修 士 学 位 論 文

題　　目

深層学習による灯謎問題の正解推定システムの構築

主査　　森 直樹 教授

副査　　藤本 典幸 教授

副査　　黄瀬 浩一 教授

令和　4　年（　2022　年）度修了

（No.　2210104046　）　　陳 偉斉

大阪府立大学大学院工学研究科
電気・情報系専攻 知能情報工学分野

目 次

図目次

表目次

1　はじめに

深層学習の発展により, 人工知能 (Artificial Intelligence: AI) による漫画や小説やクイズなどの人類の創作物への理解, 生成といった分野の研究が盛んである. しかし, 技術力の制限により、コンピュータで人類の創作物を理解することはまだ困難である. 本研究では中国の伝統的クイズ「灯謎 (トウメイ)」に注目し, 深層学習による灯謎の理解を目的とする.

漢字は表意文字であり, 1 つの文字に音, 形, 義 (意味) の情報が隠されている. 灯謎はこれらの情報で作成され, 自然言語処理の知識と強く関わる同時に, 画像処理との関連性もなくはない. そのため, 人工知能による灯謎の研究は中国語における言語処理に一定な価値があると考えられる. その一方, 人工知能を灯謎に用いた研究は僅かである. そのため, 本研究では灯謎の問題により正解と人為的に作た不正解を推定することで人工知能による灯謎を理解する可能性を探索する. 漢字の画の情報を分類モデルに導入することで, 漢字の形の情報である画情報の有効性を検証する. 実験用データセットには中国の灯謎を最も数多く集めるデータベースである中華灯謎データベースを利用し, 漢字の画情報を用いた分類モデルで灯謎の正解を推定した.

以下に本研究の構成を示す. 2 章では中国伝統的クイズゲーム「灯謎」について述べる. 3 章で本研究の要素技術について述べる. 4 章で漢字の形の情報における関連研究について述べる. 5 章で本研究に構築した研究用データセットについて紹介する. 6 章で本研究の提案手法である灯謎問題の正解推定モデルについて詳述する. 7 章で提案手法の有効性を確認するために実施した数値実験について述べる. 8 章でまとめと今後の課題について述べる.

2 灯謎

本章では, 本研究で扱う中国の伝統的クイズ「灯謎 (トウメイ)」について述べる.

2.1 灯謎の歴史

灯謎とは中国の伝統的クイズであり, 深い歴史がある. 灯謎の起源は, 中国の春秋時代にさかのぼられる. 当時, 文人たちは「隠語 (インゴ)」というクイズを作成し諸侯に自分の意見を述べる. 秦漢の時代以降, 隠語は次第に民衆の生活に入り込み, 娯楽の手段として広まっていた. 初めて灯謎と呼ばれ, 灯籠と関わるという形式で知られるのは宋の時代である. 呉の研究は灯謎の宋代以後の歴史を述べていた [8]. 当時の人は灯謎を作成して灯の上に張り, 回答者は謎底を当てる. 当たりの回答者は景品と灯謎を記載した紙を貰える. その時期, 灯謎は巧妙な思考と暗示を含まれ、当てるのは虎を撃つと同じくらい難しいため,「射虎」とも呼ばれたことある. 明清時代に灯謎は流行になる.「紅楼夢」や「鏡の端」など有名な小説は当時の光景を描いている. 現在, 中国では中華灯謎学会という全国的な愛好家団体があり, 国際的な灯謎大会を毎年のように開催されている. 現代における灯謎愛好者のコミュニケーションネットワークは清末民国期に比べて更に広い範囲に及ぼしているものの, それらの交流活動の殆どはアカデミックから離れており, 民間団体によって加担されている.

2.2 灯謎のパターン

灯謎は「謎面」,「謎目」と「謎底」で構成されている. 謎面は問題に当たり, 主に詩や熟語など一見意味不明な短い文で作成される. 謎底は答えに当たり, 主に漢字や単語などで作成される. 謎目はヒントに当たり, 謎底の特徴を説明し, 正解の推定を支援することは目的である. 灯謎の作成は「謎格」というルールを従う必要である. 灯謎の謎格により「発音で解く灯謎」,「字形で解く灯謎」,「意味で解く灯謎」3 種類に分けられる. 図 2.1 に各種類の灯謎の例を示す.

謎面: 默読
謎目: 打一学科名
謎底: 心理学

a. 発音で解く灯謎

謎面: 一百減一
謎目: 打一字
謎底: 白

b. 字形で解く灯謎

謎面: 今天
謎目: 打一国名
謎底: 日本

c. 意味で解く灯謎

図 2.1: 各種類の灯謎の例

例 a は発音で解く灯謎の例であり, 謎面は「声を出さないで読むこと」の意味であり, 謎目は「答えは学科名である」の意味である. 「声を出さないで読むこと」は「在心里学習 (心の中に勉強する)」を指し, そして「心里学」と「心理学」は発音が同じであるため, 故に謎底は学科「心理学」とある.

例 b は字形で解く灯謎の例であり, 謎面は「百減一は何」の意味であり, 謎面は「答えは 1 文字である」の意味である. 「百減一」は「漢字百の上の一を削る」を指し, 謎底は漢字「白」となる.

例 c は意味で解く灯謎の例であり, 謎面は「今日」の意味であり, 謎面は「答えは国の名前である」の意味である. 「今日」は「本日」を指し, 左右交換すると謎底は日本「白」となる.

例から見ると, 発音で解く灯謎は謎面と「同音字 (同音異義語)」と「多音字 (同形異音語)」の関連性で謎底を得る灯謎である. 字形で解く灯謎は謎面の漢字の分解, 変換, 増減, 組合で謎底を得り, 最も簡単な灯謎である. 意味で解く灯謎は謎面の内容に関する知識や常識を利用して謎底を得り, 最も困難な灯謎である. 本研究は人工知能で灯謎を解く可能性を確認するため, 最も簡単で数多くの灯謎である「字形で解く灯謎」を研究対象とする.

2.3 灯謎に関する人工知能の考案

本研究では字形で解く灯謎のうち,「字謎」と呼ばれる答えが 1 文字となる種類のみについて考える. 字謎の答えは, 単語の意味に加えて漢字の形も強く関わるので, 単純に大量な問題の文をニューラルネットワークで学習しても効果が薄いと予想される. そこで本研究では灯謎と漢字の形の情報における関連性に着目し, 漢字の「画」成分を利用した Long short-term memory (LSTM)[9] モデルで灯謎問題の正解推定システムを構築した.

3　要素技術

本章では, 本研究の要素技術について述べる.

3.1　五筆字型入力方法

五筆字型入力方法 [10] は王が 1983 に発表した中国漢字入力方法である. 五筆字型入力方法は中国語における漢字の使用頻度や入力方法を参考し, 漢字の画を「横棒」,「縦棒」,「左払い」,「点」,「鉤」に分類する. たとえば漢字「五」は「横・竪・折・横」なので「12510」で「五」が入力できる. このコーディング方法は「五筆コート」とも称される.

図 3.1: Levenshtein 距離の計算の過程

3.2　Levenshtein 距離

Levenshtein 距離 [11] は 2 つの文字列がどの程度異なっているかを示す距離の一種である. 具体的には, 1 つの文字列からもう一方の文字列に変形するのに必要な手順 (1 文字の挿入, 削除, 置換) の最小編集回数を計算する. 実際の例として, 文字列「JULY」を「JELLY」に変形する場合, 最低「JULY から JELY に変形 (U を E に置換)」と「JELY から JELLY に変形 (L の挿入)」である 2 回の手順が必要となり, Levenshtein 距離は 2 となる.

