

未定

新堂 風

令和2年2月

電気情報工学科

概 要

Portable Health Clinic（以下、PHC）は発展途上国農村部における健康促進に向けた遠隔医療システムである。ヘルスアシスタントと呼ばれるスタッフが複数の健康測定器具を医者のない農村部に持ち込み、村民に対して健康診断を行う。健康診断の結果、医者からの診断が必要であると判断された患者は都市部にいる医者と電話を通して繋がり、診断を受けることができる。このシステムによって、医者が直接診断できない発展途上国農村部においても人々は診断を受けることができる。このシステムでは、医者は患者を診断しながら症状や処方薬などをノートに取り、通話後にそれをコンピュータに入力して処方箋を作成する。このときにノートに書かれた手書き文字を認識し、その情報を元に処方箋を予測してコンピュータに入力する手間を削減できれば、医者の時間を節約でき、医者はさらに多くの人々の診断を行うことができる。

本研究では、処方箋の予測に向けたシステムの初期研究として、再帰型ニューラルネットワークを用いたオンライン手書き医療用語認識を行う。また、オンライン手書き文字のデータ拡張手法として（Ratio 手法）を提案する。

本研究ではバングラデシュでの処方箋予測に向けた実装及び評価を行う

が、現在医療用語に特化したデータセットがオープンソースとして存在しないため、PHCにおける過去の処方箋データから頻出する単語を座標の時系列データとして収集する。

様々な手書き文字に対応するため、本研究では多様な提供者から大量のデータを得る必要がある。しかし、十分なデータ量を確保するためには非常に多くの労力と時間を要する。

データ量の少ない機械学習にも対応したオンライン手書き医療用語認識の実現を目指して、現在データ拡張手法を用いたオンライン手書き医療用語認識が開発されている。拡張手法は、ストロークの回転と平行移動によってデータ量を水増しする SRP(Stroke Rotation and Parallel-shift) 手法を用いている。この用語認識はデータ量が少ない状況での学習が実現でき、高精度でのオンライン手書き医療用語認識が可能であることが確認されている。しかし、ストロークの回転と平行移動を同時に行うため、単語によっては拡張手法で文字の形が大きく変わってしまい、認識が困難になってしまう場合がある。そのため、この用語認識は学習の際のデータのとり方により認識精度にばらつきが出てしまうという問題がある。

本論文では、(データの取り方による精度ばらつきをなくす)に対応したオンライン手書き医療用語認識システムを提案する。オンライン手書き文字のデータ拡張手法として文字の縦横比を変更しデータ量を水増しする Ratio 手法を提案する。Ratio 手法は、既存の SRP 手法と比べて、文字の形を大きく変更せずにデータ量を水増しすることができるので、学習の際のデータのとり方で認識精度にばらつきが出るという問題を軽減することができる。本研究では既存の SRP (Stroke Rotation and Parallel-shift) 手法と Ratio 手法を組み合わせで実現した。収集した (要修正: 15991

語) のデータを用いて, (SRP 手法と Ratio 手法) でデータ拡張を行った後に BidirectionalLSTM で学習を行った結果, 480 語のクラスにおいて () %の精度で単語を認識した. この結果は SRP 手法のみの場合の認識精度と比べて () %高かった, データ拡張を行わなかった場合の認識精度と比べて () %高かった

目次

第1章	はじめに	1
1.1	研究背景	1
1.2	研究目的	2
1.3	論文構成	2
第2章	関連技術	3
2.1	手書き文字認識の種類	3
2.2	再帰型ニューラルネットワーク	4
2.2.1	RNN の構造	5
2.2.2	LSTM	7
2.3	関連研究	8
2.3.1	直線データと BidirectionalLSTM を用いた中国語認識システム [7]	8
2.3.2	dropStroke 手法を用いたデータ拡張と筆者同定システム [8] . . .	8
2.3.3	遠隔医療システムにおける 処方箋予測に向けた手書き医療用語認識に関する研究	9
第3章	LSTM を用いた手書き医療用語認識と SRP 手法及び Ratio 手法による データ拡張	11

3.1	システムの概要	11
3.2	前処理ブロック	12
3.2.1	点の除去	12
3.2.2	正規化	14
3.2.3	特徴量抽出	14
3.3	データ拡張ブロック	16
3.3.1	ストロークの回転	16
3.3.2	ストロークの平行移動	18
3.3.3	文字の縦横比変更	18
3.4	機械学習ブロック	20
第4章	実装	22
4.1	データ収集	22
4.1.1	医療用語コーパス	22
4.1.2	データ収集用アプリ	24
4.2	使用機器	24
4.3	学習モデル構造	25
第5章	評価	28
5.1	高橋さんの評価	28
5.2	風の評価構想	28
5.3	各値の設定	29
5.4	評価方法	29
5.5	評価結果	30

第 6 章	おわりに	32
6.1	本研究の主たる成果	32
6.2	今後の課題	33
謝辞		34
参考文献		35

第1章

はじめに

1.1 研究背景

Portable Health Clinic（以下，PHC）は発展途上国農村部における，健康促進のための遠隔医療システムである [1]．ヘルスアシスタントと呼ばれるスタッフが複数の健康測定器具を医者のない農村部に持ち込み，村民に対して健康診断を行う．健康診断の結果，医者からの診断が必要であると判断された患者は都市部にいる医者と電話を通して繋がり，診断を受けることができる．このシステムによって，医者が直接診断できない発展途上国農村部においても人々は診断を受けることができる．このシステムでは，医者は患者を診断しながら症状や処方薬などをノートに取り，通話後にそれをコンピュータに入力して処方箋を作成する．このときにノートに書かれた手書き文字を認識し，その情報を元に処方箋を予測してコンピュータに入力する手間を削減できれば，医者の時間を節約でき，医者はさらに多くの人々の診断を行うことができる．

1.2 研究目的

本研究の目的は，機械学習を行うにあたって，オンライン文字認識におけるデータの拡張方法を確立することである．オープンソースのオンライン文字認識用データセットのうち，医療用語に特化したものは存在していないため，（既存研究 ）では独自にデータ収集を行っている．しかしデータの収集には多くの時間と労力を要するため，（既存研究 ）ではデータ拡張を行ってデータ量を水増している．しかし，既存研究における拡張手法では，単語によっては文字の形が大きく変わってしまい，認識が困難になってしまう場合がある．そのため，この用語認識は学習の際のデータのとり方により認識精度にばらつきが出てしまうという問題がある．

現在，オンライン文字認識の研究においてデータ拡張を行っているものは非常に少なく，オンライン文字認識の拡張方法は未だ確立されていない．機械学習においてデータ量は精度を大きく左右する重要なパラメータである．オフライン文字認識については多くのデータ拡張方法が研究されているが，オンライン文字認識についてもデータの拡張方法について議論がなされるべきである．

1.3 論文構成

本論文の構成は以下の通りである第 2 章では，筆者らが目指しているオンライン手書き医療用語認識，及び本研究に至った取り組むべき課題について説明を行い，既存の対策の問題点をあげる．第 3 章では，問題解決に向け本研究で提案する手法について説明を行う．第 4 章では提案手法の実装について述べる．第 5 章では提案手法の評価について述べる．最後に第 6 章で本論文のまとめと今後の展望を述べる．

第2章

関連技術

本章では，手書き文字認識技術として一般に用いられるオンライン文字認識とオフライン文字認識について説明する．さらに時系列データに対応した機械学習モデルである，再帰型ニューラルネットワークについて説明する．その後再帰型ニューラルネットワークを用いたオンライン文字認識の既存研究と，文字認識におけるデータ拡張に関する既存研究について述べる．

2.1 手書き文字認識の種類

手書き文字の認識は大きく 2 つに分けることができる．オフライン文字認識とオンライン文字認識である．オフライン文字認識は，文字の画像データにおいてそれぞれのピクセルが持つ情報を特徴量として文字を認識する技術であり，手書き文字は認識されるためにイメージスキャナやデジタルカメラによって読みとられる．オンライン文字認識は手書き文字の (x, y) 座標やスピード，筆圧などを特徴量として認識する技術で，通常，タブレットなどに書き込まれた文字が認識される．

