

## 深層学習による灯謎問題解決システムの構築

### 1 はじめに

質問応答は文書をもとに入力された質問に対して正しく応答することを目的とするタスクである。今まで質問応答に対する研究は読解問題と呼ばれ、主に問題と答えの情報を含む長文を対象としてきた。その一方で、問題の中に隠された情報で問題を解ける「クイズ問題」と呼ばれるタスクも存在する。本研究ではこのクイズ問題にとりくむ。具体的には中国の伝統的クイズ問題「灯謎 (トウメイ)」を深層学習の手法を用いて解く方法について提案する。

### 2 灯謎 (トウメイ)

灯謎は、中国の伝統的クイズ問題である。質問者は問題を詩や熟語の形で出し、回答者はそれに回答する。答えは常に字または単語になる。灯謎は質問に答えるための問題文以外の文書や知識など必要がないものが多く、質問の文中から答えの情報を得ることが容易である。つまり、質問を理解すれば回答できると言える。灯謎を解くためには、問題に隠された情報をもとに、問われている内容を理解して抽出しなければならないので、灯謎の研究は一種の情報抽出として考えることもできる。

灯謎のパターンは主に謎とヒントと答えで構成される。謎は詩や熟語や普通の話言葉で記述された文である。ヒントは答えの形を説明する文である。ヒントは 1 つ以上与えられ、答えは字か単語である、問題に隠された字の構成、発音、意味などの情報から解くことができる。図 1 に灯謎の一つの例を示す。

本研究では灯謎問題のうち、「字謎」と呼ばれる、答えが一つの漢字のみ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、単純に大量の問題の文の情報のみ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。そこで本研究では漢字の形の情報に着目し、漢字と SUB 漢字成分を利用した Sequence to Sequence (seq2seq) モデルで灯謎問題を解くシステムを構築した。

問題	ヒント	答え
一百減一	(打一字)	白
百マイナースーは何？ 答えは一文字になる		

図 1: 灯謎の例

### 3 要素技術

#### 3.1 Sequence to Sequence

Sequence to Sequence (seq2seq) [1] は 2014 年に Google が発表した言語モデルである。従来の Deep Neural Network (DNN) が扱いにくい時系列データ問題を解決するため、seq2seq は Encoder-Decoder という形式のモデル構造を導入した。Encoder は入力する時系列データをベクトルに圧縮し、そのベクトルを Decoder に渡し出力系列を生成される。このモデルには機械翻訳タスクによく使われている。

#### 3.2 Gated Recurrent Unit

Recurrent Neural Network (RNN) [2] は再帰的構造を持つニューラルネットワークである。通常のニューラルネットワークでは、レイヤの出力は次のレイヤの入力としてのみ利用されるが、RNN では同じレイヤーに対して現時刻の時系列データだけでなく、前時刻の出力も合わせて入力する。

誤差逆伝播法による RNN の訓練は、逆伝播される勾配の消失 (勾配がゼロに収束)、あるいは爆発 (勾配が無限に発散) する問題がある。この問題を解決するため、Sepp らは Long short-term memory (LSTM) [3] を提唱した。LSTM のアーキテクチャは Memory Cell と 3 つの Gate (Input Gate, Output Gate, Forget Gate) から構成される。LSTM は勾配をそのまま使用することが可能であるので、勾配消失と勾配爆発の問題を解決できる。

Gated Recurrent Unit (GRU) [4] は、LSTM の Input Gate と Forget Gate を統合して隠れ状態のみ利用す

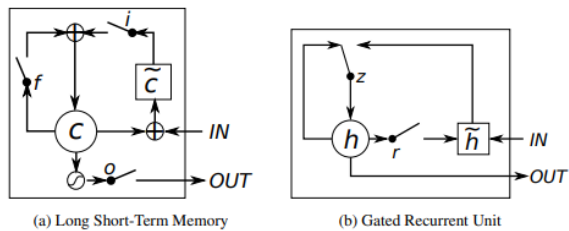


図 2: LSTM と GRU の構造 (文献 [4] より引用)

Unmodified: 彼は路を取った。  
 Shallow: イ皮彼は貝各路を耳又取った。  
 Deep: イイ皮彼は目貝夕口各路を耳又取った。

図 3: 漢字の分け方 (文献 [5] より引用)

