Vol.2013-CSEC-61 No.30
Vol.2013-IOT-21 No.30
2013/5/10

人間ロボット判別テストの バリアフリー化のための言語的作問技法

山口 通智[1,2,a)]　中田 亨[2,b)]

概要：ネット上のユーザ認証において人間と人工知能（ロボット）を判別する従来のテストは、人間とロボットとの知覚認識能力の差を用いるものが多い。これは知覚に障害のあるユーザに対しては障壁になることが指摘されてきた。本研究は、この障壁を解消するために、特定の知覚チャンネルに依存しない言語的な質問を作成する技術を開発した。人とロボットの文章の解釈能力差を判別の根拠とし、文意・文脈の認識を問う質問を作る。また無制限の新規な作問を可能とするためにネット上に投稿される文章を採取して利用するものである。

キーワード：CAPTCHA, チューリングテスト, 言語的作問, 文脈解釈, アクセシビリティ

A Generating Method of Accessible Verbal Questions for substitution of CAPTCHA-like tests

Abstract: In the case of user authentication on the network, most of the existing methods which tell humans and computers apart automatically make use of the difference in perceptual recognition between humans and computers. Researchers point out that it is difficult for visual and/or hearing impaired people to leverage it. In this paper, we propose a method which generates linguistic questions to avoid relying on the specific perceptual ability. Based on the difference of a cognitive ability between humans and computers, we generate linguistic questions such as contextual interpretation. Moreover, we utilize contents on the network to generate almost infinite kinds of questions.

Keywords: CAPTCHA, Turing test, linguistic comprehension, contextual interpretation, accesibility

1. はじめに

1.1 人間ロボット識別技術の現状

近年の情報技術とコンピュータ性能の発展により、我々は多くのサービスをオンライン上で享受できる。その一方で、不正な「ロボット」（サービスを自動操作するプログラム）によるアカウントの大量生成、不正な多重投票、パスワード認証に対する総当たり攻撃への対策が、サービス提供者にとっての重要問題となっている。このような問題を解決するものとして、CAPTCHA (Completely Automated Public Turing Test To Tell Computers and Humans Apart) [17] と呼ばれる、コンピュータによる人間ロボット判別テスト (チューリングテスト) が広く使われている。CAPTCHA は、人間には解けるが現在のロボットでは容易に解くことができない AI (Artificial Intelligence) 問題の困難性を安全性の根拠として仮定としている。また、コンピュータによる判別を行うため、作問から回答照合までの一連の処理が自動化されている。その有用性をパスワード認証の例で示す。ロボットにとって CAPTCHA は認証処理ごとに変化するパスワード機能にみえることから、仮に利用者が安易なパスワード（例、「12345」）を設定したとしても、CAPTCHA の持つ安全性までは保証できる。

現在最も広く利用されている CAPTCHA の方式は、図 1 のように、歪曲やノイズが付加された文字列の画像を作り、ユーザにその文字列を判読させ回答させるものである。ま

[1] 筑波技術大学技術科学研究科,
Department of Technology and Sciences, Tsukuba University of Technology, Kasuga 4-12-7, Tsukuba City, Ibaraki, JAPAN, #305-8521.
[2] 産業技術総合研究所セキュアシステム研究部門,
Research Institute for Secure Systems, National Institute of Advanced Industrial Science and Technology (AIST), JAPAN. Umezono 1-1-1, Tsukuba, Ibaraki, JAPAN, #305-8568.
a) ym123202@cc.k.tsukuba-tech.ac.jp
b) toru-nakata@aist.go.jp

図 1 一般的な CAPTCHA の例

た、同様の主旨で、変形した音声をユーザに聞き取らせる方式も利用されている。

しかしながら、これらの方式は、OCR(Optical Character Reader) を用いた Yan ら [21] の攻撃や SVM(Support Vector Machine) を用いた Tam ら [16] の攻撃によって破られることが指摘されている。より強い不明瞭化による対抗策も実施されているが、同時に人間の正答率も落ちてしまう問題が Bigham ら [1] や Bursztein ら [2] により報告されていて、特に音声を用いた CAPTCHA での初見利用者の正答率は、39% にとどまっている。また、ノイズの付加による不明瞭化は対象によっては撹乱の度が過ぎてしまい、例えば視聴覚障がい者の使用する点字には適用できないという欠点がある。

そこで、人間のより高度な認知能力を利用する CAPTCHA 様テスト（CAPTCHA 様のチューリングテスト）が提案されている。提示された画像の意味を問う CAPTCHA 様テストの代表例には、 Asirra [4] がある。この種の問題はロボットにとって非常に難しいと考えられていたが、Golle [7] による攻撃の成功が報告されている。その他多くの方式 [5], [12], [24] が提案されているが、複数の関連画像を準備する必要があるという難点があり、新規未使用の問題を大量に作り出すことには限界がある。音声を用いた チューリングテスト には、動物の鳴き声やブザー音などの特定音声からその発生源を問うもの [10] や音声の断片から全体を推測し聞き取りをさせるもの [25], [27] が提案されている。

また、人間のもつ知識に依存したクイズを用いる方式 [20] も提案されている。しかし、IBM の Watson や Apple の Siri といった自然言語で質問を受け付け、正しい解答をする人工知能の登場により、人とロボットとの差がほぼ無くなりつつある現状では、この方式は無力化しつつあるとの意見がある [6]。

このように、 現状多くの CAPTCHA 様テストにおいて、その危殆化が指摘されている。

さらに、バリアフリーの観点からの問題も指摘されている。人間のより高度な認知能力を利用する方式は、特定の知覚チャネルの使用を強制されるため、画像を使用した方式は視覚障がい者に、音声を使用した方式は聴覚障がい者に、当然対応できない。また、画像と音声の両方サポートする事例もあるが（図 1 ）、どちらかより脆弱な方を不正利用者に攻撃されるという弊害がある。

