LSTMの次元数 （1epochあたりの実行時間）	64dim	128dim	256dim	512dim
Corpus1	0.8000 (14s)	0.9850 (34s)	0.9775 (108s)	1.0000 (356s)
Corpus2	0.9595 (15s)	1.000 (38s)	1.0000 (112s)	0.9797 (373s)

表Iより，LSTM の次元数は大きいほうがテストデータに対する精度が良いことが分かる．学習が失敗した例では，生成されたコマンドの行番号が一致しない場合が大半だった．これは LSTM の次元数が少ないことで記憶できる情報が少なくなってしまい，行番号の情報が欠けてしまったのではないかと考える．また，次元数が大きいとより多くの情報を保持でき，精度も高くなるが，学習に多くの時間を要する．今回のモデルでは，編集コマンドの比率を変えたコーパス 2 の精度が良くなること，LSTM の次元数が 256 で十分な情報を保持できることが分かった．

実環境を模した実験では，41 人から集めた自然言語の指示文や，学習に用いるコーパスにないような表現の指示文に対して，vim のソケット通信を通じて，正しく自動編集が行われるか確認を行った．その結果，1 文の中に動詞が 1 つのようなシンプルな指示文に対して正しく動作することを確認した．

5. むすび

本研究では，自然言語による指示から入力や削除など，ファイルの自動編集に必要なコマンドを生成し，実行する知的システムの開発を行った．実環境を模した実験では，コーパスにない表現の文も含めて正しく編集を行うことを確認した．

学習システムは，コーパス内の自然言語の指示文に対して，高い精度を示した．自然言語の指示文の表現は幅広く，編集箇所が曖昧な文や編集前のテキストの情報を理解していないと編集できないような文については課題がある．そのような指示文に対しても学習できるように，コーパス内の指示文の表現を増やす必要がある．

参考文献

[1] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, “Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation”, arXiv:1609. 08144v2[cs. CL] 8 Oct 2016/, (参照 2021-09-13).

[2] Dzmitry Bahdanau, KyungHyun Cho, Yoshua Bengio, “NEURAL MACHINE TRANSLATION BY JOINTLY TO ALIGN AND TRANSLATE”, arXiv: 1409.0473v7 [cs.CL] 19 MAY 2016, (参照 2021-11-25).

[3] Bram Moolenaar: “channel-Vim 日本語ドキュメント”. vim-jp>>Vim のユーザと開発者を結ぶコミュニティサイト． 2020-10-17. <https://vim-jp. org/vimdoc-ja/channel. html#channel>, (参照 2021-09-13).

Copyright ©2022 Information Processing Society of Japan.
All Rights Reserved.