オフライン文字認識は画像認識技術であるため，文献 [2] のような畳み込みニューラ

ルネットワーク（Convolutional Neural Network, CNN）を用いた文字認識が多く研究されている。利用可能な画像データもインターネット上に多く存在するが、実際に認識を行う際には手書きされた文字をスキャナやデジタルカメラなどで読み取る必要があるなどのデメリットもある。

一方でオンライン文字認識は IAM On-Line Handwriting Database [3], CASIA Chinese Handwriting Database [4] などのデータセットが存在するが、オフライン文字認識に比べると利用可能なデータは少ない。しかし認識時には、タブレットなどのデバイスに書き込まれた手書き文字のデータが直接使われるため、オフライン文字認識と比べると手間が少ない。また、オンライン文字認識は筆順・筆圧・書き込みにかかる時間などの、オフライン文字認識では使うことができない情報を用いて認識を行うことができるなどのメリットがある。

本研究では、医者の手書き文字は筆記体が多く使われ、画像での認識は難しいと考えられること、実際に使用する際にはスキャンやデジタルカメラによる撮影を行わず、リアルタイムでの認識を想定していることから、オンライン文字認識を用いて医者の手書き医療用語の認識を行う。

2.2 再帰型ニューラルネットワーク

再帰型ニューラルネットワーク（Recurrent Neural Network, 以下 RNN）とは中間層に戻り値のある、音声、動画、文章などの時系列データを扱うニューラルネットワークである。以下で RNN の構造と、再帰型ニューラルネットワークの一種である Long Short-Term Memory（以下、LSTM）について説明する。本研究では LSTM を用いて機械学習を行う。

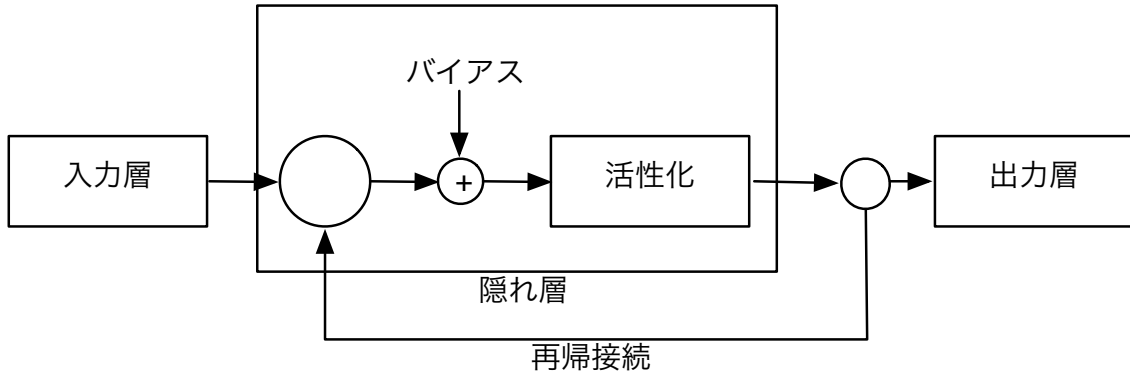


図 2.1: RNN の構造

2.2.1 RNN の構造

図 2.1 に再帰型ニューラルネットワークの構造を示す．RNN は中間層のノードごとに戻り値があるため，現在入力されているデータより前のデータの影響を考慮して計算を行うことができる．そのため RNN は連続的な情報を入力として学習を行うことができる [5]．式 2.1, 式 2.2 に，RNN がそれぞれのノードにおいて行っている計算を入力値 x と時間的順序 t を用いて示す．

$$H(t) = h(W_H x(t) + W_{self} H(t-1) + b_H) \quad (2.1)$$

$$Y(t) = y(W_Y H(t) + b_Y) \quad (2.2)$$

ここにおいて $H(t)$ は隠れ層の出力であり， W_H は隠れ層への入力の重み， W_{self} は戻り値の重み， b_H は隠れ層へのバイアス， $Y(t)$ はネットワークの出力， W_Y は隠れ層の出力の重み， b_Y は出力へのバイアス， $y()$ と $h()$ はそれぞれ出力層と隠れ層の活性化関数である．式より，隠れ層の出力は隠れ層への入力 $x(t)$ だけでなく，隠れ層からの 1 つ前の出力 $H(t-1)$ の影響も受けていることがわかる．したがってネットワークの出力

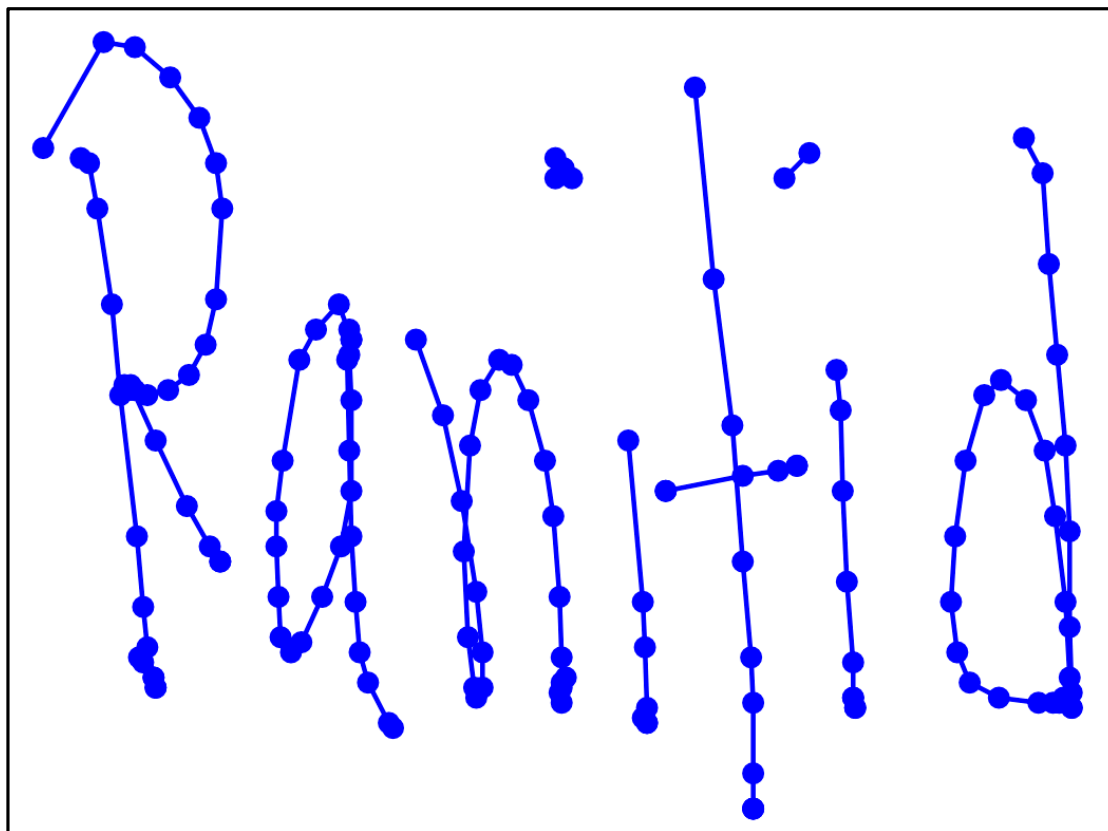


図 2.2: 文字を点で表した図

$Y(t)$ は $x(t)$ と $x(t-1)$ の影響を受けていると言える．この計算は最初をのぞいた全ての入力において行われている．

図 2.2 に認識する手書き文字の例を示す．このように，手書き文字は検出された座標が時間的順序に従って並んだものであると言える．オフライン文字認識では，手書き文字を画像データとして捉えるため，点の時間的順序を考慮に入れずに認識を行う．RNN の入力に連続した手書き文字の点データを用いることで，オンライン文字認識ではその時間的順序を考慮に入れて認識を行うことができる．