1.2 目的：人間ロボット判別テストのバリアフリー化

本研究の目的は、前述の問題点を克服し、知覚障害によらず誰でも利用できる CAPTCHA 様テストを考案することである。まず、そのようなテストの要件を列挙しよう。

(1) 【識別性要件】人間には容易に解けるが、現状のロボットには解答が難しい問題を生成できること。
(2) 【問題新規性要件】未使用で新規な問題を無制限かつ自動で作れること。
(3) 【バリアフリー要件】特定の知覚のみの使用に限定されないこと。
(4) 【問題ノンジャンル性要件】限られた知識の有無に判別結果が強く依存しないこと。

このような CAPTCHA 様テストを構成するため、本稿では、文意文脈解釈問題に基づく人間ロボット判別を提案する。これは人間とロボットの間に、文意や文脈の解釈能力の差があることを利用するものである。問題形式の具体なアイデアを挙げれば、複数文を提示しその共通の話題を答えさせる問題や、ある文章の意味的な整合性や自然さを評価させる問題などである。

本研究では、無制限の新規作問を行うため、ネット上に投稿させる文章を採取し、問題生成の「ネタ元」に利用する。

なお、この方式には明らかな欠陥があり、攻撃者が問題文をネット検索しネタ元を調べることで文意や文脈のヒントを得る恐れがある。そこで、文の子音を改変する撹乱をネタ元の文に施し、検索発見されにくくする。現在のロボット性能では、非常に大きい文章集合から類似文を見つけることは計算量的に困難であると仮定できれば、有効な対策と言えよう。

1.3 バリアフリー化に関係する研究

Holman ら [8] は、サイレンや鳥などの身近な事物を画像と音声の両方で提示し、そのいずれによっても回答可能とする方式が提案した。ただし、事物の画像音声の用意の手間に鑑みると、問題新規性要件を十分に満たすことは難しいと考えられる。

文意文脈解釈問題の研究は、人工知能や認知科学の分野で古くから研究課題になっていた。

認知科学では「サリーとアン課題」 [19] は、自分の知識の範囲と他者のそれとを区別できるかを問うテストが有名である。これも計算困難な問題であり、識別性要件を満たすだろう。

ディックの SF 小説 [3] においては、「このカバンは官給品なんだ。赤ん坊の皮でできている。」などと、異様な内容の文章を聞かせ、異様部分に対する身体的・感情的反応の時間遅れを計測する “Voigt-Kampff test” のアイデアが示されている。文意・文脈の理解と常識の発揮の能力差を利用する例としては先駆的なものと言えよう。

文意文脈理解能力に関する研究では、山本ら [23] は、人

間が作った文章と機械翻訳により生成される文章との間で、人間が感じる違和感を人間ロボット判別に用いた。同様に、上原ら [29] の、人間が作った文章とマルコフモデルによる自動合成された文章の比較の研究がある。Christopher [11] は、複数の文の中から内容が関連するものとしないものを選択させる方式を提案した。

ただし、これらの言語的問題のアイデアを問題新規性要件を満たす形で実施するには、問題作成のメカニズムについて具体的な設計が必要である。

2. 提案するチャレンジレスポンスの基本構成

本稿では、文意文脈の理解能力を試す問題（「文意文脈解釈問題」と称す）を用いた CAPTCHA 様テストを構成する。はじめに、本節において CAPTCHA 様テストのチャレンジレスポンスの枠組みを示し、次節以降で提案方式の詳細を解説する。本稿で用いる記法や CAPTCHA 様テストの定義については、付録 A.1、A.2 を参照されたい。以後は、これらの記法や定義ついての解説は省略する。

CAPTCHA 様テストシステムへの要求仕様として、利用者である人間 $\mathcal{P}$ が認証に成功する確率の下限値 $\theta_{\mathcal{P}}$、攻撃者であるロボット $\mathcal{P}^*$ が認証に成功する確率の上限値 $\theta_{\mathcal{P}^*}$、認証に要する時間の上限値 τ_{sys} が与えられているとする。これらをシステムパラメータとする。利用する AI 問題を $\{Q_i\}_{i\in\mathcal{I}}, |\mathcal{I}| \in \mathbb{N}$ とする。CAPTCHA 様テストシステムの構成者は、システムパラメータと利用する AI 問題に合わせて、AI 問題のセキュリティパラメータ $\{\kappa_i\}_{i\in\mathcal{I}}$、一回の認証で出題する問題数 $\{n_i\}_{i\in\mathcal{I}}$、人間と判断する問題の正答数 $\{k_i\}_{i\in\mathcal{I}}$ をそれぞれ最小になるように決定する。[*1]

CAPTCHA 様テストのセキュリティパラメータ $\{(\kappa_i, n_i, k_i)\}_{i\in\mathcal{I}}$ に対して、利用する AI 問題 $\{Q_i\}_{i\in\mathcal{I}}$ が人間にとっては容易でロボットに対して困難[*2]ならば、次のようにバリアフリーな CAPTCHA 様テストシステム $\mathcal{V}$ が構成できる。$\mathcal{V}$ は利用者 $\mathcal{P}$ との対話をおこない、$\mathcal{P}$ を人間と判断した場合には ‘1’ を、そうでない場合には ‘0’ を出力する。

(1) AI 問題の選択
$i \xleftarrow{\$} \mathcal{I}$ となる i を選択する。

(2) 元になる文章の収集
インターネットに接続し、Q_i に必要な言語の文章を収集する。

(3) 作問
取得した文章を利用して、Q_i に対する問題と回答のペア (z, a) を生成する。

(4) 出題
問題 z と回答の選択肢 $(0, \ldots, \kappa_i - 1)$ を、利用者 $\mathcal{P}$ に対して出力する。出力先の機器は、ディスプレイ、点字ディスプレイ、スピーカをサポートする。回答時間を計測するタイマを起動し、$\mathcal{P}$ の回答待ち状態に遷移する。

