

## 日本語クロスワードを解く：性能向上の検討

## Solving Japanese Crossword Puzzles: Examination of Performance Improvement

内木 賢吾\*<sup>1</sup> 佐藤 理史\*<sup>1</sup> 駒谷 和範\*<sup>1</sup>  
Kengo Naiki Satoshi Sato Kazunori Komatani

\*<sup>1</sup>名古屋大学 大学院工学研究科 電子情報システム専攻  
Graduate School of Engineering, Nagoya University

This paper describes a system that solves Japanese crossword puzzles, focusing on three extensions that has been recently introduced. Three extensions are: answer-type detection, introduction of a crossword database, and enhancement of knowledge base and experts. They are used in the candidate generation step, which infers candidate words for each clue in a puzzle. With these extensions, the system achieved an average 68% words correct, which is 13% better than the performance of the previous system.

## 1. はじめに

クロスワードパズル(クロスワード)は、よく知られた言語パズルのひとつである。クロスワードのルールの詳細は、言語毎に異なるが、「カギ」と呼ばれるヒントから答となるワードを推定し、「グリッド」と呼ばれるマス目を埋める点は、言語に関わらず共通している。

アメリカ式英語クロスワードを対象としたシステムには、Proverb [1] や DR.FILL [2] がある。これらのシステムは、平均的な人間を超える能力を持つ。DR.FILL は、2010 年開催のアメリカのクロスワード大会 (American Crossword Puzzle Tournament) の問題を解いた場合、600 人中 38 位に相当する成績となることが報告されている。

一方、日本語クロスワードを対象とした研究には、佐藤 [3] や神保 [4] の研究がある。これらのシステムの性能は低く、平均的な人間の能力を遥かに下回っている。

我々は、日本語クロスワードを人間並みに解くシステムの実現を目指し、昨年度よりシステムの開発を行っている [5]。本論文は、新たに導入した 3 つの拡張機構が、システムの性能向上にどの程度寄与するかを示す。

以下、2 節で本システムの全体像を述べ、3 節では、拡張機構について説明する。4 節では、本システムを新聞掲載のクロスワードに適用した結果を示す。

## 2. システム構成

我々が作成しているシステムの全体像を図 1 に示す。本システムは、次の 2 ステップでクロスワードを解く。

## 1. カギを解く

各カギに対し、そのカギの答となるワードの候補とスコア(候補の確からしさ)を推定する。一般に、複数の候補を出力する。

## 2. グリッドを埋める

得られたワード候補を用いて、交差する文字が一致するようなグリッドの埋め方を求め、クロスワードの最終解を出力する。

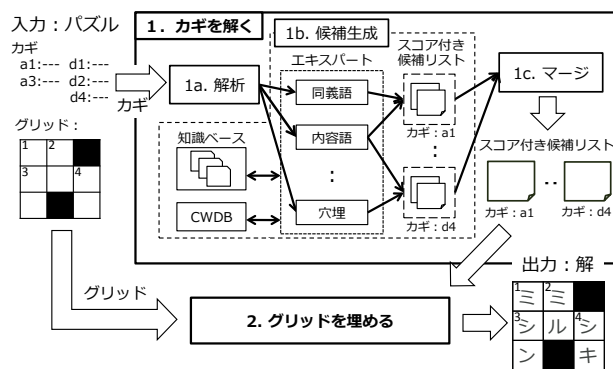


図 1: システム構成

このうち、ステップ 2 には、Proverb のアルゴリズム [1] を用いる。

ステップ 1 は、それぞれのカギに対して独立に実行するステップで、次の 3 つのサブステップから構成されている。

## a. 解析

カギを、MeCab+Unidic および CaboCha を用いて形態素・構文解析する。次に、表層的特徴に基づいて、カギを、8 つのタイプに分類する (表 1)。

## b. 候補生成

カギとその解析結果をエキスパート群に渡し、そのカギの答(ワード)の候補リストを得る。ここで、エキスパートとは、知識ベース (Knowledge Base: KB) の一部を参照して、特定のタイプのカギを解くモジュールである。KB は、本システムが利用する知識の総体であり、国語辞典、同義語辞書、反対語辞書、表記リストなどから構成されている。例えば、同義語エキスパートは、同義語を要求するカギ(「長所のこと(3)」)に対し、同義語辞書を検索し、得られた検索結果(「取り柄」、「利点」)を候補として返す。