Levenshtein 距離を計算する際, 一般的動的計画法によるアルゴリズムが用いられる. 図 3.1 に例の Levenshtein 距離の計算の過程を示す. Algorithm 1 に Levenshtein 距離のアルゴリズムを示す.

Algorithm 1 Levenshtein 距離のアルゴリズム

Input: (a, b: input strings)

Output: The Levenshtein distance between a and b

$lena \leftarrow |a|$, $lenb \leftarrow |b|$

$d \leftarrow \text{array}(0, \ldots, lena)$ of $\text{arrays}(0, \ldots, lenb)$

for $i \leftarrow 0$ **to** $lena$ **do**

$d_{i,0} \leftarrow i$

end for

for $j \leftarrow 0$ **to** $lenb$ **do**

$d_{0,j} \leftarrow j$

end for

for $j \leftarrow 1$ **to** $lenb$ **do**

for $i \leftarrow 1$ **to** $lena$ **do**

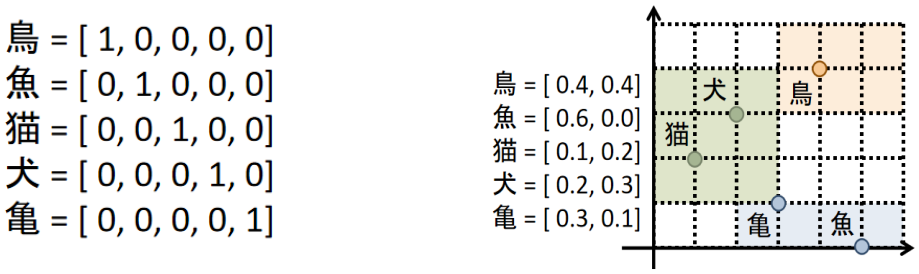
$$c \leftarrow \begin{cases} 0 & \text{if } a_i = b_j \\ 1 & \text{otherwise} \end{cases}$$

$d_{i,j} \leftarrow \min\{d_{i-1,j} + 1, d_{i,j-1} + 1, d_{i-1,j-1} + c\}$

end for

end for

return $d_{lena,lenb}$



a. One-hot 表現　　b. 分散表現

図 3.2: One-hot 表現と分散表現の例

3.3　局所表現と分散表現

コンピュータ上で自然言語を表現する手法として, 最もシンプルなものが局所表現である. その中に, 代表的な手法は One-hot 表現である. One-hot 表現は単語 (漢字も含め) をベクトルの各次元に 1 対 1 対応させる表現方法である. 非常に単純な手法であり, 実装が容易であるという利点がある. しかし, One-hot 表現では語彙数とベクトルの次元数が等しくなるため, 語彙数の増大とともにベクトルの次元数も増大し, ベクトル空間がスパースになってしまう問題がある. また, 各単語がベクトル空間上で等距離に配置されてしまうため, 単語間の意味的な関係性については定義できないことも大きな問題である.

局所表現の問題点を解決するために考案された手法が分散表現である. 分散表現は各概念をベクトルの単一次元ではなく複数次元の実数で表す. 単語の分散表現は類似した文脈で使用される単語は類似した意味をもつという分布仮定を基盤としている. 単語を実数値密ベクトルで表現することにより, 単語間の意味的な関係性をベクトル空間上での類似度として定義できるという大きな利点がある. 代表的な分散表現の生成手法として, Word2Vec [12], fastText [13], Global Vectors (GloVe) [14] などが利用されている. 図 3.2 に One-hot 表現と分散表現の例を示す.

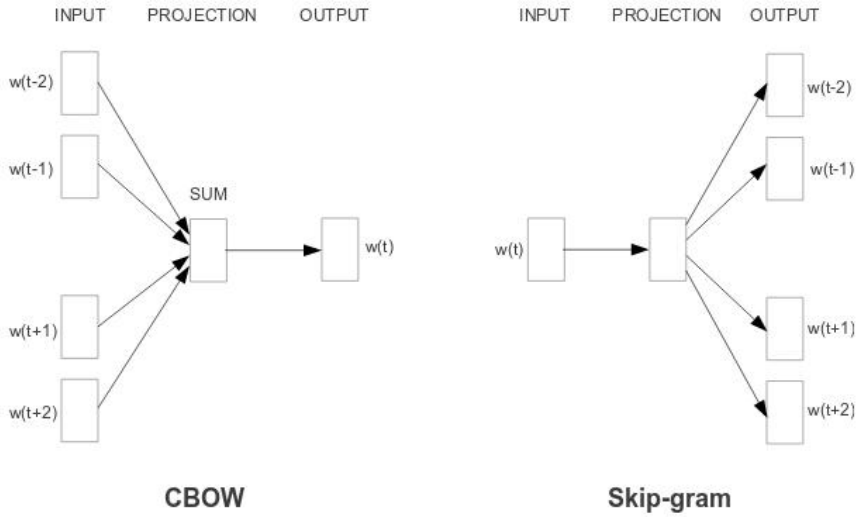


図 3.3: CBoW および skip-gram のモデル構造 (文献 [1] Fig 1 より)

3.4 Word2Vec

Word2Vec[12] は単語の分散表現を得る手法である. この手法は単語前後の文脈を利用し, 単語をベクトル化することで単語間の意味的な類似度の計算や, 「王様」–「男」+「女」=「女王」のような単語間の意味を考慮した演算などができるようになる. Word2Vec では単語の意味を文中で交換可能かどうかに注目して取得していく. 単語同士が文中で交換できる場合にはそれらは似た意味を持っている, ということである. Word2Vec では Continuous Bag-of-Words (CBoW) モデルと skip-gram モデルが提案しており, どちらも周囲のコンテキストから現在の単語を予測する. CBoW は単語の順序を問わないが, skip-gram は近くの単語の重みを大きくさせる違いがある. 2 層のニューラルネットワークで学習, その中間層の重みを分散表現として取得する. 図 3.3 に CBoW および skip-gram のモデル構造を示す

3.5 Chinese Word Vectors

Chinese Word Vectors [15] とは Li らが 2018 年に公開した中国語プロジェクトである. このプロジェクトは ngram2vec [15] の手法を利用し, かつ違う分野

の中国語コーパスで 100 種類以上の単語 (漢字も含め) 分散表現を生成し, 提供する. 生成した分散表現はテキストの形式で、GitHug に保存する [1] .

[1] https://github.com/Embedding/Chinese-Word-Vectors

3.6 Reccurent Neural Networks

Recurrent Neural Network (RNN) [2] は系列データを扱うニューラルネットワークである. 閉路構造により, RNN はある時点での情報を一時的に記憶し, 次の状態に系列データを渡すことが可能となるため, 音声やビデオ, 小説などの文章といったもののデータの取り扱いに適用されている. 理論的に, RNN は過去に入力されたすべての情報の時系列関係が考慮されるはずであるが, 実際, 長期間の系列データを扱う場合に, 誤差逆伝播による勾配が非常に大きくなる (勾配爆発) または非常に小さくなる (勾配消失) 問題があり, 大きな問題となっている. この問題を解決するため, Hochreiter らは RNN の拡張手法である Long short-term memory (LSTM) [9] を提案した.

3.7 Long short-term memory

LSTM[9] は RNN の一種であり, 長期間の時系列性を持つデータの利用に優れたモデルである. LSTM では RNN のノードの代わりに, 入力値や重みを保持することができる LSTM ブロックを用いており, 重み係数を 1 にすることで誤差の消失などを回避し過去の情報を保持すること (Constant Error Carousel: CEC) を可能としている.