(5) 回答の確認
$\mathcal{P}$ が回答した a' の受信をトリガにしてタイマを停止し、回答に要した時間 $\tau_{\mathcal{P}}$ を計算する。$\tau_{\mathcal{P}} > \tau_{sys}(\kappa)$ であれば、不正解として扱う。そうでなければ、$a \stackrel{?}{=} a'$ の判定をおこなう。等式が成立するならば正解とし、そうでなければ不正解として扱う。結果を保存し、タイマの値を初期化する。

(6) 認証
前述の (2) から (5) の処理を n_i 回繰り返す。正解数が k_i 以上ならば、‘1’ を出力し、そうでなければ ‘0’ を出力する。

図 2 に、C# を用いて試作した CAPTCHA 様テストサンプルアプリの一例を示す。本稿執筆の時点では、AI 問題ごとに別アプリとして、$(n, k) = (1, 1)$ の条件で作成した予備実験用のものである。読み上げ用のボタンについては、スクリーンリーダを持たない利用者が Microsoft Speech Platform を利用して、手軽に音声出力を試せるように配置した。

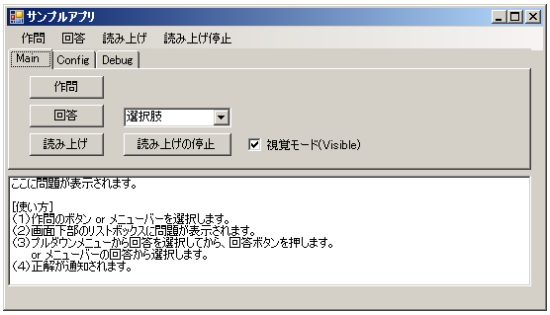

図 2 サンプルアプリの例

3. 文意文脈解釈問題の作成

文意文脈解釈問題を、新規に作り続けるための技法を述べる。

3.1 問題の情報源

文意文脈解釈問題を作るには、元になる文章 SD（Source Document、ネタ元) を必要とする。SD を獲得する方法としては、次のものが考えられる。

(1) インターネットから文章を収集する方法
多様な文章を大量に得られるが、それは公開情報であるから、攻撃者も同じ文章を検索によって収集できてしまう。SD を回答とするような方式 [23], [29] には、特に向きである。

(2) 秘匿されたデータベース内の文章を利用する方法
秘匿を安全性の根拠にするのは好ましくない。また、

*1 参考:節 4、付録 A.2
*2 正式な定義については、付録 A.2 を参照すること。

秘匿された SD であっても問題として公開されてしまうので、大量に同種の問題に挑戦することで、攻撃者は SD を収集できる。

(3) システム利用者に文章を入力させる方法
攻撃者が文章生成に参加することで SD を登録できてしまう。また、(2) と同様の問題も発生する。

さらに (2)、(3) の方式では、SD を (1) ほどに大量に低コストで蓄積できるわけではないから、無制限な作問にも不向きである。

よって本研究では、無制限な新規作問を実現するため方式 (1) を採用する。

前述の問題点を解決するため、次の仮定を用いた SD の改変を行う。

仮定 1. 子音交替により改変した文章の性質. *SD* の一部を子音交替によって改変した文章は、次の性質を持つ。

(1) その改変率が一定以下ならば、人間は元の文章と同様の意味を理解できる。
(2) 改変した文章から、元の SD を探すのは困難である。■

仮定 1-(1) の根拠は、 [13], [14] などによる。本稿で扱う子音交替は、方言などに見られる単語の子音の違いを指す。例えば、ザ行からダ行への子音交替では、「ざぶとん」を「だぶとん」と改変する。子音交替による改変は、人間の音声認識における音韻修復効果 [26] を活用できるため、単純な雑音挿入と比較して、音声による認識がしやすいと考えられる。点字による触読については、元々ある程度の読み間違いを人間の頭で訂正しながら読み進めることから、仮定 1 の成立が期待できる。

仮定 1-(2) の根拠は、次の仮定による。

仮定 2. 非常に大きなサイズの文章集合に対する高速類似文検索. インターネット上に存在する文章から、検索クエリに用いた文と類似するものを、*CAPTCHA* 様テストシステムが許容する実用的な時間 τ_{sys} 以下で取得することは難しい。■

仮定 2 の根拠を示すために、現在提案されている高速な類似文字列検索アルゴリズムの一つとして、SimString [28] の性能から検討を始める。SimString は、単語相当の長さの類似文字列検索を高速に実行できる。しかしながら、仮定 2 で想定している検索クエリは複数単語からなる文である。また、インターネット上から文章を取得するため、比較対象となる文章の集合サイズは、 SimString が想定しているものよりも非常に大きい。従って、SimString のような高速類似文字列検索アルゴリズムを用いたとしても、現在のロボットに対して仮定 2 は成立すると考えられる。

仮定 2 の根拠をより確実にするために、提案方式では、タイムスタンプを確認しながら新旧混在の SD 収集を検討する。この効果について、前述の類似文字列検索を利用した攻撃者を例に挙げて検討する。検索エンジンの多くは、最初に対象のデータセットにインデックスを付ける。しかしながら、インターネットのように新たなデータが常時発生するような場合では、データ発生からインデック化の間にタイムラグが発生する。タイムスタンプの新しい SD の利用は、インデックス化前のデータが検索対象から外れることや、インデックス化に要する攻撃者の処理時間増大を狙う。これは、特にクローラの自動巡回によるインデックス化方式に有効である。更新時間駆動型の方式には効果が小さいが、こちらは検索精度が劣るので影響は小さいと考える。一方、タイムスタンプの古い SD の利用は、対象データセットのサイズを大きくする狙いがある。対象データセットに対して、あらかじめ子音交換処理をほどこし記憶する攻撃が考えられるが、対象データセットのサイズが大きければ、その方式は現実的ではない。

3.1.1 SD の取得

SD 取得方式を具体的に示す。本研究では、青空文庫、Wikipedia、Twitter を取得先として利用した。[*3]特に Twitter では頻繁に新規の文章が生成されることから、タイムスタンプを意識して効率よく新しい SD を収集できる。

