

視覚障害者学習支援のための MathML 変換

渡辺千晶^{†1} 浅本紀子^{†2} 桑名杏奈^{†2}

概要: 現在、日本には 393 万 7 千人もの多くの身体障害者がいる。しかし求職中の身体障害者数と実際の被雇用者数には不釣り合いが生じている。身体障害者の就労をもっと増やすためには、彼らの享受する教育をより良いものにする必要がある。身体障害者の学習には、障害の特性に応じた機能を使用できる教材があれば良いと考える。本研究ではオンライン学習は身体障害者の中でも視覚障害者に対して特に有効であると考え、視覚障害者の学習を保障するアプリケーションを作成したいと考えた。そこで視覚障害者のスクリーンリーダーを利用した数式のオンライン学習を支援するため、数式の TeX コードをコンテンツ形式 MathML に変換するアプリケーションを考案した。

キーワード: 視覚障害者, MathML

1. はじめに

現在、日本には 393 万 7 千人もの多くの身体障害者がいる [1]。また年齢階層ごとの身体障害者の割合に着目してみると、高齢になる程割合が高い。加えて日本では高齢化が進行していることから、身体障害者は今後も増加していくと考えられる。また近年、公共職業安定所を通じた身体障害者の新規求職件数、就職件数が共に上昇していることから、身体障害者の就労意識は高まってきているといえる。そして、2012 年に障害者の雇用率を 1.8%だったところを 2%に引き上げることが閣議決定され、2013 年から実施されている。しかし現在の障害者の民間企業の実雇用率は 2% 達成にはほど遠い。このように、求職中の身体障害者数と実際の被雇用者数には不釣り合いが生じている。身体障害者の就労を増やすためには、彼らの享受する教育をより良いものにする必要がある。身体障害者の学習には、障害の特性に応じた機能を使用できる教材が効果的だと考える。コンピュータを用いた学習は、時間や場所を選ばず自由に学習できる環境を提供してくれる。本研究では視覚障害者のオンラインでの学習を補償するアプリケーションを作成したいと考えた。

2. 背景

2.1 歴史 [7]

視覚障害者への情報提供手段の工夫は古くから試みられており、当初は粘土板や木板に刻んだ文字や紙を切り抜いた文字等が用いられていた [3]。しかし視覚障害者にとってそれらは読むのに遅く、書くのには非常に困難であった。そこで 1825 年にルイ・ブライユが視覚障害者の文字の読み書きを効率的なものにするために、アルファベットを 6 つの点で表すという点字を考案した。その後点字が社会に

浸透して教育と福祉が発展するようになる [4]、1930 年代以降に朗読サービスの始まりがみられるようになった。これはボランティア等による対面による書物や文書の朗読と、音響機器を用いた録音テープ等のメディア配布を指す。

点字を利用するためには点字の学習をする必要がある。また点字を図書にするとかさばるのでそれを運ぶのに移動が必要となり、視覚障害者にとって移動に伴う障害は、それが僅かであっても命の危険に繋がる恐れがある。このように点字を利用するには情報入手や移動のために第三者の手助けが必要となる。実際に、視覚障害者の中で点字を利用している者の割合は 10%程度と非常に少ない [5]。朗読サービスも、ボランティア等に読んでもらうものであるために第三者の手助けが必要となる。よっていずれも人的サービスのものであり、時間や場所が制限されるため限界がある。

そこでワードプロセッサとスクリーンリーダーの開発により、現在はコンピュータ文書をスクリーンリーダーに読み上げさせて情報を入力する手法がトレンドとなっている。国立特別支援教育総合研究所が行った調査によると、調査対象者の 94.7%がパソコンを普段利用しており、そのうちの 84.4%がスクリーンリーダーを利用している [6]。この手法により、視覚障害者がある場で自力で学習することが可能となっている。

コンピュータ文書やスクリーンリーダーにも課題は残っている。例えば数式や図、表などの読み上げには、特殊な環境が必要であったり、正確に読み上げられなかったりする。十分な学習環境に行き届いているとは言い難い。現状として、大学(大学院含む)全体での専攻科別の学生在籍率について、障害の有無を考慮しない場合、社会科学、工学、人文科学が上位 3 専攻を占める。しかし障害学生においては、人文科学に学生が集中している [7]。これは、人文科学系以外の学問領域では数式や図形といった情報へのアクセスが必要であるためと推察される。