図 3.4 に LSTM の内部構造を示す. 過去の状態を記録するメモリセル, 入力, メモリセルに加える値を調整する入力判断ゲート, メモリセルの値の持続を調整する忘却判断ゲート, メモリセルの値の次層への影響を調整する出力判断ゲート, 出力で構成されている. 図 3.4 において, h_t^l は時刻 t の l 番目の層の出力を, C_t は時刻 t のメモリセルを表す.

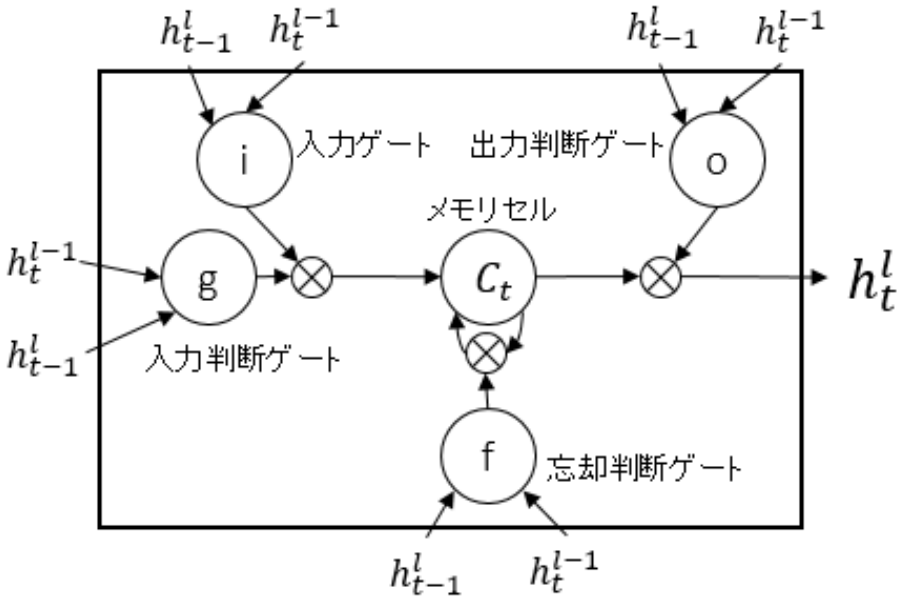


図 3.4: LSTM の内部構造 (文献 [2] Fig 1 より)

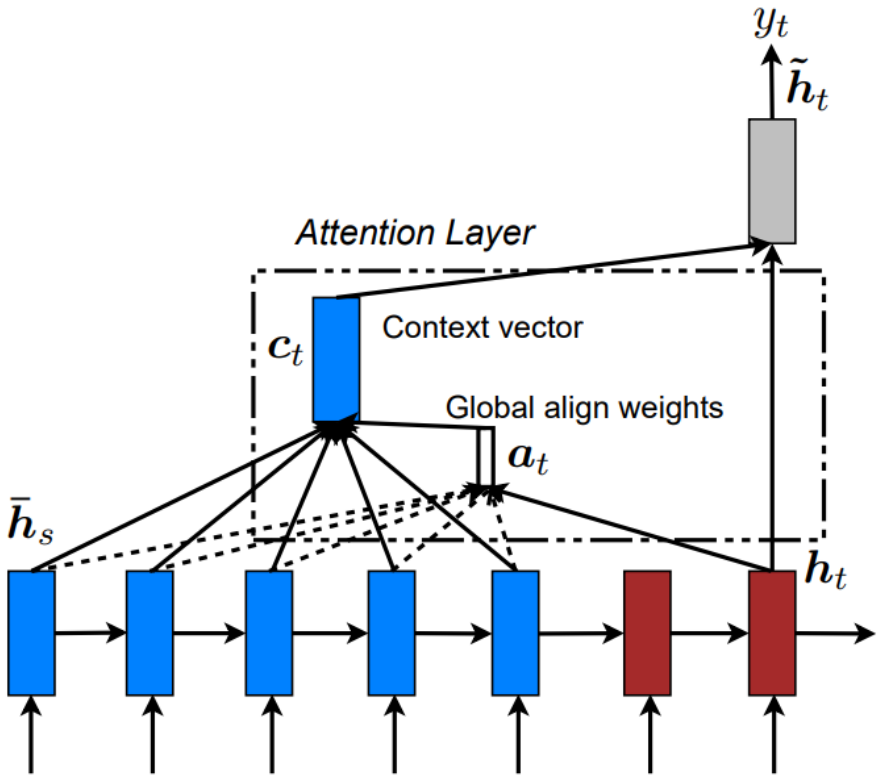


図 3.5: Global Attentional Model のモデル構造 (文献 [3] Fig 2 より)

3.8　Attention

Attention 機構 [16] とは, 要素間の関連の度合いをニューラルネットワークにより学習することにより関連が強い要素の影響を大きく受けるよう工夫がされたモデルである. Luong らの Global Attentional Model [3] により提案された. 図 3.5 に Global Attention Model のモデル構造を示す. 翻訳タスクにおける LSTM において, 単語間の関連の大きさ $a_t(s)$ を現在の隠れ状態 h_t と過去の入力の隠れ状態 $\bar{h}_s$ を用いて, 式 3.1, 3.2 と計算する. これらを過去の隠れ状態 $\bar{h}_s$ と合わせて足すことで, 時刻 t における Context vector である c_t を得る.

$$a_t(s) = align(h_t, \bar{h}_s) \tag{3.1}$$

$$= \frac{exp(score(h_t, \bar{h}_s))}{\sum_{s'} exp(score(h_t, \bar{h}_{s'}))} \tag{3.2}$$

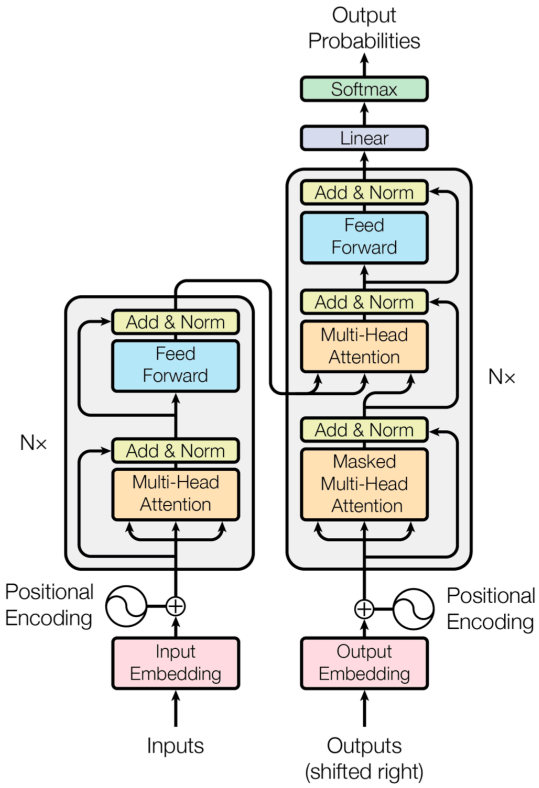


図 3.6: Transformer の概要図 (文献 [4] Fig 1 より)

3.9 Transformer

Transformer [4] は RNN や Convolutional Neural Network (CNN) を用いず, Attention のみを重点的に利用した Encoder-Decoder モデルである. 図 3.6 に Transformer の概要図を示す. RNN は時系列データに対して有効な手法ではあるものの, 逐次的に単語を処理する必要があるため, 並列処理が不可能であり, 学習に時間がかかるが, Transformer は Attention のみを使用することで並列計算を可能としている.