作問要求をトリガとして SD 取得を開始する。青空文庫の利用では、公開されている作品リストからランダムに選択する。Wikipedia の利用では、「おまかせ表示」機能を用いる。Twitter の利用では、API [*4] を用いる。取得した SD に対し、顔文字の削除などの加工をしてから、文字列 m 単位に分割する。m を要素とする集合を $\mathcal{M}$ とする。作問に使用する文字数 $\ell(\kappa)$ に対して、$\ell(\kappa) \leq \sum_{i \in |\mathcal{M}|} |m_i|$ を満たす文字量が必要になる。m への分割は、「。」などの文末記号や「、」などで区切り、$|m| \leq \ell(\kappa)/\kappa$ を目安にして文字列の長さを調整する。もし、取得はしたが作問に利用しなかった m が存在すれば、それを再利用してもよい。

3.1.2 子音交替

子音交替は、問題文表示の直前におこなわれる。言語 L に含まれるすべての行の種類を $\mathcal{U}_L$ とし、文字列 s に含まれるすべての行の種類を $\mathcal{U}_s$ とする。$u(\in \mathcal{U}_s)$ 行から v 行への子音交替を i 箇所適用することを $f(u,v,i)$ と表す。s に対して $f(u,v,i)$ を j 回適用する子音交替関数は、$\mathbf{F}^{\{f(u_h,v_h,i_h)\}_{h\in[1,j]}}(s,j)$ と表す。ただし、$\exists h, h'$ に対して $h \neq h'$ ならば $u_h \neq u_{h'}$ である。子音交替関数は、次のように動作する。(i,j) の値は、システムパラメータを考慮して決める。

(1) s より $\mathcal{U}_s$ を計算する。
(2) u, v を次のように選択する。

$$u \xleftarrow{\$} \mathcal{U}_s, \quad v \xleftarrow{\$} \mathcal{U}_L \backslash u, \quad \mathcal{U}_s \leftarrow \mathcal{U}_s \backslash u$$

(3) s 中に含まれる u 行の文字数を超えない範囲で、 i を決定する。

[*3] これらのサービスを利用する際の著作権問題については、本稿では考慮しないことにする。
[*4] Twitterizer/Twitterizer · GitHub, https://github.com/Twitterizer/Twitterizer

(4) $f(u, v, i)$ を s に適用する。
(5) (2) から (4) の処理を、j 回繰り返す。

3.2 文意文脈問題の安全性と問題作成

本稿で提案するチューリングテストの安全性は、AI 問題としての文意文脈解釈問題の困難性に基づいている。その困難性の仮定の詳細を記述する。

仮定 3. *AI* 問題の困難性. *SD* として収集、加工した文字列の集合を $\mathcal{M}$ とし、その要素を m とする。文字列の属する言語 L を明示的に示す場合、$\mathcal{M}_L$ のように表す。Extr を、指定されたキーワードから関連する文字列を抽出する関数とする。**Corp** をコーパスを収集する関数とし、**Stri** をコーパスからランダムな文章 *WD*（*Word Salad*）を生成する関数とする。$\mathbf{Tras}_{L_0 \to L_1}$ を言語 L_0 から $L_1 (\neq L_0)$ への機械翻訳とする。**F** を、前述した子音交換関数の省略表記とする。$\mathcal{G}_i$ を作問アルゴリズム、$\mathcal{H}_i$ を作問空間から回答空間への写像とすれば、次に示す $\{Q_i = (\mathcal{G}_i, \mathcal{H}_i)\}_{i \in [0,2]}$ は *AI* 問題であり、現在のロボットで解くことは難しい。

(1) 図 3 の $\mathcal{G}_0$ に示される複数の文からその文章の意味を解釈する問題 Q_0
(2) 図 3 の $\mathcal{G}_1$、$\mathcal{G}_2$ に示される自然な文章 NS(Natural Sentence) かどうかを判別する問題 Q_1, Q_2 ■

作問アルゴリズムによる各 AI 問題の例を図 4 に示す。最初の $\mathcal{G}_0, \mathcal{G}_1, \mathcal{G}_2$ の作問例は、子音交替をする前のものである。最後の例は、$\mathcal{G}_1$ の作問例に対して、子音交替を適用したものである。

仮定 3 の根拠は、仮定 1 と自然言語処理における意味解釈や NS 判別の困難性による。以降は、各 AI 問題の具体的な構成法とその困難性について示す。

3.2.1 AI 問題 Q_0 の構成

Q_0 では、キーワードに関連した文字列の抽出 (**Extr**) に検索エンジンを利用する。しかしながら、単純にキーワードを用いて検索をおこなうと、キーワードが含まれた問題文が生成されてしまう。この対策として、本研究ではシソーラス[*5] を利用した検索をおこなう。さらに、ダミーの情報を入れることで、意味解釈に対するノイズを加えている。

作問アルゴリズム $\mathcal{G}_0$ は、回答 a に相当するキーワード *keyword* とダミーのキーワード *dummy* を決める。サンプルアプリでは、あらかじめキーワード候補の集合 $\mathcal{K}$ を用意しておき、$keyword \xleftarrow{\$} \mathcal{K}$, $dummy \xleftarrow{\$} \mathcal{K} \backslash keyword$ のように取得する。

得られたキーワードを用いて、次のように検索をおこなう。

(1) *keyword*, *dummy* に対応するシソーラスをオンラインで取得し、検索クエリ文字列集合 $\{q^{(k)}\}, \{q^{(d)}\}$ としてそれぞれ取得する。