しかし，実際に RNN で出力に反映できる過去の入力情報は短く，時系列 10 ステップ

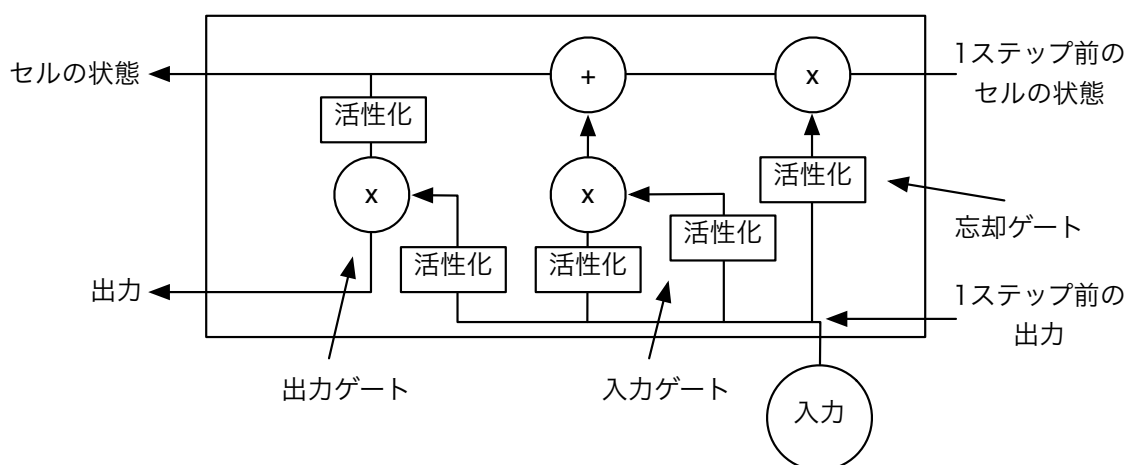


図 2.3: LSTM ブロックの構造

分程度であると言われている [6].

2.2.2 LSTM

図 2.3 に LSTM ブロックの構造を示す. LSTM は RNN よりも長期にわたる記憶を実現するための方法のひとつで, RNN の隠れ層の各ユニットを LSTM ブロックに置き換えたものである. RNN のユニットと異なり, LSTM ブロックは記憶を持ったセルの構造をしている. ゲートと呼ばれる特殊な構造がセルに情報を与えるかセルの情報を除去するかを決める.

忘却ゲートがセル内の情報をリセットし, 入力ゲートは, 入力が現在のセルにどれだけの影響を与えるか決める. 出力ゲートは出力が残りのネットワークにどれだけの影響を与えるか決める. この方法によって LSTM はより長い時系列データにおける再帰型ニューラルネットワークの利用を可能にしている.

2.3 関連研究

ここでは LSTM を用いたオンライン文字認識の既存研究と、文字認識におけるデータ拡張に関する既存研究について述べる。

2.3.1 直線データと BidirectionalLSTM を用いた中国語認識システム [7]

文献 [7] では中国語の手書き漢字の座標データの中から、直線上の点や近接する点など、取り除いても文字として成り立つ点を除去したのち直線データに変換し、BidirectionalLSTM を用いて認識を行っている。BidirectionalLSTM については 3.4 節で述べる。点の除去を行うことで入力データを簡易化し、さらに直線データへの変更を行うことで連続する点同士の関係性（点同士の距離、次の点への角度、同一字画上にあるかなど）を特徴量として抽出することができる。機械学習モデルには BidirectionalLSTM を用いることで、入力されているより過去の座標の情報だけでなく未来の座標の情報も考慮に入れた学習を行うことができる。

2.3.2 dropStroke 手法を用いたデータ拡張と筆者同定システム [8]

文献 [8] では中国語の手書き漢字において、座標の時系列データから一部のストロークを除去する処理を複数回繰り返すことでデータ拡張を行い、CNN を用いたオフライン認識で筆者の同定を行っている。文字としては不完全なデータになるが筆者の特徴を大きく変えることにはならず、100% に近い精度での筆者同定を可能にしている。ただし、ストロークを除去することで異なる文字・単語になってしまう可能性があるため、文字もしくは単語の認識においてこの手法は適切であるとは言えない。

同著者からは、筆者同定のためのデータ拡張として dropSegment 手法というものも提案されているが [9]，これも異なる文字・単語になってしまう可能性がある．また，この手法もオフライン認識のためのデータ拡張であるため，オンライン文字認識のためのデータ拡張手法として適切であるとは言えない．

2.3.3 遠隔医療システムにおける

処方箋予測に向けた手書き医療用語認識に関する研究

文献（ ）では、処方箋の予測に向けたシステムの初期研究として、再帰型ニューラルネットワークを用いたオンライン手書き医療用語認識手法の提案を行っている．また、オンライン手書き文字のデータ拡張手法として、ストロークの回転と平行移動によってデータ量を水増しする SRP (Stroke Rotation and Parallel-shift) 手法を提案している．

(ここ全く同じ) 手書き文字データは座標の時系列データとして捉えることができる．そのデータから直線上の点と近接する点を除去して機械学習への入力を簡易化し、さらに点データを直線データに変更して特徴量の抽出を行っている．文献（ ）ではバングラデシュでの処方箋予測に向けた実装及び評価を行うが、現在医療用語に特化したデータセットがオープンソースとして存在しない．そこで、Portable Health Clinic における過去の処方箋データから頻出する単語を座標の時系列データとして収集し、再帰型ニューラルネットワークの一種である BidirectionalLSTM を用いて学習を行っている．

様々な手書き文字に対応するため、文字認識では多様な提供者から大量のデータを得る必要がある．しかし、十分なデータ量を確保するためには非常に多くの労力と時間を要する．そこで筆者らは SRP 手法を提案している．

収集した 15991 語のデータを用いて, SRP 手法でデータ拡張を行った後に BidirectionalLSTM で学習を行った結果, 480 語のクラスにおいて 89.5%の精度で単語を認識している. この結果はデータ拡張を行わなかった場合の認識精度と比べて 16.1%高い. しかし, ストロークの回転と平行移動を同時に行うため, 単語によっては拡張手法で文字の形が大きく変わってしまい, 認識が困難になってしまう場合がある. そのため, この用語認識は学習の際のデータのとり方により認識精度にばらつきが出てしまうという問題がある

第3章

LSTMを用いた手書き医療用語認識と SRP手法及びRatio手法によるデータ 拡張

本章では，先行研究の BidirectionalLSTM を用いたオンライン手書き医療用語認識を説明する．また，オンライン文字認識用のデータ拡張手法として先行研究の SRP 手法を説明し，新手法として Ratio 手法を提案する．

3.1 システムの概要

図 3.1 にシステムの概要を示す．提案システムでは，タブレットから収集した手書き医療用語の座標データに対して前処理を行う．その後学習プロセスでは前処理後のデータに SRP 手法を用いたデータ拡張を適用し，機械学習における学習データとする．推定プロセスでは，学習を行ったモデルに前処理後のデータを入力し，用語を推定する．

ニューラルネットワーク構造とデータ前処理，特徴量抽出に関しては2.3.1項の文献 [7]