ここで、数式や図、表などを多く含む科学技術文書の歴史に目を向けてみる。タイプライターや手書きで書かれていた時代に比べて、ワードプロセッサが開発されてからは

^{†1} お茶の水女子大学人間文化創成科学研究科
Ochanomizu University, Graduate Courses, Humanities and Sciences,
Advanced Sciences, Computer Science
^{†2} お茶の水女子大学
Ochanomizu University
^{†3} お茶の水女子大学
Ochanomizu University

3. MathML コンテンツ形式への変換の提案

3.1 MathML プレゼンテーション形式の問題

MathML のプレゼンテーション形式で記述された数式の中には、ブラウザ上での表示はコンテンツ形式のものと同じであっても、コンピュータ上では本来とは異なった意味で認識されてしまうものがある。例えば、

$$\frac{dy}{dx}$$

の数式については、これがコンテンツ形式で記述されたものだと微分記号として認識されるが、プレゼンテーション形式で記述されたものだと分数として認識されてしまう。実際にこの例をスクリーンリーダーで読み上げさせた場合、「dx 分の dy」というように分数として読み上げられる。「微分に関する文書を読んでいたのに、いきなり分数が出てきておかしい」などとユーザーが違和感を覚えれば、確かに本来の意味とは異なるが形は同じなので、考えれば微分記号のことだろうと判断することができる。しかし、微分の初学者など違和感を覚えることができなければ、スクリーンリーダーに読み上げられた「分数」のまま文書の解釈を進めてしまう可能性もある。

ここでソースコードが数式と似てもおらず意味が全く異なる例を次に挙げる。

$${}_n C_r = \binom{n}{r}$$

上記の式の右辺について、ブラウザ上では同じ表示でも、コードの書き方が二通りある。一つは本来の二項係数を表示させる書き方 (TeX では、`\binom{n}{r}`) だが、二つは括線の太さを 0 にした $\frac{n}{r}$ を大括弧でくくった書き方 (`\left(\genfrac{}{}{0pt}{}{n}{r}\right)`) であり、 $\frac{n}{r}$ の分数と認識される可能性がある。後者をスクリーンリーダーで読み上げさせた場合、「括弧 r 分の n」というように分数として読み上げられる。

3.2 提案

そこで本研究では、数式の意味を厳密に表現するコンテンツ形式の MathML を、既存の文書を変換して生成するアプリケーションを提案する。変換元としては、プレゼンテーション形式 MathML より普及しており、またプレゼンテーション形式 MathML よりソースコードが短い、TeX のソースコードを採用する。

また TeX で簡易的に記述された数式を数式処理システムが正しく読み込めない際に、変換後の数式をシステムの入力に受け渡すことができるようにすることも本研究の目標のひとつとする。

4. 提案手法

本研究は大きく分けて 2 つの処理段階で構成される。

- 数式の意味推定：前後の文脈に基づいた数式の意味

推定

- 変換：MathML コンテンツ形式への変換
各処理段階の詳細について以下に記述する。

4.1 数式の意味推定

複数の意味を持つ数式をコンテンツ形式 MathML に変換するにあたって、その数式が文書においてどのような意味で使われているかを区別する必要がある。その数式の意味は前後の文脈からある程度推定することが可能である。例えば、

$$\frac{dy}{dx}$$

の数式については、微分の記述である場合が多いが、分数とも解釈し得る。この数式が、「微分」、「グラフ」、「傾き」などといった単語の近くに出現すれば微分である可能性が高い。また「変数 d」、「約分」などといった単語の近くに出現すれば分数である可能性が高い。

4.1.1 意味の自動推定

この推定をコンピュータで自動化するにあたり、全ての可能性を列挙することは現実的には難しい。そこで、テキストマイニングに用いられる機械学習の手法を用いて自動的に数式の意味を推定できるよう、学習器を生成する。この手法により、未知語を含む文書における数式の意味推定も可能になる。