[*5] 類語辞典・シソーラス - Weblio 辞書, http://thesaurus.weblio.jp/

$(z, a) \leftarrow \mathcal{G}_0(\kappa, aux)$
  $(\mathcal{M}, keyword) \leftarrow aux$
  $(s_0, \ldots, s_{\kappa-1}) \leftarrow \mathbf{Extr}(\kappa, \mathcal{M}, keyword)$
  for all i such that $i \in [0, \kappa - 1]$ **do**
    $z_i \leftarrow \mathbf{F}(s_i)$
  end for
  $z \leftarrow (z_0, \ldots, z_{\kappa-1})$
  $a \leftarrow keyword$
$(z, a) \leftarrow \mathcal{G}_1(\kappa, aux)$
  $\mathcal{M} \leftarrow aux$
  $\{c_j\}_{j \in \mathbb{N}} \leftarrow \mathbf{Corp}(\kappa, \mathcal{M})$
  $x \xleftarrow{\$} [0, \kappa - 1]$
  $s_x \leftarrow \mathbf{Stri}(\{\mathbf{c_j}\}_{\mathbf{j} \in \mathbb{N}})$
  $z_x \leftarrow \mathbf{F}(s_x)$
  for all i such that $i \in [0, \kappa - 1] \backslash x$ **do**
    $m_i \xleftarrow{\$} \mathcal{M}$
    $z_i \leftarrow \mathbf{F}(m_i)$
  end for
  $z \leftarrow (z_0, \ldots, z_{\kappa-1})$
  $a \leftarrow x$
$(z, a) \leftarrow \mathcal{G}_2(\kappa, aux)$
  $(\mathcal{M}_{L_0}, \mathcal{M}_{L_1}) \leftarrow aux$
  $x \xleftarrow{\$} [0, \kappa - 1]$
  $m_x \xleftarrow{\$} \mathcal{M}_{L_1}$
  $s_x \leftarrow \mathbf{Trans}_{L_1 \to L_0}(m_x)$
  $z_x \leftarrow \mathbf{F}(s_x)$
  for all i such that $i \in [0, \kappa - 1] \backslash x$ **do**
    $m_i \xleftarrow{\$} \mathcal{M}_{L_0}$
    $z_i \leftarrow \mathbf{F}(m_i)$
  end for
  $z \leftarrow (z_0, \ldots, z_{\kappa-1})$
  $a \leftarrow x$

図 3 作問アルゴリズムの概要

(2) 次の処理を $\kappa - 1 (> 1)$ 回繰り返す。
- $q \xleftarrow{\$} \{q^{(k)}\}$ とし、q をキーワードに検索を実行する。
- 検索結果の一つをランダムに選び保存する。

(3) $\kappa \geq 3$ ならば、$q' \xleftarrow{\$} \{q^{(d)}\}$ となる q' に対して、(2) に相当する処理を一度おこなう。

取得された文字列に対して子音交替を適用することで、問題文 z を得る。選択肢については、$\mathcal{K} \leftarrow \mathcal{K} \backslash \{keyword, dummy\}$ から、ランダムかつ互いに異なる要素を $\kappa - 2$ 個 *keyword*,*dummy* に混ぜて生成する。

このように生成された文意文脈解釈問題は、形態素解析を用いたキーワード抽出をおこなう攻撃者に対して有効であると考える。子音交替の適用は、形態素解析の精度を落とす効果がある。また、検索文字列に子音交替が適用された場合には、検索語をあいまいにすることができる。

3.2.2 AI 問題 Q_1 の構成

Q_1 では、WS を不自然な文章として用いる。WS の生成

$\mathcal{G}_0$ による作問例 (文章のキワードを抽出)
選択肢: (1) 政治、(2) 健康、(3) 経済、(4) 天気
- 国務をきちんとこなして欲しい。
- 参院態勢でみんな混乱　日本はどうなる？
- 資本主義社会では、経済理想をもって、
- 健康診断に費用がかかる。

回答: (1)
$\mathcal{G}_1$ による作問例 (ワードサラダの選択)
(1) 七時過ぎに夕食を食べた。
(2) もし晴れたなら遊園地に困った!
(3) お金が足りず本が買えない。
(4) 益体のないことを考えていると、
回答: (2)
$\mathcal{G}_2$ による作問例 (機械翻訳文の選択)
(1) 経済が 2 パーセント成長すると予測します。
(2) ビールのボトルケースを買ったとき、
(3) はじめまして！よろしくお願いします！
(4) 少年と明日よく再び掛かっていること。
回答: (4)
$\mathcal{G}_1$ の作問結果に子音交替を適用したもの
(1) ナナジスジニ　ユウヒョクヲタベタ。
(2) モヒハレタナラ　ユウエロチニコマッタ！
(3) オガネガタリズ　ホロガガエナイ。
(4) ヤクタイノナイコトヲ　ガロガエテイルト、

図 4　作問の例

には、マルコフモデルに基づく方式がよく利用される [29] が、本研究では異なる方式を利用する。$m \in \mathcal{M}$ の単位で形態素解析 [9] と構造解析 [15] を適用して、コーパスの生成をおこなう。コーパスは、文節 [*6]や係り受け構造のものを構成単位とする。係り受け構造は二個以上の文節からなり、ある文節に着目したとき、それに連続して係る文節の数を p で表す。係り受け構造の例を次に示す。

「もし明日の朝が晴れならば」の係り受け解析結果
- もし → 晴れならば
- 明日の → 朝が → 晴れならば

「晴れならば」に対する係り受け構造

$p = 2$.
- もし → 晴れならば
- 朝が → 晴れならば

$p = 3$.
- 明日の → 朝が → 晴れならば

$\ell(\kappa)/\kappa$ 以下を目安とした文字数の範囲で、コーパスからランダムに要素を取り出して WS を構成する。作成された WS は、文法的には正しいと推測される。

このように生成された文意文脈解釈問題は、ワードサラダの検出攻撃 [22] に対して有効であると考える。係り受け構造を WS の構成要素とすることで、マルコフモデルでは生成できない「自然な句」を含んだ文章を構成できる。その一方で、人間が WS を NS と誤判断する影響について考慮する必要がある。