## c. マージ

複数のエキスパートから得られた複数の候補リストを、1 つの候補リストに集約し、それぞれのワード候補に対する最終的なスコアを決定する。各エキスパートに対して

連絡先: 内木賢吾, 名古屋大学大学院 工学研究科 電子情報システム専攻, 〒464-8603 愛知県名古屋市千種区不老町 C3-1(631), 052-789-4435, k.naiki@nuee.nagoya-u.ac.jp

表 1: カギの表層タイプ分類

文数	特徴	表層タイプ	カギの例
1つ	穴がある	穴埋一般	□も木から落ちる (2)
		穴埋列挙	□足、□金、□髪 (2)
		穴埋複数	□には□がある (2)
	指示語がある	指示語	作家はこれが仕事 (4)
その他	列挙	炊事、洗濯、掃除 (2)	
	一般	山のふもと (2)	
複数	AはB、Cは?	対比	父はパパ。母は? (2)
	その他	マルチプル	ソロ。一人で歌う (4)

括弧内の数字は答の文字数を表す

定義される信頼度はそれぞれ異なるため、各候補に対して、信頼度とスコアを乗算した値を足し合わせる。このとき、異なる  $m$  個の候補リストに同じ候補が含まれる場合、足し合わせた結果をさらに  $m^2$  倍する。最後に、それらの値の総計が 1 になるように正規化し、それらの値を各ワード候補の最終的なスコアとする。

### 3. 拡張機構

本節では、新たに導入した 3 つの拡張機構について説明する。

#### 3.1 答のタイプ推定

日本語クロスワードの答となるワードは、原則として名詞である。しかしながら、カギの中には、「車を□する (4)」のように、答の品詞細分類を強く示唆するものや、「鼻が長い動物 (2)」のように、答の意味タイプを強く示唆するものが存在する。このような答のタイプを適切に推定できれば、そのタイプに合致する候補のスコアを上げたり、合致しない候補のスコアを下げたりすることが可能となる。

現時点までに、次の 2 種類のタイプ推定を実装した。これらのタイプ推定は、ステップ 1a (解析) の最後に行う。

##### 1. 品詞タイプ推定

カギの末尾が「□+する」、「~すること」、「動詞+する」である場合、そのカギの答の品詞タイプとして、サ変名詞を割り当てる。

##### 2. 意味タイプ推定

カギの文末が、「動物」、「魚」、「植物」、「料理」など 22 種類の名詞に該当する場合、そのカギの答に、それぞれの意味タイプを割り当てる。たとえば、「鼻の長い動物 (2)」というカギに対して、意味タイプ「動物 (1.5501)」を割り当てる。ここで、1.5501 という記号列は、分類語彙表の分類番号に対応する。

カギの答に、品詞タイプや意味タイプが割り当てられた場合は、ステップ 1c (マージ) の最後に、候補のスコアに対して、以下のような加点や減点を行う。

##### 1. 品詞タイプに基づく加点

ワード候補がサ変名詞の場合に、スコアを  $k$  倍する。 $k$  の値は、タイプ推定に用いたパターン毎に異なり、5 (「□+する」)、3 (「~すること」)、2 (「動詞+する」) を用いる。

##### 2. 意味タイプに基づく加点・減点

ワード候補の分類語彙表における分類番号と、意味タイプの分類番号が一致する場合、スコアを 5 倍する。一致しない場合は 1/5 倍する。ワード候補が分類語彙表に含まれない場合は、スコアを変更しない。

#### 3.2 CWDB の実装

クロスワードでは、あるワードを導くカギとして、同一のカギ、あるいは、似たようなカギが使われることがある。特に、同一作者のクロスワードにおいては、過去のクロスワードで使用したカギを使い回すことが、しばしば行われる。このような理由により、既存のクロスワードのカギとワードのペアは、カギを解くための強力な知識源となる。事実、Proverb では約 35 万件、Dr.FILL では約 380 万件のカギとワードのペアをデータベース化して用いており、このことが、これらのシステムの高性能化につながっている。

我々の知る限り、日本語クロスワードの大規模な実例データベースは存在しない。そこで、今回、新聞や Web などから 296 個のクロスワードを収集し、カギとワードのペア 10,817 件をデータベース化した。本論文では、これを CWDB と呼ぶ。本システムにおいて、CWDB は、知識ベースを構成する 1 つとして位置付けられる。

CWDB は、次の 2 つのエキスパートで参照される。

##### 1. 完全一致検索エキスパート

引数として与えられたカギで CWDB を検索し、カギ文字列をそのまま含むエントリーを求め、そのエントリーのワードを候補として出力する。複数の候補が得られた場合、それらはすべて同一スコアとする。