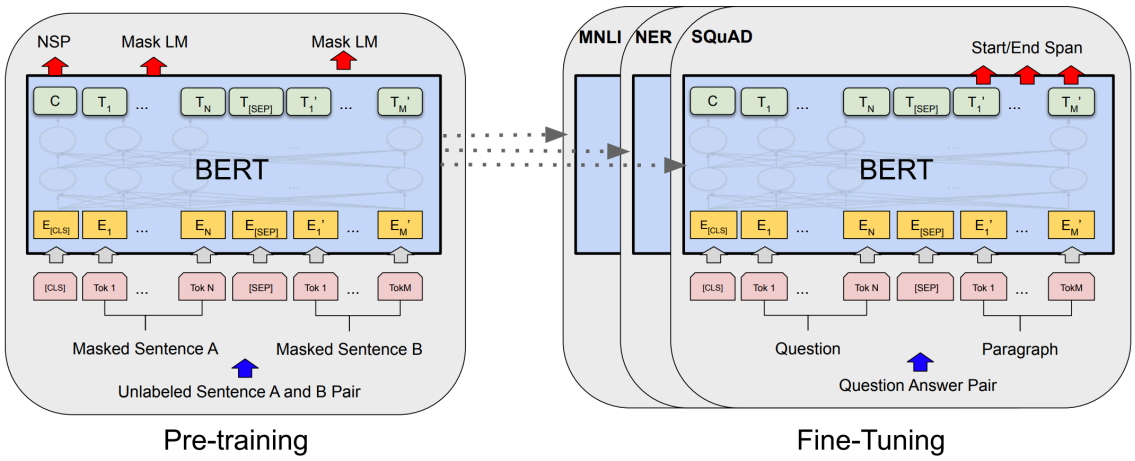


図 3.7: BERT の概要図 (文献 [5] Fig 1 より)

3.10　Bidirectional Encoder Representation from Transformers

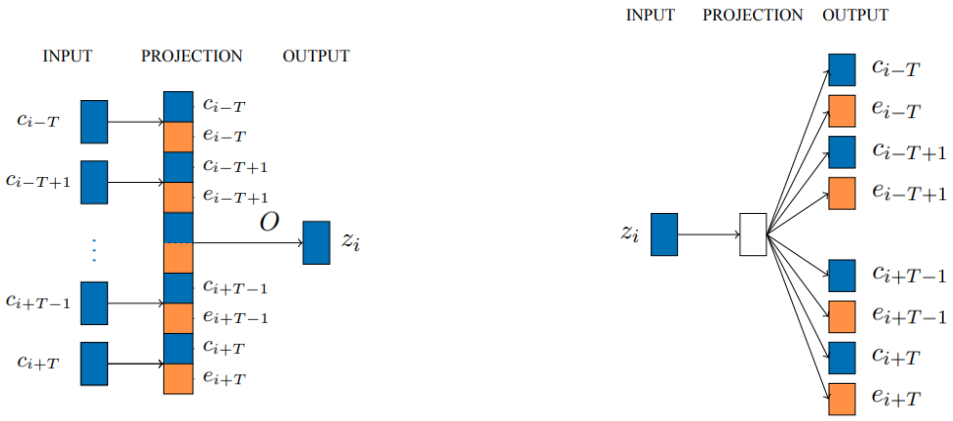
BERT (Bidirectional Encoder Representations from Transformers) [5] とは Transformer による双方向のエンコーダであり, 2018 年 10 月に Google が発表した言語モデルであり, 文章分類, 質問応答, 固有表現抽出等の多様なタスクで公開当時の最高性能を達成するといった大きな成果が報告されている. 図 3.7 に BERT の概要図を示す. BERT は ELMo (Embeddings from Language Models) [17] をもとにしたモデルであり, 双方向 LSTM を Transformer に置き換えている. 事前学習する際, Transformer によって, 全ての層で左右両方の文脈を加味した学習が可能となる.

BERT には事前学習と fine-tuning の 2 つの学習ステップが存在する. 事前学習では, 周囲の単語からある単語を予測する MLM (Masked Language Model) と, 入力された 2 文が連続した文章であるかを予測する NSP (Next Sentence Prediction) によってモデルを学習する. このような事前学習によって得られたモデルを fine-tuning することで, 様々なタスクに対応できる.

さらに, 以前はモデル毎に語彙を 1 から学習させるため, 非常に多くの時間とコストがかかっていたが, BERT ではオープンソースで公開されている文脈を既に学習させた Pre-Trained BERT モデル を使用することにより短時間で学習ができる.

3.11 Optuna

Optuna [18] は，オープンソースのハイパーパラメータ自動最適化フレームワークである．Tree-structured Parzen Estimator というベイズ最適化アルゴリズムを用いて，過去の試行に基づいて有望そうな領域を推定し再度試行する．これを繰り返すことで最適なハイパーパラメータの値を自動的に発見する．使用する際は目的関数を定め，その値がより大きくまたは小さくなるように推定を進める．Optuna の主な特徴として，Define-and-Run スタイルの API，学習曲線を用いた試行の枝刈り，並列分散最適化が挙げられる．



a. charCBOW　　b. charSkipGram

図 4.1: charCBOW と charSkipGram のモデル構造 (文献 [6] Fig 1 より)

4　関連研究

字形で解く灯謎の解答に用いられる研究としては, 漢字の形の情報抽出に関する研究がある. 次に機械学習を用いた漢字の形の情報抽出に関する研究をいくつか紹介する.

4.1　Component-Enhanced Character Embedding

Component-Enhanced Character Embedding [6] は漢字の部首情報を利用した漢字の分散表現を生成する手法である. この手法では, Word2Vec で漢字の分散表現を生成する同時に, 漢字より小さい単位である「部首」の分散表現を追加情報連結する. この手法により, 漢字の部首情報を含まれた charCBOW と charSkipGram が提案された. 図 4.1 に charCBOW と charSkipGram のモデル構造を示す.

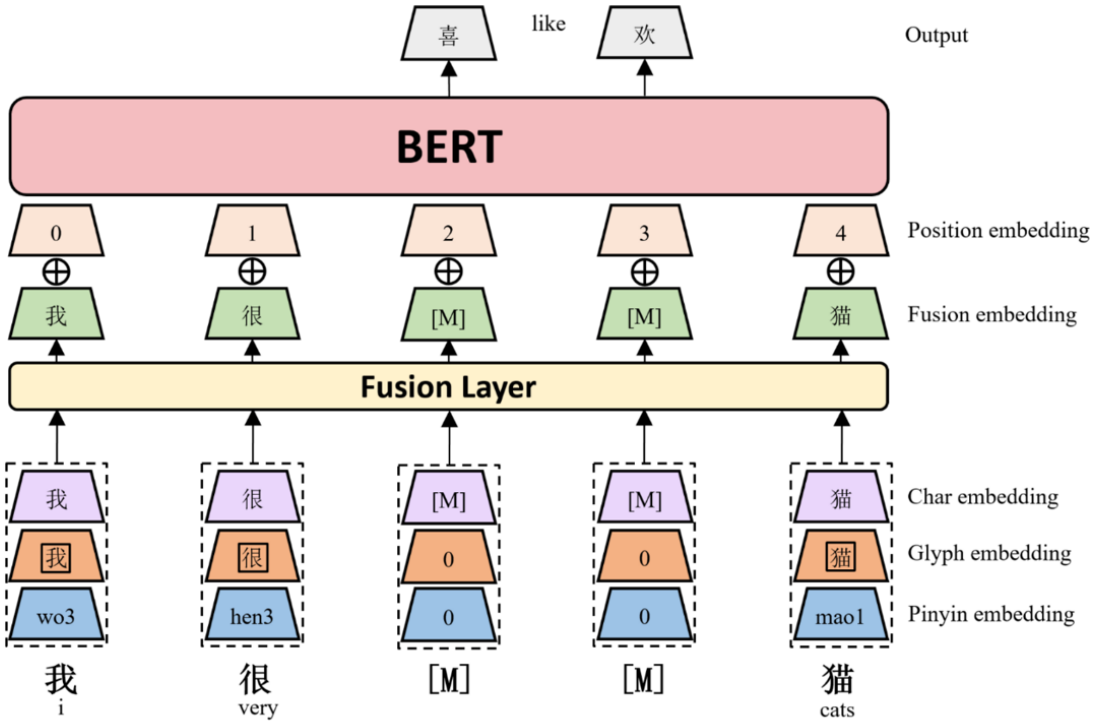


図 4.2: ChineseBERT のモデル概要図 (文献 [7] Figure 1. 参照)