[*6] 自立語とそれに後続する付属語で構成される

3.2.3 AI 問題 Q_2 の構成

Q_2 では、機械翻訳結果を不自然な文章として利用する。[23] と異なり、認証で使用する言語 L_0 ではなく、言語 $L_1 (\neq L_0)$ を直接機械翻訳する。サンプルアプリでは、L_0 を日本語、L_1 を英語とした複数の機械翻訳[*7] [*8] を利用した。この理由は、生成される文章の種類を増加させるためである。機械翻訳の例を次に示す。例からも分かるが、口語的な表現は機械翻訳がより難しいと考えられるため、Twitter などを SD として利用することで、より不自然な文章の生成を期待できる。

例: Tell me, if you will, a little of your background.
人間による翻訳の例
- よろしければ、ご経歴を教えて下さい。

機械による翻訳の例
- 少し背景で、あなたがそうするならば、私に言ってください。
- あなたがそうするならば、私にあなたの背景の少しを話してください。
- する場合は、少し背景のことを、私に伝えてください。

この例は、別な問題点も示唆している。機械翻訳では「background」 → 「背景」のように、使用される翻訳語の多様性が少ない傾向にある。この特徴を狙った攻撃を回避するためには、シソーラスを利用した単語の置換が有効である。[*9]

4. 議論

提案方式である言語的作問技法を用いるにあたり、検討が必要な内容を取り上げる。これらについては、今後の実験を通して得られる知見と合わせた考察が重要である。

作問で表示する文字数 $\ell(\kappa)$.

文意文脈解釈問題は、図 1 のような方式に比べて、表示される文字数が多くなる。一度に多くの文字を認識することは、問題の難易度が向上するだけでなく、精神的負担にもなる。特に、音声や点字ディスプレイの利用者は、この影響を受けやすい。表示文字数が少ないほど人間に対する負担が少なくなるが、問題文に含まれる情報量が落ちてしまう。よって、問題が簡単になったり、人間にも解けない程に意味を失う場合がある。点字ディスプレイの 40 から 80 という一行あたりの表示文字数は、利用者利便性の視点からの目安になると考えられる。

作問で表示する選択肢数 κ.

[*7] Weblio 翻訳, http://translate.weblio.jp/
[*8] エキサイト翻訳, http://www.excite.co.jp/world/
[*9] 本稿執筆時点では、サンプルアプリへ対してこの機能を実装していない。

選択肢数は、総当り攻撃への耐性に影響する。攻撃者が AI 問題に n 回挑戦し k 回以上正解する確率は、次のようになる。

$$\mathrm{Succ}_{\mathcal{P}^*}(\kappa, n, k) := \sum_{i=k}^{n} \binom{n}{i} \frac{1}{\kappa}^{i} \left(1 - \frac{1}{\kappa}\right)^{n-i}$$

$\mathrm{Succ}_{\mathcal{P}^*}(\kappa, n, k) > \epsilon_{\mathcal{P}^*}(\kappa, n, k)$ ならば、システム要求仕様を満たさない。それだけでなく $\mathrm{Succ}_{\mathcal{P}^*}(\kappa, n, k) > \epsilon_{\mathcal{P}}(\kappa, n, k)$ ならば、人間と誤認してしまう。

Q_1, Q_2 のような文意文脈問題では、選択肢の増加は問題文の文字数増加につながるため、図 1 のような方式に比べて、繰り返し数を多くする必要がある。

AI 問題の難易度.

本研究では、AI 問題の困難性に関する仮定を成立させるため、効率的なアルゴリズムをもつ攻撃者に対抗する処理を組み込んでいる。これらの効果を定量的に評価し、処理の安全性に影響するパラメータの値を決める必要がある。例として、対象文字列に適用する子音交替の頻度、WS の生成に用いる係り受け構造の利用頻度がある。

5. まとめ

本稿では、セキュリティ技術の特定の知覚への依存の問題を取り上げ、代表例である人間／人工知能判別テストでの要件を論じ、解決例を提示した。文意文脈解釈問題とすることで知覚依存を解消し、時々刻々作り出されるネット上の文章データを作問の種にすることで問題の新規性を保ち強度を与えるものである。

今後の課題として、文意文脈解釈問題の難易度の人間と人工知能とでの差の詳しい検証を考えている。

付　　録

A.1 表記

$\mathcal{S}$ を集合とする。$s \xleftarrow{D} \mathcal{S}$ は、$\mathcal{S}$ から確率分布 D で要素 s を選択することを表す。特に $D = \$$ の場合は、一様ランダムな分布から要素 s を選択することを表す。a, b を変数、リテラル、数字のいずれかとし、c を変数とする。$c \leftarrow a \circ b$ は、a, b に対して演算 $\circ$ の結果を c に代入することを表す。$c \leftarrow a$ は、a を c に代入することを表す。$|A|$ は、A が集合であればその濃度を、A が文字列ならばその文字数を表す。

確率的多項式時間 を PPT(Probabilistic Polynomial-Time) と表す。$\mathcal{A}(x; w)$ は、x を入力、w を内部乱数とした確率的アルゴリズム $\mathcal{A}$ を表す。w については、特に必要のない場合には省略する。アルゴリズム $\mathcal{A}$ により y が出力されることを $y \leftarrow \mathcal{A}(\cdot;\cdot)$ と表す。高々 τ の実行時間により、問題 Q を ϵ 以上の確率で解くアルゴリズム $\mathcal{A}$ を (ϵ, τ)-$\mathcal{A}$ と表す。

$\mathcal{P}, \mathcal{P}^*$ を証明アルゴリズムとする。CAPTCHA 様テストにおいては、人間 は $\mathcal{P}$ に、攻撃者 (不正な PPT 的ロボット) は $\mathcal{P}^*$ に相当する。$\mathcal{V}$ を検証アルゴリズムとする。CAPTCHA 様テストにおいては、人間ロボット判別アルゴリズムに相当する。