##### 2. 内容語検索エキスパート

引数として与えられたカギを MeCab+Unidic で形態素解析し、内容語を抽出する。得られた内容語で CWDB を検索し、それらの内容語を 1 つ以上含むエントリーを求め、そのエントリーのワードを候補として出力する。各候補のスコアは、一致した内容語の TF-IDF 値の和とする。

#### 3.3 KB・エキスパートの追加・増強

##### 3.3.1 KB・エキスパートの追加

「フィンランドの首都 (5)」のようなカギを解くためには、国名と首都名の対応知識が必要であり、そのような知識が利用可能であれば、高い信頼度で一意にカギを解くことができる。このような知識とそれを利用するエキスパートとして、以下の 4 つを新たに導入した。

##### 1. 国・首都検索

人手で作成した国名首都辞書 (196 エントリ) を用いる。「〈国名〉の首都」や「首都は〈都市名〉」のようなカギに対して候補を出力する。

##### 2. 都道府県・県庁所在地・特産品検索

ウェブサイト「あのみちこのまち<sup>\*1</sup>」、「日本の名物特産品・特産物<sup>\*2</sup>」より抽出・作成した都道府県辞書 (47 エントリ) を利用する。各都道府県の項目には、県庁所在地、シンボルの木・花・鳥、観光名所、行事、特産物が記述されている。「〈都道府県名〉の名物」や「〈特産物〉で有名な県」のようなカギに対して候補を出力する。

##### 3. 元素記号・物質名検索

人手で作成した周期表辞書 (118 エントリ) を用いる。例えば、「元素記号は〈元素記号〉」や「〈物質名〉の元素記号」のようなカギに対して候補を出力する。

##### 4. N 番目検索

十二支・太陽系・アルファベット・指・曜日の 5 つのカテゴリに対して、N 番目を表す数字 N と語との対応

\*1 <http://www.gds.ne.jp/html/top.html>

\*2 <http://www.gojapan.jp/toku3/>

表 2: システムのワード正解率

作者	パズル名		カギ数	ワード正解率									
				Full		w/o A-type		w/o CWDB		w/o New KB		w/o All	
ポラン	開発	計	376	289	77%	282	75% (-2%)	241	64% (-13%)	260	69% (-8%)	216	57% (-20%)
	評価	120907	29	29	100%	29	100%	28	97%	28	97%	28	97%
		120921	28	26	93%	26	93%	21	75%	26	93%	21	75%
		121019	29	27	93%	27	93%	24	83%	23	79%	22	76%
		121221	28	23	82%	23	82%	19	68%	19	68%	18	64%
		130215	28	22	79%	20	71%	22	79%	22	79%	20	71%
		130118	26	19	73%	18	69%	13	50%	21	81%	10	38%
		121102	30	21	70%	21	70%	19	63%	18	60%	19	63%
		130201	27	18	67%	18	67%	14	52%	18	67%	14	52%
		121116	28	15	54%	16	57%	20	71%	15	54%	17	61%
		121005	27	12	44%	11	41%	11	41%	11	41%	8	30%
	計	280	212	76%	209	75% (-1%)	191	68% (-8%)	201	72% (-4%)	177	63% (-13%)	
ろだん	開発	計	333	220	66%	219	66% (-0%)	215	65% (-1%)	225	68% (+2%)	193	58% (-8%)
	評価	130111	28	23	82%	23	82%	11	39%	22	79%	3	11%
		130222	24	19	79%	19	79%	16	67%	19	79%	16	67%
		121109	30	23	77%	21	70%	14	47%	23	77%	7	23%
		121012	26	18	69%	17	65%	17	65%	17	65%	16	62%
		120914	27	17	63%	17	63%	12	44%	16	59%	12	44%
		121123	30	17	57%	22	73%	19	63%	22	73%	20	67%
		130125	26	14	54%	13	50%	15	58%	14	54%	15	58%
		130208	24	11	46%	11	46%	15	63%	12	50%	13	54%
		121026	26	11	42%	11	42%	11	42%	11	42%	11	42%
		120928	25	4	16%	4	16%	12	48%	4	16%	12	48%
	計	266	157	59%	158	59% (-0%)	142	53% (-6%)	160	60% (+1%)	125	47% (-12%)	
開発用合計			709	509	72%	501	71% (-1%)	456	64% (-8%)	485	68% (-4%)	409	58% (-14%)
評価用合計			546	369	68%	367	67% (-1%)	333	61% (-7%)	361	66% (-2%)	302	55% (-13%)