るニューラルネットワークである。図 2 に LSTM と GRU の構造を示す。

## 4 データセット

### 4.1 漢字の分け方

漢字の分け方 [5] について、主に大まかに分ける Shallow Mode と細かいに分ける Deep Mode 二つの分け方がある。

図 3 に漢字の分け方を示す。

### 4.2 IDS データセット

IDS (Ideographic Description Sequence)[6] とは、中国語、日本語、韓国語の漢字データを ‘unicode’, ‘漢字’, ‘サブ漢字’ の形で集まったデータセットである。

本研究の対象となる灯謎は中国語で作られたため、IDS の中国語データを利用した。しかし、IDS データセットには漢字の成分不足 (漢字「爽」の「メ」など) という問題がある、そこで本研究で使ったデータ 9285 件の中に漢字の成分が不足であるデータ 782 件に対して、漢字の画の書き順で補足した。そして今回で使った漢字の‘画’データを手動で作成した。

表 1 に本研究に使われている漢字数と手動で SUB 漢字成分を補足した数を示す。

表 1: 使用した漢字数と補足したデータ数

使用した漢字数	補足した漢字数	画を作成した漢字数
9285	782	9285

表 2: 問題の種類

問題の総数	文中の字の情報のみで解ける問題	文中の字の情報のみで解けない問題
79725	72937	6788

## 4.3 中華灯謎ベース

中華灯謎ベース [7] は、中国各地の灯謎ファン達が集めた灯謎問題 1,362,911 件を収録したデータセットである。

本研究では灯謎のヒントの文を使わないため、答えが一文字である問題 79,725 件のみ利用し、研究用の灯謎のデータセットを構築した。

研究用灯謎データセットについて、文中の字の情報のみで解ける問題 72,937 件のみを扱った。

表 2 に問題の種類を示す。

## 5 提案手法

分類モデルで灯謎問題を解くと、答えである漢字の数が 8648 個であるため、モデルの正解率はとても低い。そこでまず出力を漢字の「画」とする seq2seq の構築からはじめる。

Cao らの研究 [8] により、漢字の「画」は「横」、「縦」、「左はらい」、「点」、「折れ」5 種類に分類できる。この手法と TorchText ライブラリ [9] の「unk」、「pad」、「sos」、「eos」トークンを利用すると、モデルの出力は 8648 次元から 9 次元に削減できる。

### 5.1 モデル構造

図 4 に実験に用いた seq2seq モデルの構造を示す。

本研究のモデルでは、まず問題を漢字に分け、順次で Embedding 層で分散表現に変換し、そして Encoder に入力して問題の情報を含まれた Vector を生成した。

続いて、問題の情報を含まれた Vector と「開始」を表示した「sos」トークンを Decoder に入力し、答えは SUB 漢字を書き順で出力した。

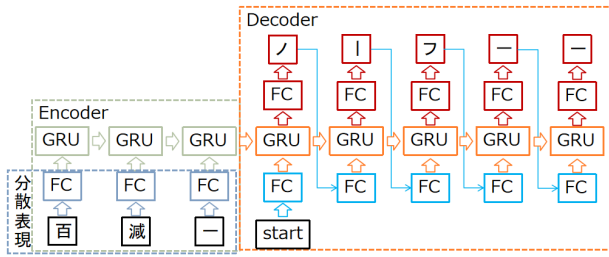


図 4: モデルの構造

表 3: 資料のデータ数

Train Data	Valid Data	Test Data
58350	7293	7294

## 6 実験

### 6.1 データ処理

本実験では問題文中の字の情報のみで解ける問題で作成した研究用の灯謎データ 72,937 件を利用した. その中からランダムに Train Data 58,350 件, Valid Data 7,293 件, Test Data 7,294 件を抽出し, 実験に使用した.

表 3 に資料のデータ数を示す.

### 6.2 実験内容

本研究は漢字の成分を利用する可能性を確認するため, 入力する問題の分かち書き方法により, 「Word」, 「Word + Char」, 「Word + Char + Shallow」, 「Word + Char + Deep」, 「Word + Char + Shallow + Deep」, 「Char」, 「Char + Shallow」, 「Char + Deep」, 「Char + Shallow + Deep」 九つの状況で, 各自 GRU を基づいた Seq2Seq モデルで対照実験をした.

表 4 に各実験の入力次元数を示す.

表 5 にその他のパラメータを示す.

### 6.3 実験結果

100 Epoch を経た実験結果として, 各入力条件を変えた Train Loss を確認した. その中で, 「Word + Char + Shallow」条件の Train Loss は 0.152 に収束し, 結果は最もよいことを確認した. Train Data に対して, Valid Data の中に最もよい「Char」条件の Valid Loss も 2.318 に収束した. 表 6 に実験結果を示す.