A.2 CAPTCHA 様テストの定義

Ahn ら [18] が示した記法を利用するが、セキュリティパラメータ関連の取り扱いが若干異なることに注意されたい。κ を AI 問題のセキュリティィィパラメータとし、$\ell(\cdot)$ を多項式とする。本稿では言語的作問を扱うので、κ は回答の選択肢数などに、$\ell(\kappa)$ は作問で用いる文字数などに影響を与える。作問に使用する文字集合を C とする。κ と補助入力 aux を入力とした PPT な作問アルゴリズム $\mathcal{G}$ により生成される問題の空間を $\mathcal{T}_\kappa \subseteq C^{\ell(\kappa)}, \mathcal{T} := \{\mathcal{T}_k\}_{k \in \mathbb{N}}$ とし、その回答の空間を $\mathcal{S}_\kappa \subseteq [0, \kappa - 1], \mathcal{S} := \{\mathcal{S}_k\}_{k \in \mathbb{N}}$ と表す。$\mathcal{H} := \{H_k\}_{k \in \mathbb{N}}$ は、$H_k : \mathcal{T}_\kappa \to \mathcal{S}_\kappa$ となる写像とする。

AI 問題は $Q = (\mathcal{G}, \mathcal{H})$ で定義される。Q は、$H_k(z) = a$ となる問題と回答のペア $(z, a) \in \mathcal{T}_\kappa \times \mathcal{S}_\kappa$ を生成する。アルゴリズム $\mathcal{A}$ が、AI 問題 Q を解く確率を次のように定義する。

$$\mathrm{Adv}^{Q}_{\mathcal{A}}(\kappa) := \Pr\left[\begin{array}{l}(z, a) \leftarrow \mathcal{G}_D(\kappa, aux; w_0); \\ a' \leftarrow \mathcal{A}(\kappa, z; w_1); a = a';\end{array}\right]$$

$(\mathrm{Adv}^{Q}_{\mathcal{P}^*}, \tau)$-$\mathcal{P}^*$ であるいかなる攻撃者も存在しない場合は、Q を $(\mathrm{Adv}^{Q}_{\mathcal{P}^*}, \tau)$-困難な AI 問題とよぶ。

システムの対象となる人間 $\mathcal{P}$ が、高々 $\tau_{\mathcal{P}}$ の時間と少なくとも $\epsilon_{\mathcal{P}}$ の確率で、$(\epsilon_{\mathcal{P}^*}, \tau_{\mathcal{P}^*})$-困難な AI 問題 Q をを解くことを $(\epsilon_{\mathcal{P}}, \tau_{\mathcal{P}}, \epsilon_{\mathcal{P}^*}, \tau_{\mathcal{P}^*})$-AI 仮定と表す。$(\epsilon_{\mathcal{P}}, \tau_{\mathcal{P}}, \epsilon_{\mathcal{P}^*}, \tau_{\mathcal{P}^*})$-AI 仮定において CAPTCHA 様テストが成立するには、次の条件を満たす必要がある。ただし、τ_{sys} はシステムが許容する作問から回答までの遅延時間の上限値、$\theta_{\mathcal{P}}$ はシステムが許容する $\mathcal{P}$ による正答確率の下限値、$\theta_{\mathcal{P}^*}$ はシステムが許容する $\mathcal{P}^*$ による攻撃成功確率の上限値とする。

$$\begin{array}{ll}\tau_{\mathcal{P}}(\kappa) \le \tau_{sys}(\kappa) & \\ \epsilon_{\mathcal{P}}(\kappa) \ge \theta_{\mathcal{P}}(\kappa) & \\ \epsilon_{\mathcal{P}}(\kappa) \gg \epsilon_{\mathcal{P}^*}(\kappa) & \text{if } \tau_{\mathcal{P}^*}(\kappa) \le \tau_{sys}(\kappa). \\ \epsilon_{\mathcal{P}^*}(\kappa) \le \theta_{\mathcal{P}^*}(\kappa) & \text{if } \tau_{\mathcal{P}^*}(\kappa) \le \tau_{sys}(\kappa).\end{array} \tag{A.1}$$

注意.

- AI 問題を解く確率が人間とロボットの間で乖離がある場合、AI 問題を繰り返し解くことでその乖離を大きくできる。これを利用した CAPTCHA 様テストは、n 回の出題のに対し、 k 回以上の正答によって人間と認証をする。
- Ahn ら [18] は、(n, k) を CAPTCHA のセキュリティパラメータとみなしている。本稿では、CAPTCHA 様テストとしてのセキュリティパラメータを (κ, n, k) と

する。繰り返しを考慮して ϵ, θ, τ を表記する場合は、$\epsilon(\kappa, n, k)$, $\theta(\kappa, n, k)$, $\tau(\kappa, n)(:= \tau(\kappa, 1) \times n)$ とする。

- 人間が AI 問題を解く確率は、その言語や知識などにより変化する。本稿では、標準的な日本人の成人を対象とする。
- 攻撃者 $\mathcal{P}^*$ は、内部乱数 w_0 を除き、アルゴリズム $\mathcal{G}$ に関する知識があるとする。

最後に、本稿で用いる CAPTCHA 様テストの定義を示す。

定義 1. *CAPTCHA* 様テスト. *AI* 問題 Q のセキュリティパラメータを κ、*CAPTCHA* 様テストシステムが一回の認証で繰り返す出題数を n、人間と認証する正答数のしきい値を k とする。人間の正答率の下限値 $\theta_{\mathcal{P}}(\kappa, n, k)$、ロボットの攻撃成功率の上限値 $\theta_{\mathcal{P}^*}(\kappa, n, k)$、認証処理を許可する時間の上限値 $\tau_{sys}(\kappa, n)$ に対して式 *(A.1)* の関係が成立するならば、$(\epsilon_{\mathcal{P}}(\kappa, n, k), \tau_{\mathcal{P}}(\kappa, n), \epsilon_{\mathcal{P}^*}(\kappa, n, k), \tau_{\mathcal{P}^*}(\kappa, n))$-*AI* 仮定のもとでの $(\theta_{\mathcal{P}}(\kappa, n, k),\ \theta_{\mathcal{P}^*}(\kappa, n, k),\ \tau_{sys}(\kappa, n))$-*CAPTCHA* 様テストと称す。 ■

参考文献

[1] Jeffrey P. Bigham and Anna C. Cavender. Evaluating existing audio captchas and an interface optimized for non-visual use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1829–1838. ACM, 2009.