4.2　ChineseBERT

ChineseBERT [7] は, ShannonAI が 2021 に公開した汎用言語モデル BERT の訓練済みモデルである. ChineseBERT は漢字の特徴を考慮し, BERT に既存の Char Embedding の上, 漢字の音声情報を抽出する Pinyin Embedding と画像情報を抽出する Glyph Embedding を導入することで, 漢字の情報をより豊富させる. ChineseBERT の事前学習は, 周囲の単語から単語を予測する Masked Language Model (MLM) のみ利用し, 単語にマスクを付ける Whole Word Masking (WWM) と 漢字にマスクを付ける Char Masking (CM) 手法でモデルを学習する. このような事前学習によってえられたモデルは中国語言語処理タスクに良い表現をえられる. 図 4.2 に ChineseBERT のモデル概要図を示す.

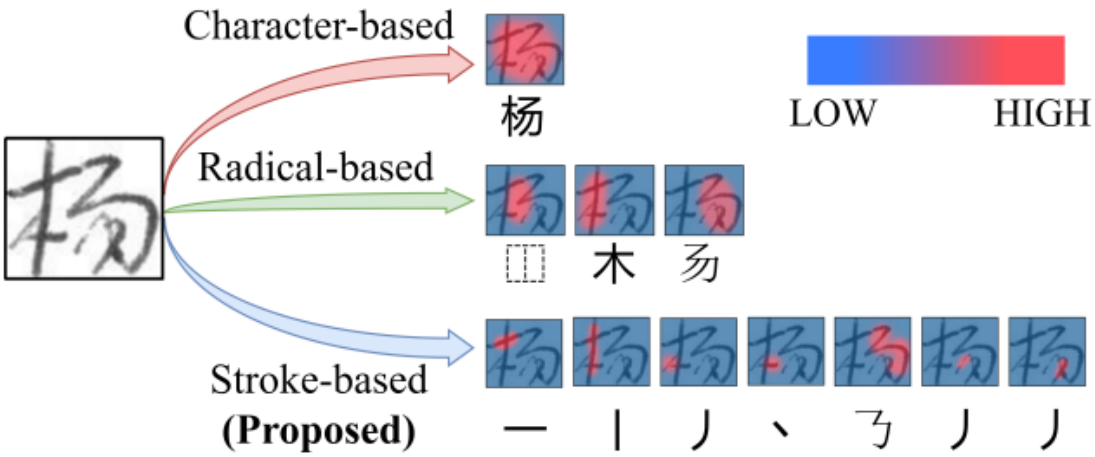


図 4.3: 中国語の分解手法 (文献 [7] Figure 1. 参照)

4.3　Zero-Shot Chinese Character Recognition with Stroke-Level Decomposition

従来の研究は主に漢字を中国語の最小単位として扱うが, そこに起きた Zero-Shot 問題 (訓練データに存在しないデータを処理できない) は避けられない, そこで Cao らは, 木構造に基づいた漢字分解手法 Hierarchical Decomposition Embedding (HDE) [19] を提案した. HDE では, 漢字の構造を表現する漢字構成記述文字 (Ideographic Description Characters) と漢字の成分を表現する radicals で漢字を分解し, 木構造の先行順で並べる漢字の分解手法である. Chen らはそれらの手法の上, 五筆字型入力方法を参考して radicals より小さい単位である漢字の画を基づいた漢字識別モデルを提案した [20]. 図 4.3 に中国語の分解手法を示す.

5　データセット

本章では, 実験用の二値分類灯謎データセットの作成について述べる.

5.1　中華灯謎データベース

中華灯謎データベース[2] は 2002 年に灯謎愛好者が公開した灯謎データベースである. 中国最大な灯謎データベースとして中華灯謎データベースは様々な「灯謎」1,404,526 件と灯謎を作成する「謎材」1,649,602 件を収録している.　図 5.1 に中華灯謎データベースの例を示す.　中華灯謎データベースの灯謎の情報は以下の 9 つの情報に集められており, 表の形式で保存されている.

- 謎面 : 灯謎の問題
- 謎目/謎格 : 答えのヒント/灯謎作成のルール
- 謎底 : 灯謎の答え
- 作者 : 灯謎の作者
- 灯謎備注 : 灯謎の解答に対する説明
- 参与管理 : 記入錯誤などへの指摘
- 発行時間 : 公開時間
- 来源類別 : データオリジナリティ
- 来源説明 : データオリジナリティに関する説明

本研究は中華灯謎データベースのうち, 答えが 1 文字である「字謎」を収集し, 謎面, 謎目, 謎底部分のみ使用する.

[2]http://www.zhgc.com/mk/

谜面	谜目/谜格	谜底	作者	灯谜备注	参与管理	发表时间	来源类别	来源说明
虽有见底信号，一刀割了八成	4字通信名词	信号分子【霹春:信号】	瓢雯吐	【信号二字入底】	指谜	2015/10/25	各地谜事	2015莫秋杯_冬妮娅队
此刻陆放翁，病得可不轻	即物赠/土音	时游疾重	逸翔		指谜	2015-3-12	各地谜事	各地媒体收集
马先生援鲁上头版	乐山美食	乌鱼片	许淫金		指谜	2021年8、9、10、11、12月	各地谜事	乐山灯谜协会月会“与虎谋皮”
“场主积薪其中”	乐山绿心地名·卷帘	火柴地	许淫金	【注：场，平坦的空地。蓁，柴火。依格读着地柴火。面取《聊斋志异·狼三则》句】	指谜	2021年8、9、10、11、12月	各地谜事	乐山灯谜协会月会“与虎谋皮”
“揣着明白装糊涂”	乐山美食	冒菜	许淫金		指谜	2021年8、9、10、11、12月	各地谜事	乐山灯谜协会月会“与虎谋皮”
藏北崛起肩上担	乐山绿心地名	芦山	许淫金		指谜	2021年8、9、10、11、12月	各地谜事	乐山灯谜协会月会“与虎谋皮”
“不才明主弃”	乐山土特产	甩菜	许淫金		指谜	2021年8、9、10、11、12月	各地谜事	乐山灯谜协会月会“与虎谋皮”
看望外祖父到赵州	乐山地名	张公桥	许淫金		指谜	2021年8、9、10、11、12月	各地谜事	乐山灯谜协会月会“与虎谋皮”
半笼老鸡旁，鸟飞沐春光。	乐山地名	竹溪	许淫金		指谜	2021年8、9、10、11、12月	各地谜事	乐山灯谜协会月会“与虎谋皮”
甘露初酒岷江上	乐山美食	甜水面	许淫金		指谜	2021年8、9、10、11、12月	各地谜事	乐山灯谜协会月会“与虎谋皮”

首页　前页　第13072/13072页[共130714条，每页10条]　后页　尾页

図 5.1: 中華灯謎データベース

5.2　二値分類灯謎データセットの作成

本研究は灯謎のうち,「字謎」と呼ばれる答えが 1 文字である灯謎 79,725 件を収集し, 字形で解く字謎 72,937 を実験データ (以下は字謎データと称する) として使用した.