図 5, 図 6 に各誤差の変化曲線を示す.

表 4: 各実験入力次元数

実験	入力次元数
Word	51783
Word + Char	53507
Word + Char + Shallow	53897
Word + Char + Deep	53701
Word + Char + Shallow + Deep	58350
Char	4561
Char + Shallow	4951
Char + Deep	4755
Char + Shallow + Deep	4971

表 5: 実験用パラメータ (通用)

パラメータ	数値
出力サイズ	9
分散表現の次元数	256
隠れ層の次元数	512
バッチサイズ	128
Dropout	0.5
最適化手法	Adam
学習率	0.001
損失関数	Cross-Entropy

実験の結果は Test Data による「Precision」, 「Recall」, 「F1」の値で評価した. その結果, 実験全体の正解率は F1 値が約 0.6 にとどまり, 「Word + Char + Shallow + Deep」は最もよいことが確認できる. 表 7 に Test Data の結果を示す.

## 7 まとめと今後の課題

本研究は, GRU による Seq2Seq を用いて, 灯謎問題を解くモデルを構築し, 各入力条件を変えた実験で正解率を確認した. 結果として, 実験全体の正解率は F1 値が約 0.6 にとどまった. その中で, 「Word + Char + Shallow + Deep」の精度は比較的高いことを確認した.

今後の課題として, Attention Mechanism を利用した新しいモデルと IDS 漢字構造情報の導入でモデルの精度向上を目指す.

表 6: Train Loss と Valid Loss

実験	Train Loss	Valid Loss
Word	0.158	3.905
Word + Char	0.193	3.725
Word + Char + Shallow	0.152	3.748
Word + Char + Deep	0.228	3.535
Word + Char + Shallow + Deep	0.219	3.528
Char	0.688	2.318
Char + Shallow	0.460	2.623
Char + Deep	0.455	2.653
Char + Shallow + Deep	0.489	2.611

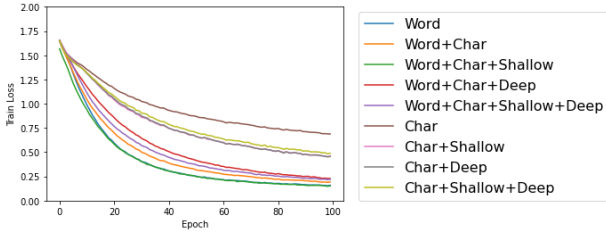


図 5: Train Loss

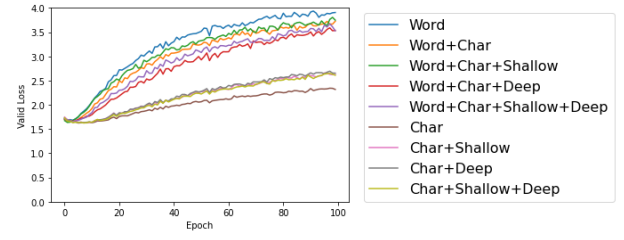


図 6: Valid Loss

表 7: Test Data 実験結果

実験	Precision	Recall	F1
Word	0.541	0.713	0.592
Word + Char	0.506	0.824	0.595
Word + Char + Shallow	0.553	0.686	0.589
Word + Char + Deep	0.597	0.676	0.610
Word + Char + Shallow + Deep	0.609	0.724	0.636
Char	0.576	0.771	0.631
Char + Shallow	0.625	0.686	0.625
Char + Deep	0.633	0.665	0.620
Char + Shallow + Deep	0.633	0.672	0.624

## 参考文献

- [1] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks, 2014.
- [2] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, Vol. 404, p. 132306, Mar 2020.
- [3] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.
- [4] Junyoung Chung, Çağlar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, Vol. abs/1412.3555, , 2014.
- [5] Viet Nguyen, Julian Brooke, and Timothy Baldwin. Sub-character neural language modelling in Japanese. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pp. 148–153, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.
- [6] <https://github.com/cjkvi/cjkvi-ids>.
- [7] <http://www.zhgc.com/mk/>.
- [8] Shaosheng Cao, Wei Lu, Jun Zhou, and Xiaolong Li. cw2vec: Learning chinese word embeddings with stroke n-gram information. *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32, No. 1, Apr. 2018.
- [9] Michael Petrochuk. Pytorch-nlp: Rapid prototyping with pytorch natural language processing (nlp) tools. <https://github.com/PetrochukM/PyTorch-NLP>, 2018.