[2] Elie Bursztein, Steven Bethard, Celine Fabry, John C. Mitchell, and Dan Jurafsky. How good are humans at solving captchas? a large scale evaluation. In *Proceedings of the 2010 IEEE Symposium on Security and Privacy*, SP '10, pages 399–413. IEEE Computer Society, 2010.

[3] Philip K. Dick. Voigt-kampff test. In *Do Androids Dream of Electric Sheep?* 1968.

[4] Jeremy Elson, John R. Douceur, Jon Howell, and Jared Saul. Asirra: a captcha that exploits interest-aligned manual image categorization. In *Proceedings of the 14th ACM conference on Computer and communications security*, CCS '07, pages 366–374. ACM, 2007.

[5] Frank Clay Fisk, Sri Ramanathan, Matthew Adam Terry, and Matthew Bunkley Trevathan. Advanced captcha using images in sequence. 2013.

[6] Robert M. French. Moving beyond the turing test. *Commun. ACM*, 55(12):74–77, 2012.

[7] Philippe Golle. Machine learning attacks against the asirra captcha. In *Proceedings of the 15th ACM conference on Computer and communications security*, CCS '08, pages 535–542. ACM, 2008.

[8] Jonathan Holman, Jonathan Lazar, Jinjuan Heidi Feng, and John D'Arcy. Developing usable captchas for blind users. In *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, Assets '07, pages 245–246. ACM, 2007.

[9] Taku Kudo. Mecab : Yet another part-of-speech and morphological analyzer. *http://mecab.sourceforge.net/*.

[10] Jonathan Lazar, Jinjuan Feng, Tim Brooks, Genna Melamed, Brian Wentz, Jon Holman, Abiodun Olalere, and Nnanna Ekedebe. The soundsright captcha: an improved approach to audio human interaction proofs for blind users. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, CHI '12, pages 2267–2276. ACM, 2012.

[11] Christopher Liam. System and method for delivering a human interactive proof to the visually impaired by means of semantic association of objects, 2012.

[12] Mir Tafseer Nayeem, Md. Saddam Hossain Mukta, Samsuddin Ahmed, and Md. Mahbubur Rahman. *2012 IEEE 15th International Conference on Computational Science and Engineering*, 0:178–185, 2012.

[13] G.E. Rawlinson. *The Significance of Letter Position in Word Recognition*. University of Nottingham, 1976.

[14] K. Saberi and DR Perrott. Cognitive restoration of reversed speech. *Nature*, 398(6730):760, 1999.

[15] Yuji Matsumoto Taku Kudo. Japanese dependency analysis using cascaded chunking. In *Proceedings of the 6th Conference on Natural Language Learning*, pages 63–69, 2002.

[16] Jennifer Tam, Jiri Simsa, Sean Hyde, and Luis von Ahn. Breaking audio captchas. In *Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems*, Advances in Neural Information Processing Systems 21, pages 1625–1632. MIT Press, 2008.

[17] Luis von Ahn, Manuel Blum, Nicholas Hopper, and John Langford. The official captcha site.

[18] Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford. Captcha: Using hard ai problems for security. In *In Proceedings of EUROCRYPT*, volume 2656 of *LNCS*, pages 294–311. Springer-Verlag, 2003.

[19] H. Wimmer and J. Perner. Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children's understanding of deception. *Cognition*, 13(1):103, 1983.

[20] Pablo Ximenes, Andre Santos, Marcial Fernandez, and Jr. Celestino, Joaquim. A captcha in the text domain. In *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, volume 4277 of *LNCS*, pages 605–615. Springer-Verlag, 2006.

[21] Jeff Yan, Ahmad Salah, and El Ahmad. Breaking visual captchas with naive pattern recognition algorithms, 2007.

[22] 森本 浩介, 片瀬 弘晶, and 山名 早人. N-gram と離散型共起表現を用いたワードサラダ型スパム検出手法の提案. **情報処理学会研究報告. データベース・システム研究会報告**, 148:1–8, jul 2009.

[23] 山本 匠, J.D. Tygar, and 西垣 正勝. 機械翻訳の違和感を用いた captcha の提案. **情報処理学会研究報告**. *CSEC, [コンピュータセキュリティ]*, 2009(37):1–8, 2009.

[24] 上原 章敬, and 匠 山本 徳一郎, 鈴木, and 西垣 正勝. 4 コマ漫画 captcha の検討. **情報処理学会研究報告**. *CSEC, [コンピュータセキュリティ]*, 2011(13):1–8, 2011.

[25] 福岡 千尋, 西本 卓也, and 渡辺 隆行. 音韻修復効果を用いた音声 captcha の検討 (高齢者の認知機能保障技術及び一般). **電子情報通信学会技術研究報告**. *WIT*, **福祉情報工学**, 108(332):83–88, 2008.

[26] 西本 卓也, 西亀 健太, and 嵯峨山 茂樹. 音声 captcha のための音韻修復効果の検討. **聴覚研究会資料**, 38(6):639–644, 2008.

[27] 西本 卓也, 松村 瞳, and 渡辺 隆行. 音声 captcha システムにおける削除法と混合法の比較 (福祉と音声処理, 一般). **電子情報通信学会技術研究報告**. *WIT*, **福祉情報工学**, 109(260):55–60, 2009.

[28] 岡崎 直観 and 辻井 潤一. 高速な類似文字列検索アルゴリズム. In **情報処理学会創立 *50* 周年記念全国大会**, pages 1C–1, 2010.

[29] 鴨志田 芳典 and 菊池 浩明. 文章合成の不自然さの評価と応用. **ファジィシステムシンポジウム講演論文集**, 26:1069–1074, 2010.