(例: 十二支, 4, ウサギ(卯)) を定義した辞書を用いる。「〈カテゴリ〉の〈N〉番目」のようなカギに対して候補を出力する。

### 3.3.2 KBの増強

#### 1. 同義語辞書

「百科辞書\*3」(16,644語)、「カタカナ英単語\*4」(15,174語)、「日本語 Wikipedia\*5」(329,018語)より同義語を抽出し、新たな同義語辞書(357,833語)を作成した。この辞書を、既存の同義語辞書(分類語彙表:101,070語)と併せて使用する。

#### 2. 百科事典

大規模な百科事典として、日本語 Wikipedia を利用する。記事のタイトルを見出し語、本文の1文目を説明文とみなし、説明文に対する内容語検索を行う。検索の対象とする記事(見出し語)は、記事のタイトルに英語や記号を含まない739,025語である。

## 4. 実験

### 4.1 使用したクロスワード

本実験では、2012年から2013年にかけて朝日新聞名古屋版に掲載された、8×8のクロスワード47問(カギ1255個)を用いた。これらのクロスワードの作者は、ポラン、ろだんの2名である。これらのクロスワードを、以下に示すように、システム開発用、システム評価用の2つに分割して使用した。

#### 1. システム開発用セット

2012年1月-8月に掲載された27問(カギ709個)。内訳は、ポラン作成が14問(カギ376個)、ろだん作成が13問(カギ333個)である。これらのクロスワードは、3節で述べた拡張機構を開発するために使用した。

#### 2. システム評価用セット

2012年9月-2013年2月に掲載された20問(カギ546個)。内訳は、ポラン作成が10問(カギ280個)、ろだん作成が10問(カギ266個)である。これらのクロスワードは、システム開発においては全く使用しておらず、完全に新規のクロスワードである。

### 4.2 システム性能評価および考察

システム開発用およびシステム評価用の各クロスワードをシステムに入力し、得られた最終結果において、何個のワードが正解と一致するのかを調べた。その結果を表2に示す。

この表において、Fullの列が、3つの拡張機構を備えた、現在のシステムの性能を示している。これに対して、w/o A-Type、w/o CWDB、w/o New KBは、それぞれ、拡張機構の1つを除いた場合の性能を表す。さらに、w/o Allは、すべての拡張機構を除いた場合の性能を示す。これらの列の括弧内の数値は、Fullとの性能の差を表す。

この表より、以下のことがわかる。

- 3つの拡張機構の導入により、性能が向上した  
拡張機構を用いた場合(Full)を用いなかった場合(w/o All)と比較すると、開発用セットでは14%、評価用セットでは13%の性能向上が見られる。それぞれのセットを作者別に分けた場合も、すべての場合で性能向上が見られる。これらの事実により、新たに導入した3つの拡張機構は、全体として、有効に機能しているとみなすことができる。

\*3 <http://pddlib.v.wol.ne.jp/japanese/index.htm>

\*4 <http://homepage2.nifty.com/katakanaEnglish/>

\*5 <http://ja.wikipedia.org/wiki/>

2. 3つの拡張機構の中では、CWDBの導入が最も性能向上に寄与した

それぞれの拡張機構の性能向上への寄与は、それぞれの拡張機構を除いた場合の性能低下の割合で見積ることができる。開発用および評価用のどちらのセットにおいても、CWDBを除いた場合 (w/o CWDB) の性能低下が一番大きく、それぞれ、-8%と-7%である。

3. 他の2つの拡張機構も、一定の効果が見られる

開発用および評価用のセットにおいて、CWDBのみを除いた場合と、3つの拡張機構をすべて除いた場合との差は、どちらも6%である。この差は、CWDBを除く2つの拡張機構による効果である。

4. 性能は、クロスワードの作者に依存する

ポラン作成のクロスワードに対する性能は、開発用セットと評価用セットにおいて、それぞれ77%、76%であり、ほとんど性能に差が見られない。これに対して、ろだん作成のクロスワードに対する性能は、それぞれ66%、59%であり、ポランに対する性能より低く、かつ、2つのセットにおいて、差がある。