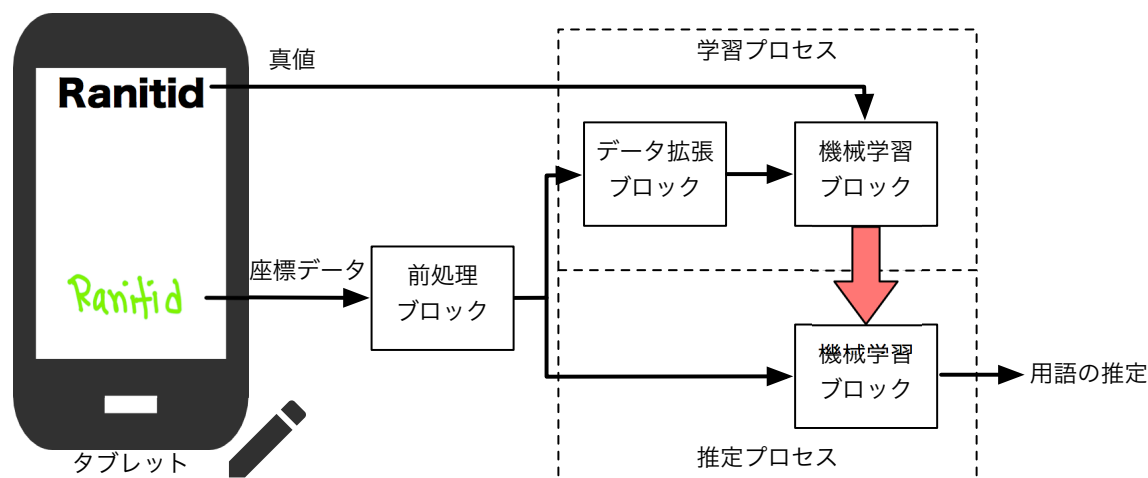


図 3.1: 提案システムの概要

を参考にする。以下、各手順について述べる。

3.2 前処理ブロック

図 3.2 に前処理ブロックの概要を示す。前処理ブロックでは、点の除去、正規化、特徴量抽出を行う。直線上の点や近接する点といった、取り除いても文字として成り立つような点の除去を行い、その後正規化を行う。特徴量抽出では点データを直線データへ変換する。以下、前処理ブロックにおける各処理を示す。

3.2.1 点の除去

この処理では、各単語におけるデータの提供者ごとの点の数の差を小さくするため、取り除いても文字として成り立つような点の除去を行う。本研究では直線上の点と近接する点の 2 種類を除去する。

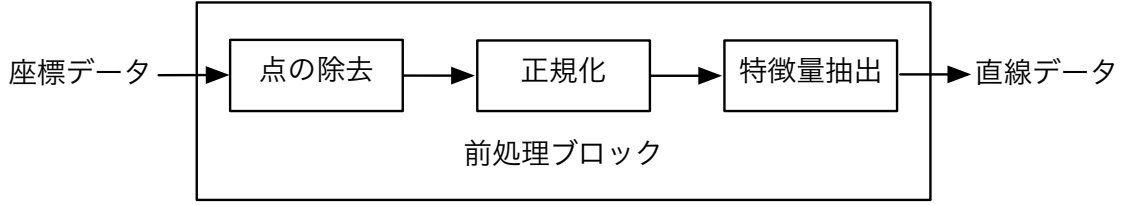


図 3.2: 前処理ブロックの概要

収穫する医療用語手書きデータは、図 2.2 で示したように (x, y) 座標の時系列データとして存在している。本研究では文献 [7] に従って、これに各点の筆順情報 s を合わせた (x, y, s) を収集する。1 つの単語を式 3.1 のように収集する。

$$[[x_1, y_1, s_1], [x_2, y_2, s_2], \dots, [x_n, y_n, s_n]] \quad (3.1)$$

x_i と y_i は点の座標、 s_i はその点は何画目のストローク上にあるかを示したものである。

手書きデータは書くスピードの違いなどが原因で、同じ単語でもデータ提供者によって点の数が大きく異なってしまい、うまく認識ができない可能性がある。取り除いても文字として成り立つような点の除去を行うことでデータ提供者ごとの点の数の差を小さくすることができる。

本研究で除去する点は 2 種類ある。1 つ目は直線上の点である。図 3.3(a) に除去する点のイメージ図を示す。点 i が点 $i-1$ と点 $i+1$ の直線上にある場合、点 $i-1$ から点 i を通って点 $i+1$ に向かう線は i を除去しても直線として成り立つため、 i を除去する。 $\Delta x_i = x_{i+1} - x_i$, $\Delta y_i = y_{i+1} - y_i$ として式 3.2 を用いてコサインの値を算出し、その値が閾値 T_{cos} より大きい場合、点 i を除去する。

$$\frac{\Delta x_{i-1} \Delta x_i + \Delta y_{i-1} \Delta y_i}{(\Delta x_{i-1}^2 + \Delta y_{i-1}^2)^{0.5} (\Delta x_i^2 + \Delta y_i^2)^{0.5}} > T_{cos} \quad (3.2)$$

図 3.3(b) にオリジナルの単語データの例と、図 3.3(c) に直線上の点が除去された後の単語データの例を示す。

2つ目は近接する点の除去である．図 3.3(a) にイメージ図を示す．点 i が点 $i-1$ と極端に近接している場合，点 $i-1$ から点 i を通って $i+1$ に向かう線は点 $i-1$ から点 $i+1$ までの直線と概ね等しくなるため，点 i を除去する．直線上の点の除去を行った後，式 3.3 を用いて 2 点間の距離を算出し，その距離が閾値 T_{dist} より小さい場合，点 i を除去する．

$$\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} < T_{dist} \quad (3.3)$$

図 3.3(d) に近接する点が除去された後の単語データの例を示す．

3.2.2 正規化

入力データをさらに簡潔にするため，点の除去を施したデータの正規化を行う． x 座標について説明する．全てのデータの中から最大値 x_{max} と最小値 x_{min} を取り出し，式 3.4 を用いてある点の x 座標 X を X_{nor} に正規化する．

$$X_{nor} = \frac{X - x_{min}}{x_{max} - x_{min}} \quad (3.4)$$

y 座標に対しても同様の計算を行い，結果的に (x, y) をそれぞれ 0 以上 1 以下のデータとする．

3.2.3 特徴量抽出

機械学習への入力のため，特徴量を抽出する．正規化した点データ 2 点間を繋ぎ，直線データを作成する．直線データに変換することで，点の座標だけでなく直線の長さ，角度などより多くの特徴量を抽出することができる．線データ L_i は，点 i と点 $i+1$ から式 3.5 のように構成され，本研究ではこのデータを学習における入力データとする．