まずは収集した字謎データをもとに, 謎底 (以下は正解と称する) の低頻度語 (正解として使われる頻度が 4 以下) を取り除く, 謎面, 謎目と正解のペアを正例として True ラベルを付け, 漢字ボキャブラリを生成した.

次に漢字ボキャブラリを利用して「漢字を漢字の画に分解した辞書データ」を生成し, これらの辞書データと Levenshtein 距離を利用して Levenshtein が大きい「簡単な不正解」と Levenshtein が小さい「困難な不正解」を生成した. 図 5.2 に不正解生成の例を示す.

図 5.2: 不正解の生成

表 5.1: データセットの情報

簡単データセット	訓練データ	テストデータ
データ総数	106,464	26,616
正例	53,277	13,263
負例	53,187	13,353
困難データセット	訓練データ	テストデータ
データ総数	106,464	26,616
正例	53,277	13,263
負例	53,187	13,353

最後に正解と同じ謎面, 同じ謎目に不正解の漢字を付与したデータを負例のデータとして False ラベルを付ける. このようにして正解データと不正解データのペアにより「簡単データセット」と「困難データセット」を作成していき, データセットを 8 対 2 の比率で訓練データとテストに分けて実験する. データ不均衡問題を避けるため, 正解と不正解のデータ数を揃えた. 表 5.1 にデータセットの情報を示す.

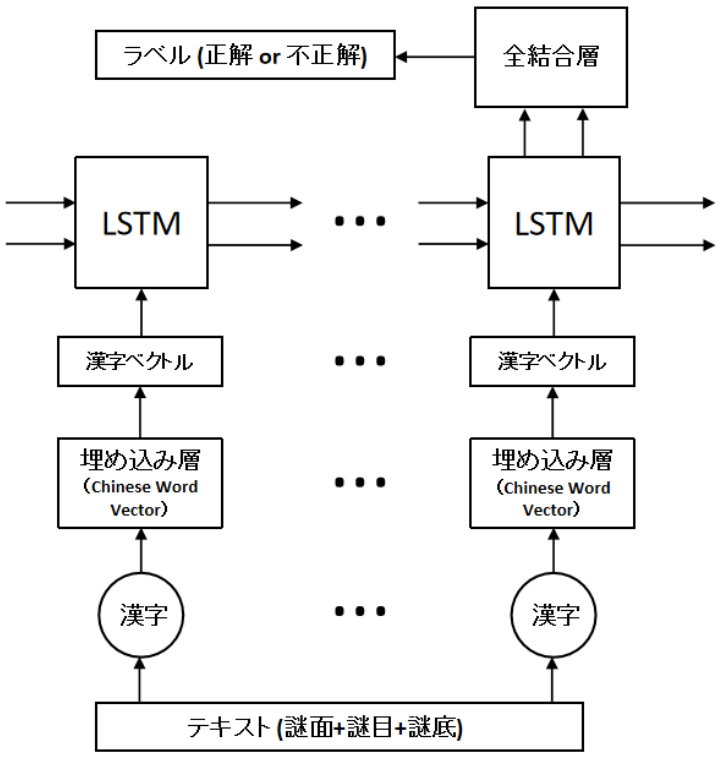


図 6.1: Chinese Word Vector を用いた LSTM のモデル構造

6　提案手法

6.1　灯謎問題の正解推定分類器の構築

本研究は事前学習済みの Chinese Word Vector と LSTM を用いて漢字の画情報を加えた灯謎問題の正解推定モデルを提案する. 二値分類灯謎データセットにある中国語簡体字と繁体字を当時に扱うため, 本研究は簡体字と繁体字両方含む Zhihu_QA コーパスで訓練した Chinese Word Vector を利用した. 図 6.1 に Chinese Word Vector を用いた LSTM のモデル構造を示す. 入力の灯謎を Chinese Word Vector により漢字に分割し, 漢字の分散表現のリストに変換する. 続いて, Encoder である LSTM に順次に入力し, 前文との関係を考慮した分散表現に変換する. その後謎面, 謎目, 謎底全文の分散表現を生成して線形層に入れて灯底は正解かどうかを推定する.

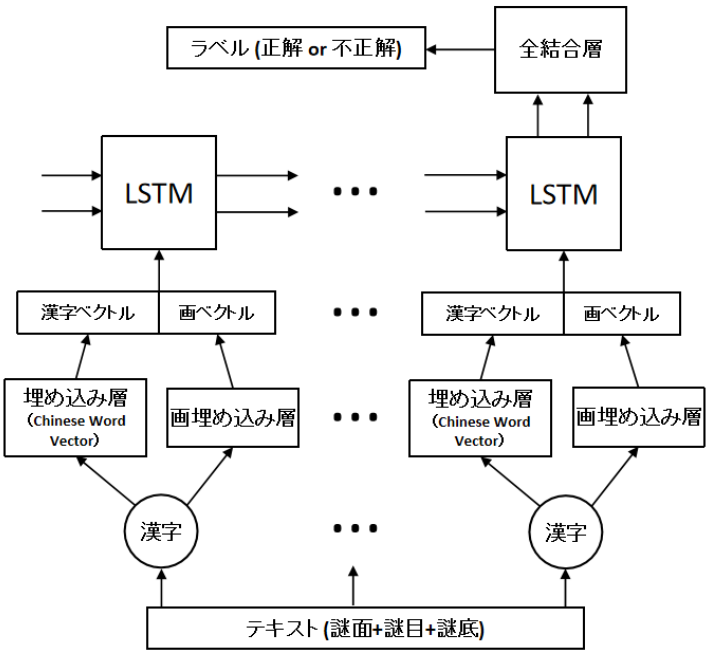


図 6.2: 提案手法によるモデル構造

謎目は灯謎の正解推定に重要な役割を果たす. 例えば, 謎目「打一字」は灯謎の謎底が 1 文字であることを示す. 謎目「打一五筆字」はさらに謎底が 5 画の字であることを示す. 故に本研究では, Chinese Word Vector に漢字の画情報を組み込むことで灯謎問題の正解推定モデルを改善する. 図 6.2 に提案手法によるモデル構造を示す. 具体的に, まず入力漢字と五筆字型入力方法を用いて, 漢字の書き順により漢字の画の五筆コートを生成する. 続いて, 生成したコートを五次元ベクトルに変換し, 埋め込み層により画のベクトルを生成する. 図 6.3 に漢字の画のベクトル生成例を示す.

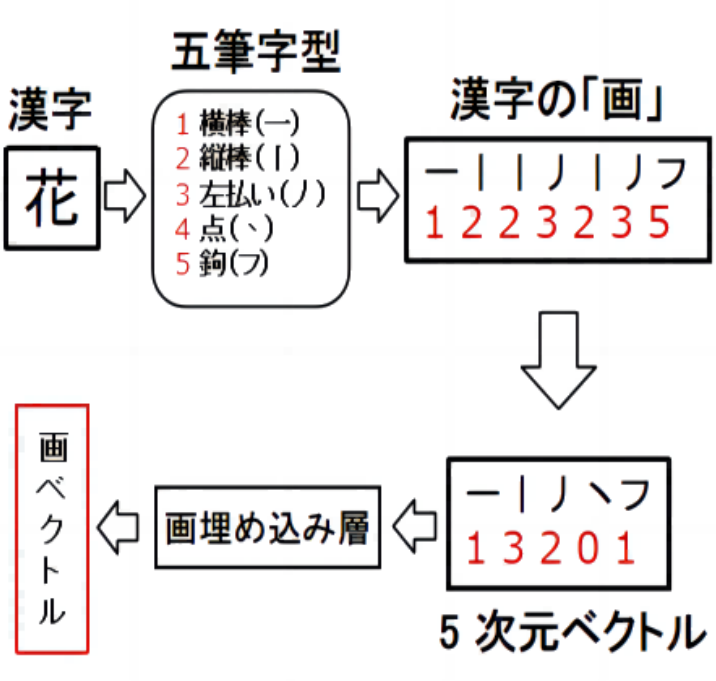


図 6.3: 画ベクトルの生成

7　数値実験

本章では, 本研究で取り組んだ数値実験について述べる.