新たに導入した拡張機構は、システムの性能向上に結び付いたと言えるが、現在のシステムの性能は、平均的な人間の能力には及ばない。それぞれの拡張機構には、機能強化の余地が残っており、それらを実装することで、ある程度の性能の向上は期待できる。しかしながら、人間並みの能力を達成するためには、さらに、20%程度の性能向上が必要であり、そのためには、新たな機能の導入が不可欠であろう。その候補として、連想や言い換えの導入を検討する予定である。

### 4.3 孤字と精度の関係

タテ・ヨコでワードが交わっていないマス目のことを、文献 [6] では「孤字」と呼んでいる。孤字の存在は、アメリカ式英語クロスワードでは許容されていないが、日本語クロスワードでは許容されており、ほとんどのパズルに存在する\*6。孤字の存在は、日本語クロスワードの顕著な特徴の一つである。

孤字以外のマス目ではタテとヨコの2つのワードが交差するため、2つのワードを相互に限定(制約)する力が働く。一方、孤字では、ワードが交差しないため、このような力は働かない。このため、孤字を含むワードの正解率は、他のワードの正解率より低くなることが予想される。

もし、この予想が正しく、かつ、2名の作者において、ワードに孤字を含む割合が異なれば、それが原因で、ワード正解率の差が生じている可能性がある。この仮説を検証するために、それぞれの作者において、ワードに孤字を含む割合、および、孤字を含まない場合と含む場合のワード正解率を調査した。

その結果を表3に示す。この表では、それぞれの作者に対して、ワード長(L)毎に、孤字を含まない場合と含む場合のワード正解率を示した。この表から次のことがわかる。

1. 孤字を含まない場合と含む場合を比較すると、後者の方がワード正解率が低い  
ワード正解率は、孤字を含まない場合が71%、含む場合が61%であり、10%の差が見られる。これは、上記の予想を裏付けるものである。正解率の差は、作者別、ワード長別でも、ほぼ一貫して観察される。
2. ワードに孤字を含む割合は、いずれの作者も約1/3強で、同程度である

表3: 弧字の有無とワード正解率

作者	L	弧字を含む	ワード正解率			
			弧字を含まない		弧字を含む	
ポラン	2	27%	52/ 66	79%	18/ 25	72%
	3	38%	50/ 61	82%	27/ 37	73%
	4	37%	28/ 39	72%	16/ 23	70%
	5以上	44%	13/ 16	81%	8/ 13	61%
	計	35%	143/182	79%	69/ 98	70%
ろだん	2	16%	39/ 61	64%	6/ 12	50%
	3	40%	37/ 55	67%	22/ 36	61%
	4	49%	17/ 34	50%	15/ 33	45%
	5以上	57%	10/ 15	66%	11/ 20	55%
	計	38%	103/165	62%	52/101	51%
合計		36%	246/347	71%	121/199	61%

ポランは、短いワードにおいて孤字を含むことが多く、一方、ろだんは、長いワードにおいて孤字を含むことが多い。このように、孤字の使用に関しては差が見られるが、孤字を含む割合は、全体としては同程度(35%と38%)である。すなわち、ワード正解率の差の原因を、孤字の過多に求めることはできない。

上記の結果は、作者に依存したワード正解率の差の原因は、グリッドの配置ではなく、カギの難しさあることを示唆するものである。今後、この点についても、さらに検討していく予定である。

### 参考文献

- [1] Michael L. Littman, Greg A. Keim, and Noam Shazeer. A Probabilistic Approach to Solving Crossword Puzzles. *Artificial Intelligence*, Vol.134:1-2, pp.23-55, 2002.
- [2] Matthew L. Ginsberg. DR.FILL: Crosswords and an Implemented Solver for Singly Weighted CSPs. *Journal of Artificial Intelligence Research*, Vol.42, pp.851-886, 2011.
- [3] 佐藤理史. 日本語クロスワードパズルを解く, 情報処理学会 自然言語処理研究会 NL-147-11, pp.69-76, 2002.
- [4] 神保 一樹, 高村 大也, 奥村 学. 拡張 Potts Model を用いたクロスワードパズルの解き方, 第22回人工知能学会全国大会, 2J3-01, 2008.
- [5] 内木賢吾, 佐藤理史, 駒谷和範. 日本語クロスワードパズルのカギの解法. 情報処理学会第74回全国大会講演論文集, No.2, 3R-5, pp.267-268, 2012.
- [6] うさお著. 『クロスワードパズル事典』. 東京堂出版, 2005.

\*6 孤字なしで、大きなサイズの日本語クロスワードを作ることはできないと言われていた。