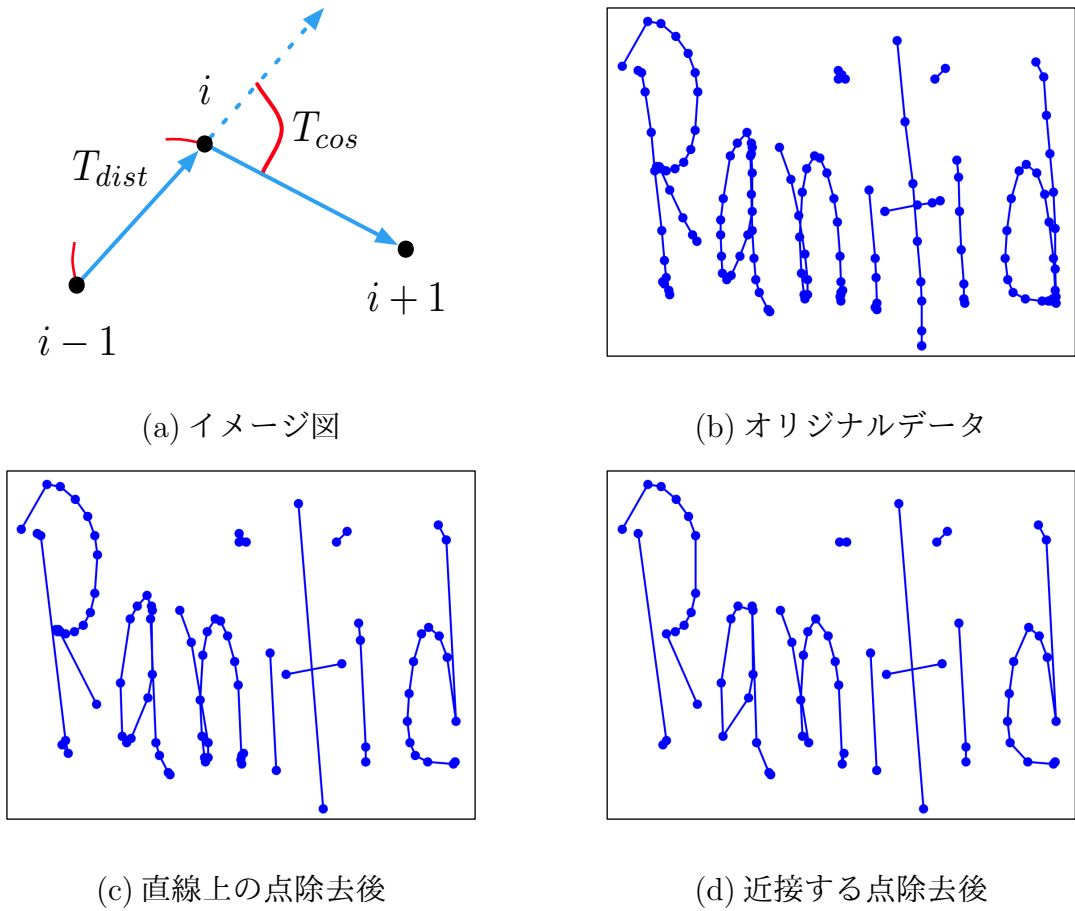


図 3.3: データ前処理：余分な点の除去

$$L_i = [x_i, y_i, \Delta x_i, \Delta y_i, I(s_i = s_{i+1}), I(s_i \neq s_{i+1})] \quad (3.5)$$

ここで, $\Delta x_i = x_{i+1} - x_i$, $\Delta y_i = y_{i+1} - y_i$ であり, $I()$ は括弧内の条件が真であるときに 1 でありそれ以外では 0 である.

L_i において x_i と y_i は直線の始点を表し, Δx_i と Δy_i は線の始点から終点までの x 軸方向, y 軸方向の距離を表す. また $I(s_i = s_{i+1}) = 1$ は, 直線の始点と終点が同一ストローク上にあることを示し, $I(s_i \neq s_{i+1}) = 1$ はその直線で次のストロークに移ったことを示す.

3.3 データ拡張ブロック

様々な手書き文字に対応するため、本研究では多様な提供者から大量のデータを得る必要がある。しかし、十分なデータ量を確保するためには非常に多くの労力と時間を要する。そこで、はじめに先行研究のSRP (Stroke Rotation and Parallel-shift) 手法を説明する。その後、文字の縦横比を変更するRatio手法を提案する。これらの手法を用いてデータ拡張を行うことで、手書き文字データの多様性を高める。

3.3.1 ストロークの回転

ストロークの始点と終点の座標から中点を求め、その点を中心にストローク上の点をそれぞれ回転させることでストローク全体を回転させる。この処理を1つのデータに対して複数回施すことで、データ拡張を行う。

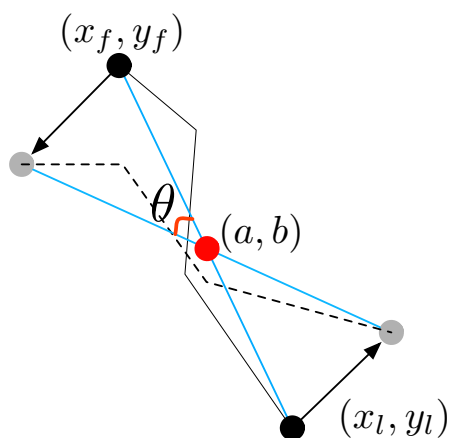
図 3.4(a) にストロークの回転の原理を示す。ストロークの始点を (x_f, y_f) 、終点を (x_l, y_l) とする。始点と終点の中点を (a, b) とすると、式 3.6 より (a, b) の値が求められる。

$$(a, b) = \left(\frac{x_f + x_l}{2}, \frac{y_f + y_l}{2} \right) \quad (3.6)$$

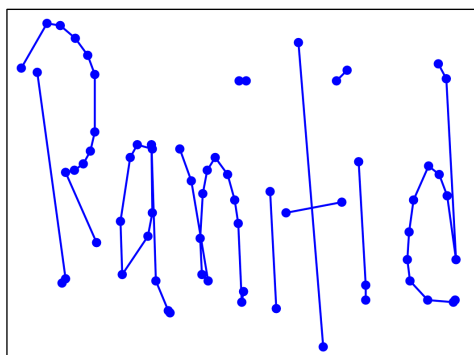
ストローク上の任意の点の座標を (x, y) としたとき、その点を、点 (a, b) を中心に角 θ だけ移動させた後の座標 (X, Y) は 式 3.7 で表される。

$$\begin{pmatrix} X - a \\ Y - b \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x - a \\ y - b \end{pmatrix} \quad (3.7)$$

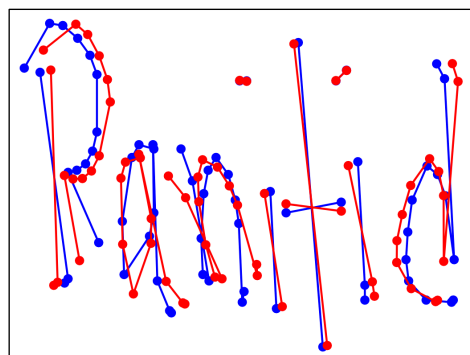
この式をストローク上のすべての点に用いることで、ストローク自体を角 θ 回転させ



(a) 回転の原理



(b) データ前処理後



(c) 回転前（青）と回転後（赤）

図 3.4: ストロークの回転

る．図 3.4(b) にストローク回転前の単語データの例，図 3.4(c) にストローク回転後の単語データの例を示す．この処理を，ストロークごとに角度を変えながら行うことで元のデータとは異なる形の文字・単語を生成する．それを 1 つの単語データに対して N 回行い，データ量を N 倍に拡張する．

3.3.2 ストロークの平行移動

ストローク上の点の座標それぞれに一定の値を加え，ストローク全体を平行移動させることでデータ拡張を行う．図 3.5(a) にストロークの平行移動の原理を示す．ストローク上の任意の点の座標を (x, y) としたとき，その点を x 方向に dx ， y 方向に dy だけ平行移動させた後の座標 (X, Y) は 式 3.8 で表される．

$$(X, Y) = (x + dx, y + dy) \quad (3.8)$$

この式をストローク上のすべての点に用いることで，ストローク自体を x 方向に dx ， y 方向に dy 平行移動させる．図 3.5(b) にストローク平行移動前の単語データの例，図 3.5(c) にストローク平行移動後の単語データの例を示す．この処理を，ストロークごとに dx と dy の値を変えながら行うことで元のデータとは異なる形の文字・単語を生成する．それを1つの単語データに対して N 回行い，データ量を N 倍に拡張する．

3.3.3 文字の縦横比変更

(まだ全然かけてない，なんとなくで書いておこう) 1つの単語における，すべての点の y 座標 (もしくは x 座標) の平均値を取り，すべての点の y 座標 (もしくは x 座標) に対し (ここになんか書くぞ)．図に文字の縦横比変更の原理を示す．文字の拡張比率を a ，1つの単語上のすべての点の y 座標の平均を \bar{y} とし，同単語上の任意の点の座標を (x, y) としたとき，その点の y 座標が \bar{y} より大きいときに y 方向にー移動させた後の座標 (X, Y) は 式 3.9 で表される．