7.1　実験概要

提案手法で示した正解推定分類器を用いて, 簡単データセットおよび困難データセットにおける灯謎問題の答えが正解か否かを二値分類で予測する. モデルの入力として, 謎面, 面目, 謎底の間に「;」分割符号として入れ, 灯謎の正例のテキストを生成した. 次に Chinese Word Vector によりテキストを漢字に分け, 漢字ベクトルを生成した. その同時に, 各漢字の画により 5 次元ベクトルを生成し, 線形層を利用して画ベクトルを生成した. 続いて漢字ベクトルと画ベクトルを結合して LSTM に入力した. 最後に予測値と真値の誤差を計算して学習を進めた. 以上は一回実験の流れである.

7.2　数値設定

実験のハイパーパラメータは Optuna によって最適化する. そして提案手法を従来の分類手法である LSTM (Word2Vec) および BERT[5] と比較する.

LSTM モデルの実験用パラメータ設定として, 分散表現の次元数は 300, 画の分散表現の次元数は 30, 隠れ層の次元数は 256, バッチサイズは 128, Dropout は 0.5, 最適化手法は Adam, 学習率は 0.00003, Epoch 数は 400 とした.

表 7.1 に LSTM モデルの実験用パラメータを示すその一方, BERT モデルは学習済みの "bert-base-chinese" を利用し. 表 7.2 に BERT モデルの実験用パラメータを示す

表 7.1: LSTM の実験条件

分散表現の次元数	300
画の次元数	30
隠れ層の次元数	256
batch size	128
Dropout	0.5
誤差関数	Cross-entropy Loss
最適化手法	Adam
学習率	0.00003
Epoch	400

表 7.2: BERT の実験条件

隠れ層の次元数	768
batch size	128
Dropout	0.5
誤差関数	Cross-entropy Loss
最適化手法	AdamW
学習率	0.00005
Epoch	400

表 7.3: 各手法によるテストデータの予測結果

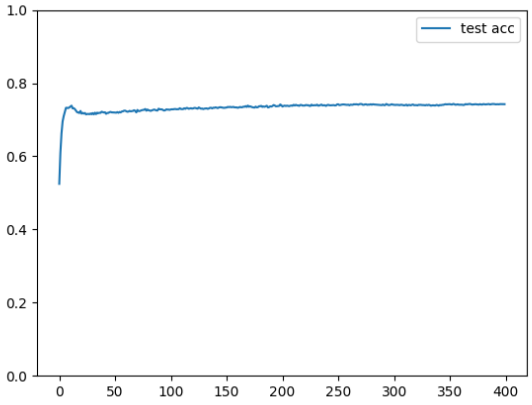
手法	Accuracy	Precision	Recall	F1 値
BERT	73.26%	74.08%	72.40%	73.22%
LSTM+Word2Vec	81.68%	79.73%	84.78%	82.18%
LSTM+Word2Vec+Stroke	**85.76**%	**85.55**%	**85.94**%	**85.74**%

7.3　実験結果

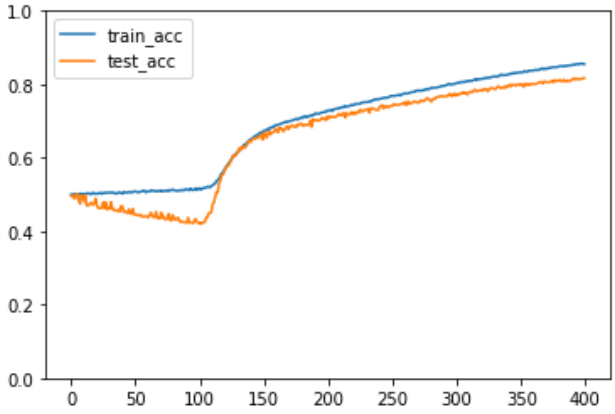
7.3.1　簡単データセットによる実験結果

実験結果は精度を示す Accuracy, Precision, Recall, F1 値で評価した. 簡単データセットによる実験結果として, 提案手法の Accuracy と F1 値は各自 85.76% と 85.74% となり, 従来手法の BERT モデルの 73.26% と 73.22% をそれぞれ 12.50% と 12.52% 上回り, 従来手法の LSTM+Word2Vec モデルの 81.68% と 82.18% をそれぞれ 4.08% と 3.56% 上回った. 表 7.3 に各手法による灯謎の正解の予測結果を示す. 太字である “LSTM+Word2Vec+Stroke” 項目は提案手法である.

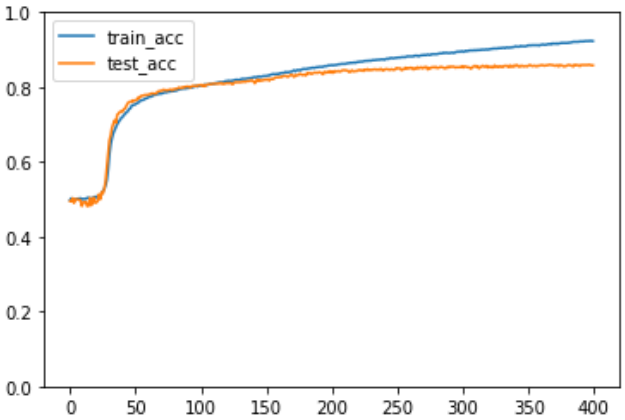
図 7.1 a に BERT の精度の推移を示す. 図 7.1 b に LSTM+Word2Vec の精度の推移を示す. 図 7.1 c に提案手法の精度の推移を示す. 共に横軸が Epoch 数であり, 縦軸が精度である. 表 7.4 に BERT の混同行列を示す. 表 7.5 に LSTM+Word2Vec の混同行列を示す. 表 7.6 に提案手法の混同行列を示す.



a. BERT の精度の推移



b. LSTM+Word2Vec の精度の推移



c. 提案手法の精度の推移

図 7.1: 各手法の精度の推移

表 7.4: BERT の混同行列

		予測値		
		正解	不正解	合計
真値	正解	9,602	3,661	13,263
	不正解	3,438	9,915	13,353
	合計	13,040	13,576	26,616

表 7.5: LSTM+Word2Vec の混同行列

		予測値		
		正解	不正解	合計
真値	正解	11,245	2,018	13,263
	不正解	2,859	10,494	13,353
	合計	14,104	12,512	26,616

表 7.6: 提案手法の混同行列

		予測値		
		正解	不正解	合計
真値	正解	11,398	1,865	13,263
	不正解	1,925	11,428	13,353
	合計	13,323	13,293	26,616

表 7.7: 各手法によるテストデータの予測結果

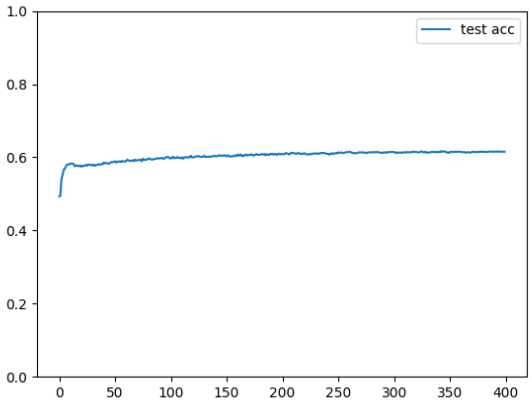
手法	Accuracy	Precision	Recall	F1 値
BERT	61.53%	61.13%	61.53%	61.33%
LSTM+Word2Vec	76.88%	75.80%	78.70%	77.22%
LSTM+Word2Vec+Stroke	**79.61**%	**78.76**%	**80.89**%	**79.81**%