$$(X, Y) = (x \times 1, y \times (1 + a)) \quad (3.9)$$

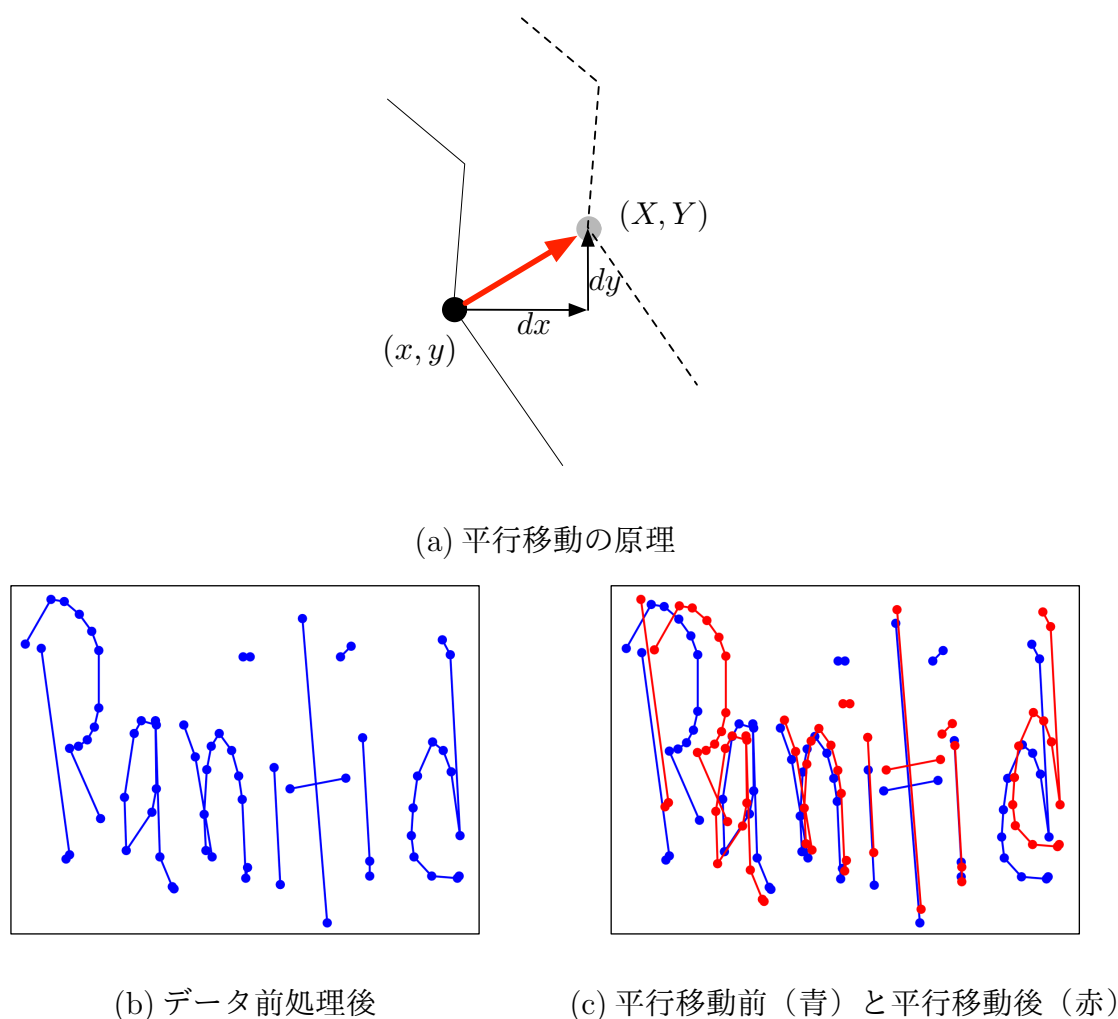


図 3.5: ストロークの平行移動

この式を単語上のすべての点に用いることで、単語全体の点を一斉に移動させる。

．図 3.6(b) にストローク平行移動前の単語データの例，図 3.6(c) に文字の縦横比率変更後の単語データの例を示す．この処理を，拡張回数ごとに a と x, y どちらに適応するかをランダムに変えながら行うことで元のデータとは異なる形の文字・単語を生成する．それを 1 つの単語データに対して N 回行い，データ量を N 倍に拡張する。

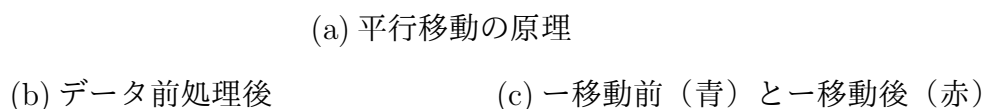


図 3.6: ストロークの平行移動

3.4 機械学習ブロック

本研究では BidirectionalLSTM を用いて学習を行う。図 3.7 に BidirectionalLSTM の概要を示す。BidirectionalLSTM は従来の LSTM に未来の入力から計算を行う逆方向のモデルを加え、出力を同一の出力層に統合するものである。

学習プロセスでは、データ拡張が施された直線データを入力として用いる。本研究において BidirectionalLSTM は、現在入力されている直線データより前に書かれた直線データだけでなく、後に書かれる直線データも用いて学習を行う。

推定プロセスでは、学習が行われたモデルに前処理後のデータを入力し、用語の推定を行う。

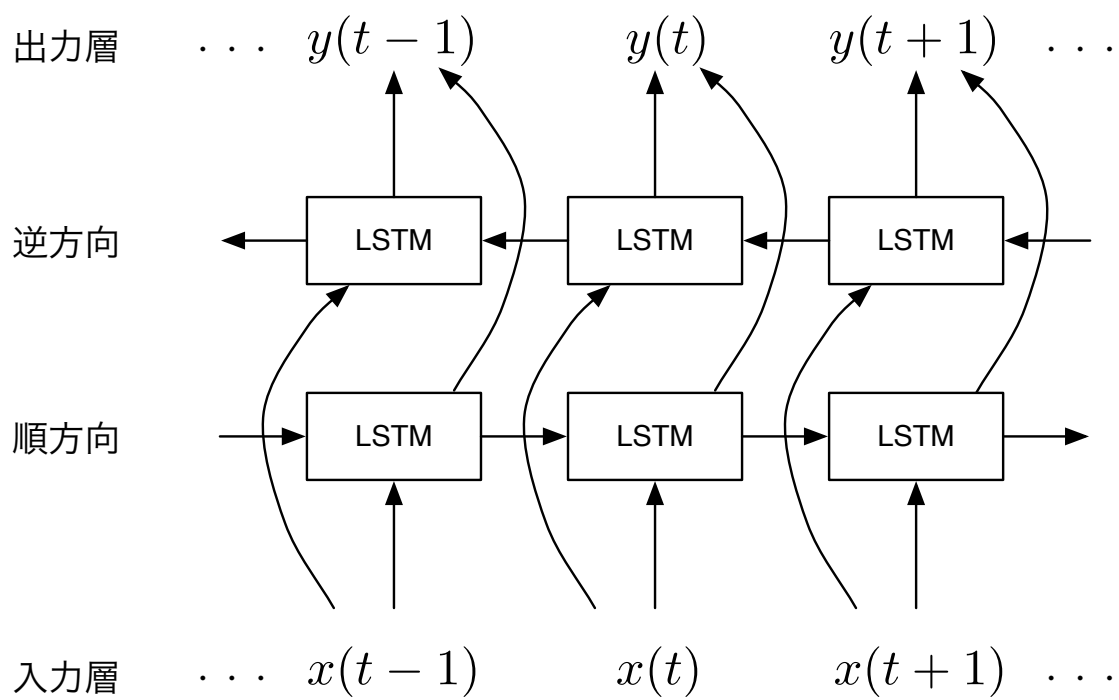


図 3.7: BidirectionalLSTM の概要

第 4 章

実装

本章では，データ収集の方法について説明し，SRP 手法と LSTM を用いたオンライン手書き医療用語認識手法の実装について説明する．

4.1 データ収集

本研究では，学習に使用したいデータがオープンソースで存在していないため，既存研究において独自に収集しているデータを用いる．以下収集したデータの内容，収集手順を説明する．

4.1.1 医療用語コーパス

過去の処方箋データから医療用語コーパスを作成した．コーパスとは，言語を分析するための基礎資料として書き言葉や話し言葉の資料を収集し，研究用の情報を付与したものである．本研究では医療用語の手書き認識を行う．そこで 1.1 節で述べた PHC の過去の処方箋データからコーパスを作成し，それを機械学習の正解データとして用い

SL	CC	Drugs				Advices	Test
1	Low back pain,	Tab - Naprosyn - 500 mg - 1+0+1	After meal		5 Days	Previous rules will continue to medication, Do not work long being bent , Use high commode , Take plenty of water , Avoid lifting weights, Please	S.creatinine ,
		Tab - Beklo - 5 mg - 1+0+1	After meal		5 Days		
2	headache, low back pain, acidity, constipation.	Tab - Neoeptin-R - 150 mg - 1+0+1	Before meal		15 Days	Take plenty of water , Do not work long being bent , Avoid lifting weights, Avoid foods that are high in fat and contain higher amount of oil and spices, Please consult with Eye specialist.	
		Tab - Ace - 500 mg - 1+1+1	After meal		3 Days		
		Tab - Calbo D - - 0+0+1	After meal		Cont.		
		Syp - Avolac - - 1 tea spoon	After meal	3 times -- if constipation.			
3	DM on insulin					the Diabetic food-chart, Please check blood sugar fasting and before dinner. then adjust the dose of insulin.	

図 4.1: PHC の処方箋データ（一部）

```

1 426,Ostocal D
2 381,Maxpro
3 376,Ranitid
4 367,Napa
5 322,Ferocit
6 285,Neuro B
7 278,Comet
8 261,Omidon
9 261,Multivit plus
10 261,Calbo D
11 255,Amdocal
12 226,Neoeptin
13 223,Tryptin
14 215,Rivotril
15 213,Enavit

```

図 4.2: 薬名欄のコーパス（一部）

た． 図 4.1 にコーパス作成に使用した PHC の過去の処方箋データを示す．8324 名分の過去の処方箋データは，それぞれの欄が症状，薬名などに分けられる．図 4.2 に本研究で作成したコーパスの例として，薬名欄のコーパスの一部を示す．それぞれの欄において全ての文を 1 単語ごとに区切り，単語の出現回数を数えて単語を並び替えた．

本研究では薬名欄に頻出する単語（360 語，英語）と医者からのアドバイス欄に頻出する単語（120 語，バングラ語）を用いてコーパスを作成し，データ収集を行った．



図 4.3: データ収集アプリの構造

4.1.2 データ収集用アプリ

図 4.3 にデータ収集用アプリの構造を示す。オンライン手書き文字を収集するため、株式会社 CodeNext [10] に Android アプリの作成を依頼した。このアプリは 4.1.1 項で作成したコーパスから単語を 1 語ずつ表示し、データ提供者は表示された単語をタブレットに手書きで記入する。手書きされた文字は正解データとともにデータベースに保存される。図 4.4 にアプリ画面のイメージ図を示す。

4.2 使用機器

図 4.5 に使用機器を示す。データ収集用のタブレットは Samsung 社の Galaxy Tab S3 を用いた。1 つのタブレットに 1 つのスタイラスペンが付属している。機械学習には NVIDIA 社の GeForce 1080 が 1 枚搭載された デスクトップ PC を用いた。表 4.1 に実装環境を示す。

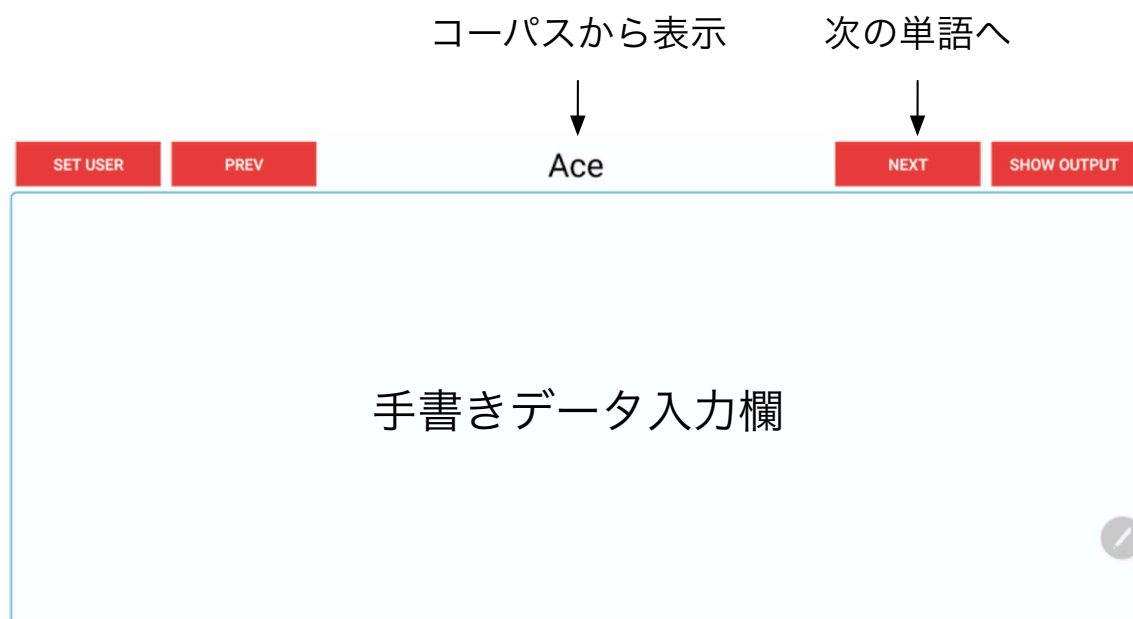


図 4.4: データ収集アプリイメージ図

表 4.1: 実装環境

OS	Ubuntu 16.04
GPU	NVIDIA GeForce 1080
メモリ	8GB
プロセッサ	3.70GHz Intel Core i7-8000K
タブレット	Samsung Galaxy Tab S3

4.3 学習モデル構造

図 4.6 に本研究で用いた学習モデルの構造を示す．文献 [7] を参考に作成し，実装には python のニューラルネットワークライブラリである Keras [11] を用いた．本研究で用いるデータはデータ長が一定ではないため，データの後ろを 0 でパディングすること



図 4.5: 使用機器

でデータ長最大の値である 260 に揃え，学習の入力とした．LSTM 層の出力次元数は 300 に設定し，プールサイズの値の平均値をそれぞれ出力するプーリング層を配置した．その後パラメータの数を増やすために全結合層を配置した．なお，過学習を防ぐためプーリング層と全結合層の間にドロップアウト [12] をそれぞれ 0.3 の割合で設定した．