7.3.2　困難データセットによる実験結果

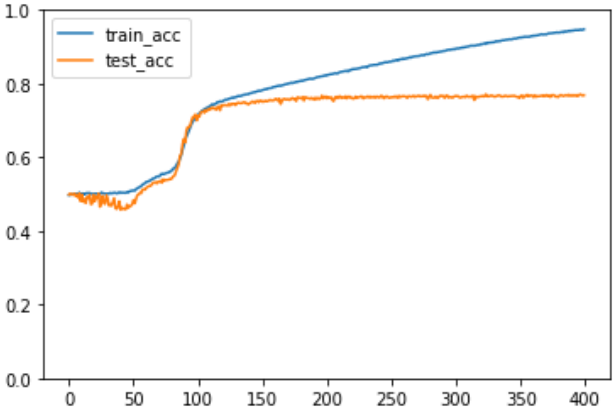
困難データセットによる実験結果として, 提案手法の Accuracy と F1 値は各自 79.61% と 79.81% となり, 従来手法の BERT モデルの 61.53% と 61.33% をそれぞれ 18.08% と 18.48% 上回り, 従来手法の LSTM+Word2Vec モデルの 76.88% と 77.22% をそれぞれ 2.73% と 2.59% 上回った. 表 7.7 に各手法による灯謎の正解の予測結果を示す. 太字である “LSTM+Word2Vec+Stroke” 項目は提案手法である.

図 7.2 a に BERT の精度の推移を示す. 図 7.2 b に LSTM+Word2Vec の精度の推移を示す. 図 7.2 c に提案手法の精度の推移を示す. 共に横軸が Epoch 数であり, 縦軸が精度である. 表 7.8 に BERT の混同行列を示す. 表 7.9 に LSTM+Word2Vec の混同行列を示す. 表 7.10 に提案手法の混同行列を示す.

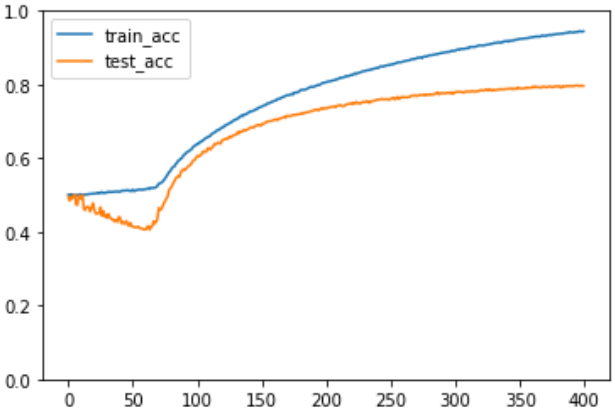
結論として, 簡単データセットと困難データセットどちらでも, 提案手法を用いた場合の Accuracy, Precision, Recall, F1 値がそれぞれ最大となる. 漢字の画情報の特徴量は灯謎正解を推定するのに有効であることが示せた. そして, どんな手法でも困難データセットに対する精度は簡単データセットに下げる. そのため, 人工知能が灯謎の難易度に対する感知は人間と同じであることは確定できる. 原因として, 灯謎のヒントである謎目は, 答えである謎底の特徴を記載するものが多いであることを考えられる. 例えば, 謎目「打一五筆字」は「五画の 1 文字」を指し, その場合は漢字の画数から正解と不正解を推定できる. しかし, 困難データセットは, 正解と不正解の画数や, 形が似ているため, このような推定手法は利用できない. そのため精度は下げられた.



a. BERT の精度の推移



b. LSTM+Word2Vec の精度の推移



c. 提案手法の精度の推移

図 7.2: 各手法の精度の推移

表 7.8: BERT の混同行列

		予測値		
		正解	不正解	合計
真値	正解	8161	5102	13,263
	不正解	5155	8198	13,353
	合計	13,316	13,300	26,616

表 7.9: LSTM+Word2Vec の混同行列

		予測値		
		正解	不正解	合計
真値	正解	10,438	2,825	13,263
	不正解	3,332	10,021	13,353
	合計	13,770	12,846	26,616

表 7.10: 提案手法の混同行列

		予測値		
		正解	不正解	合計
真値	正解	10,729	2,534	13,263
	不正解	2,893	10,460	13,353
	合計	13,622	12,994	26,616

8　まとめと今後の課題

本研究では中国の伝統的クイズ「灯謎」のデータセットを新たに構築し, 漢字の画ベクトルを用いた灯謎問題の正解推定の手法を提案した. 漢字の画ベクトルを導入することで, 灯謎問題の正解推定モデルは漢字の構造情報を利用でき, 従来手法を上回る精度を達成した.

今後の課題として, 答えが 1 文字の「字謎」に加えて, 全部の灯謎問題に利用できる手法の考案,「画」より詳細な漢字の形の情報を導入した灯謎問題の正解推定の精度向上が挙げられる.

9　謝辞

本研究に対し査読に加え, 御助言をいただいた藤本典幸教授と黄瀬浩一教授に厚くお礼申し上げます. 本研究を進めるにあたり御指導, 御鞭撻を賜りました森直樹教授に深く感謝申し上げます. 研究のアイデアから, 方針, 本論文の作成に至り日頃から御指導頂きました岡田真助教に深く感謝いたします. まだ, 中華灯謎データベースの利用許可を下さった管理者呉紅様に感謝申し上げます. 最後に, 研究に関して建設的な意見を下さった創発ソフトウェア研究室の皆様に感謝いたします.

2022 年 2 月 24 日

参考文献

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space, 2013.

[2] W. Zaremba, I. Sutskever, and O. Vinyals. Recurrent neural network regularization, 2014.

[3] M.-T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation, 2015.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2017.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *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)*, pp. 4171–4186, June 2019.

[6] Y. Li, W. Li, F. Sun, and S. Li. Component-enhanced Chinese character embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 829–834, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics.

[7] Z. Sun, X. Li, X. Sun, Y. Meng, X. Ao, Q. He, F. Wu, and J. Li. ChineseBERT: Chinese pretraining enhanced by glyph and Pinyin information. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 2065–2075, Online, Aug. 2021. Association for Computational Linguistics.

[8] 呉修喆. 漢字文化における文字遊戯の近代的形成: 燈謎を例にして. PhD thesis, University of Tokyo (東京大学), 2017.

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[10] 五筆字型計算机漢字輸入技術. 河南科学技術出版社, 1985.

[11] V. I. Levenshtein, et al. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, Vol. 10, pp. 707–710. Soviet Union, 1966.

[12] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality, 2013.

[13] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[14] J. Pennington, R. Socher, and C. Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, Oct. 2014. Association for Computational Linguistics.

[15] S. Li, Z. Zhao, R. Hu, W. Li, T. Liu, and X. Du. Analogical reasoning on Chinese morphological and semantic relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 138–143, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[16] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate, 2014.

[17] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long*

Papers), pp. 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[18] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A next-generation hyperparameter optimization framework, 2019.

[19] Z. Cao, J. Lu, S. Cui, and C. Zhang. Zero-shot handwritten chinese character recognition with hierarchical decomposition embedding. *Pattern Recognit.*, 107:107488, 2020.

[20] J. Chen, B. Li, and X. Xue. Zero-shot chinese character recognition with stroke-level decomposition. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 615–621, 2021.