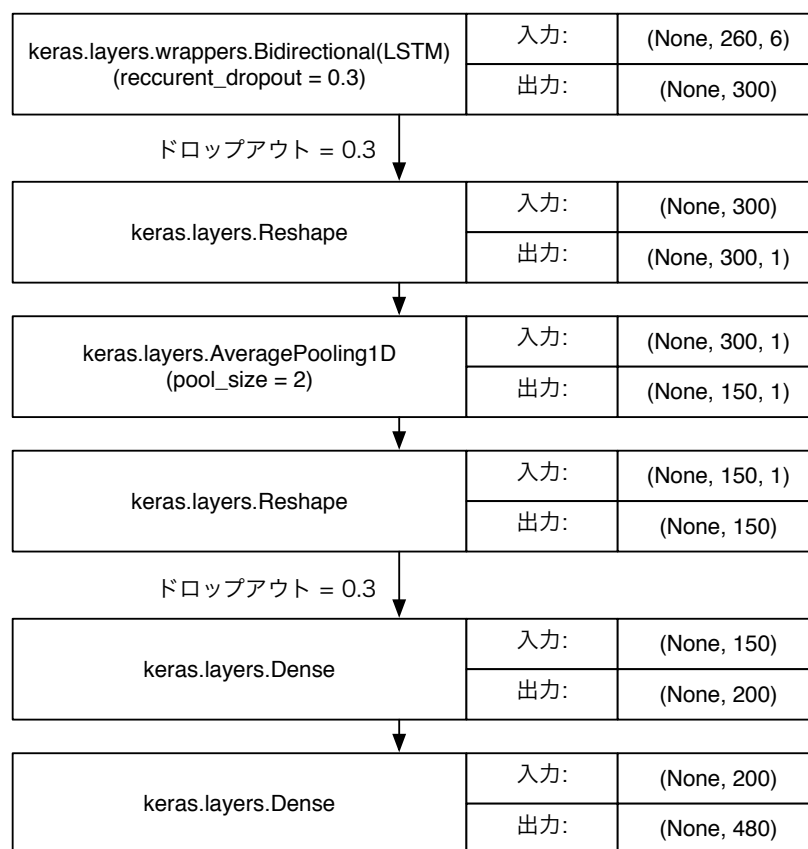


図 4.6: 層の構成

第5章

評価

本章では，4.1 節で 39 名から収集した 15911 語を用いて行った評価の概要と，その結果を示す．

5.1 高橋さんの評価

元データに対し，データ拡張なし・データ拡張あり (100 倍) の 2 パターンで精度を比較している

5.2 風の評価構想

[やったこと]SRP と Ratio で 125 倍，SR と Ratio で 100 倍，PS と Ratio で 100 倍で拡張 [やること] 単純に高橋さんとの精度の比較を行う．さらに関連研究（中国語）の元データを 1/100 にしデータを 100 倍して，データ数を同じにした．という条件のもとで拡張の有効性を検証する．

表 5.1: データ拡張時の値

データ拡張の種類	μ	σ^2
ストロークの回転移動	0	25
ストロークの平行移動	0	1.0×10^{-4}

5.3 各値の設定

データ前処理における 2 つの閾値は、それぞれ $T_{cos} = 0.99$, $T_{dist} = 0.005 * \max(H, W)$ とした。ここで H はデータ提供者がデータを入力する場所の縦幅、 W は横幅を示している。

表 5.1 にデータ拡張時の各値を示す。回転でのデータ拡張では、ガウス分布を用いて平均 $\mu = 0$ 、分散 $\sigma^2 = 25$ となるような乱数の角度をストロークごとに生成し、ストロークの回転を施した。この処理を 1 つの単語に対して 10 回行った。平行移動でのデータ拡張では、ガウス分布を用いて平均 $\mu = 0$ 、分散 $\sigma^2 = 0.0001$ となるような乱数の dx と dy を各ストロークごとに生成し、ストロークの平行移動を施した。この処理を、回転でのデータ拡張を行った単語に対してそれぞれ 10 回行った。

ガウス分布のパラメータは、それぞれ文字の形を崩さないと筆者が判断できるもののうち最大の値を設定した。SRP 手法でのデータ拡張を行うことで、データ量を元の量の 100 倍とした。

5.4 評価方法

データ提供者 39 名の中から、データ提供が途中で止まっていた 12 名を除いた 27 名のうちから 3 名をランダムで抜き出し、その 3 名のデータをテスト用のデータとした。残りの 24 名と 12 名のデータを学習用のデータとした。その後学習用データに対して

データ拡張を行い、検証用データと 9 : 1 の割合で分けた後機械学習を行った。学習用データにはテストに用いられるデータ提供者 3 名のデータは一語も含まれていない。この過程を 10 回繰り返し、各回の精度の平均値、中央値などを求めることで、本研究の手書き医療用語認識の有効性を測る。さらにデータ拡張を行った場合のテスト結果と、行わなかった場合のテスト結果を比較することで、SRP 手法を用いたデータ拡張の有効性を測る。

5.5 評価結果

表 5.2 に評価の結果を示す。データ拡張を行った際はエポック数を 5、ミニバッチサイズを 512 に設定し、データ拡張を行わなかった場合はエポック数を 100、ミニバッチサイズを 64 に設定した。エポック毎に異なるミニバッチを作成するため、エポック毎に学習データをシャッフルした。過学習を防ぐため、どちらにも keras の EarlyStopping [13] を用いた。最適化関数には Adam [14] を用い、学習率は 0.001 とした。

結果として、データ拡張を行わなかった場合の平均精度が 73.4%であったのに対し、データ拡張を行った場合の平均精度は 89.5%であった。データ拡張を行わなかった場合でも比較的高い精度が出ていたが、テストデータの取り方によるばらつきが非常に大きかった。一方でデータ拡張を行った場合は全体として精度が安定していた。分散、標準誤差もデータ拡張を行った場合の方が小さかった。データ拡張を行わなかった場合の方が過学習を起こす回数が多かったことが原因としてあげられる。過学習を起こす回数に違いが出た理由としては、同じデータをエポック数を増やして何度も学習したためデータの多様性が低くなったことがあげられる。

表 5.2: 評価結果

評価対象	精度 [%] (平均値)	精度 [%] (中央値)	分散	標準誤差
データ拡張なし	73.4	83.7	5.3×10^{-2}	2.3×10^{-1}
データ拡張あり	89.5	91.4	5.7×10^{-3}	7.5×10^{-2}

第6章

おわりに

6.1 本研究の主たる成果

本研究では、遠隔医療における処方箋予測に向けた、BidirectionalLSTM を用いたオンライン手書き医療用語認識手法の提案と、オンライン文字認識用のデータ拡張手法である SRP(Stroke Rotation and Parallel-shift) 手法・ChangeRatio 手法の提案を行った。独自にデータ収集を行う必要があったため、コーパスとして PHC の過去の処方箋データを用いることで処方箋に頻出する医療単語を抽出しデータ収集を行った。また、様々な手書き文字に対応するため、ストロークの座標を移動させることによるデータ拡張 (SRP 手法)・———— (ChangeRatio 手法) を行った。拡張したデータを入力として BidirectionalLSTM を用いて評価を行った。

評価の結果、収集した 15991 語のデータと 480 語のクラスにおいて====の精度で単語を認識した。この結果はデータ拡張を行わなかった場合の認識精度と比べて====高かった。

6.2 今後の課題

謝辞

参考文献

- [1] A. Ahmed, A. Rebeiro-Hargrave, Y. Nohara, R.I. Maruf, P.P. Ghosh, N. Nakashima, and H. Yasuura, “Portable health clinic: A telehealthcare system for unreached communities,” *Smart sensors and systems*, pp.447–467, Springer, 2015.
- [2] A. Yuan, G. Bai, L. Jiao, and Y. Liu, “Offline handwritten english character recognition based on convolutional neural network,” *Document Analysis Systems (DAS)*, 2012 10th IAPR International Workshop onIEEE, pp.125–129 2012.
- [3] FKI, “Iam on-line handwriting database”. <http://www.fki.inf.unibe.ch/databases/iam-on-line-handwriting-database>.
- [4] C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang, “Casia online and offline chinese handwriting databases,” *Document Analysis and Recognition (ICDAR)*, 2011 International Conference onIEEE, pp.37–41 2011.
- [5] B. Chakraborty, P.S. Mukherjee, and U. Bhattacharya, “Bangla online handwriting recognition using recurrent neural network architecture,” *Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image ProcessingACM*, p.63 2016.

- [6] 岡谷貴之, 深層学習, 講談社, 2015.
- [7] X.-Y. Zhang, F. Yin, Y.-M. Zhang, C.-L. Liu, and Y. Bengio, “Drawing and recognizing chinese characters with recurrent neural network,” IEEE transactions on pattern analysis and machine intelligence, vol.40, no.4, pp.849–862, April 2018.
- [8] W. Yang, L. Jin, and M. Liu, “Chinese character-level writer identification using path signature feature, dropstroke and deep cnn,” ICDAR, pp.546–550, 2015.
- [9] W. Yang, L. Jin, and M. Liu, “Deepwriterid: An end-to-end online text-independent writer identification system,” IEEE Intelligent Systems, vol.31, no.2, pp.45–53, 2016.
- [10] “株式会社 codenext”. <https://codenext.jp>.
- [11] “Keras”. <https://keras.io/ja/>.
- [12] keras, “Core レイヤー”. <https://keras.io/ja/layers/core/>.
- [13] keras, “コールバック”. <https://keras.io/ja/callbacks/>.
- [14] D.P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, pp.●●–●●, 2014.