4.1.2 学習器の生成

ここでやりたいことは、例えば「この数字 1.557 と、先程計算した $\frac{dy}{dx}$ の値とを比較する」等の TeX 文章があったとき、

文章中の「 $\frac{dy}{dx}$ 」が分数なのか微分なのかを自動判定してくれる学習器を作ることである。

学習器の生成には、テキストマイニングの分野で一般的に用いられている手法や機能、プログラム等を用いた。

まず文章を形態素 (名詞・動詞等の言葉の最小単位) に分割する必要がある。これにはオープンソースの形態素解析ツール MeCab を用いた。一般的な MeCab には TeX の数式を判別する機能はないため、MeCab の辞書に TeX コードと記号を新しい単語として追加した。この MeCab を用いると、図 3 のように数式が分割される。

```

\frac 記号,texコード,*,*,*,*,*,*
{ 記号,texコード,*,*,*,*,*,*
- 記号,texコード,*,*,*,*,*,*
b 名詞,一般,*,*,*,*,*
+ 記号,texコード,*,*,*,*,*,*
\sqrt 記号,texコード,*,*,*,*,*,*
{ 記号,texコード,*,*,*,*,*,*
b 名詞,一般,*,*,*,*,*
^ 記号,texコード,*,*,*,*,*,*
2 名詞,数,*,*,*,*,*
- 記号,texコード,*,*,*,*,*,*
4 名詞,数,*,*,*,*,*
a 名詞,一般,*,*,*,*,*
c 名詞,一般,*,*,*,*,*
} 記号,texコード,*,*,*,*,*,*
} 記号,texコード,*,*,*,*,*,*
{ 記号,texコード,*,*,*,*,*,*
2 名詞,数,*,*,*,*,*
a 名詞,一般,*,*,*,*,*
} 記号,texコード,*,*,*,*,*,*
EOS
    
```

図 3 MeCab による形態素解析

続いて、自然言語を含む文書をコンピュータが扱えるような形にするために、文書を素性ベクトル化する。これには、python のライブラリである sklearn の関数 CountVectorizer を用いた。

その後、ベクトル化した文書一つ一つに文書の内容に基づいてラベル付けを行う。

こうしてベクトル化した文書とラベルの組を一つの学習データとする。この学習データを数多く用意するほど、データマイニングの精度が良くなる。

本研究で与える学習データは以下のようなイメージである。

[学習データ 1]

文書：分数 $\frac{dy}{dx}$ を約分する

ベクトル化：[..., 1, 1, 0, 0, 0, 0, ...]

ラベル：分数

[学習データ 2]

文書：関数の傾き $\frac{dy}{dx}$ を計算する

ベクトル化：[..., 0, 0, 1, 1, 1, 0, 0, ...]

ラベル：微分

[学習データ 3]

文書：変数 y を x で微分した $\frac{dy}{dx}$

ベクトル化：[..., 0, 0, 0, 0, 0, 1, 1, ...]

ラベル：微分

[学習データ 4]

文書：変数 d を用いて $\frac{dy}{dx}$ と表せる

ベクトル化：[..., 0, 0, 0, 0, 0, 1, 0, ...]

ラベル：分数

こうして用意した学習データのパターン認識を行う。学習データのグループ分けをすることで、その後未知のデータが与えられた際にどのグループに帰属するかを判断することができる。これには、パターン認識モデルとしてサポートベクターマシンを用いる。その際に利用するコストパラメータとカーネルパラメータそれぞれの丁度良い値の組み合わせの探索を行うため、グリッドサーチを用いる。その度にクロスバリデーションにより検証を行う。

4.2 MathML コンテンツ形式への変換

機械的に変換できるものは機械的に、複数の意味を持つ数式については 4.1 で推定した意味に基づいて TeX 文書内の数式部分を、MathML コンテンツ形式に変換する。LaTeX 文書を MathML プレゼンテーション形式に変換するツール (TtM) が公開されているので、それを利用する。

二次方程式の解の公式をコンテンツ形式に変換した結果のイメージ図をあげる(図 4)。

```

<mrow>
<apply>
<eq/>
<ci>x</ci>
<apply>
<divide/>
<apply>
<mo>&PlusMinus;</mo>
<apply>
<minus/>
<ci>b</ci>
</apply>
<apply>
<root/>
<apply>
<minus/>
<apply>
<power/>
<ci>b</ci>
<cn>2</cn>
</apply>
<apply>
<times/>
<cn>4</cn>
<ci>a</ci>
<ci>c</ci>
</apply>
</apply>
</apply>
<cn>2</cn>
</apply>
</apply>
<times/>
<cn>2</cn>
<ci>a</ci>
</apply>
</apply>
</mrow>
    
```

表記: $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

図 4 コンテンツ形式の二次方程式の解の公式

5. 利活用

5.1 スクリーンリーダーでの利用

本研究で変換したコンテンツ形式の MathML コードを XML 形式で保存し、ブラウザに表示させてスクリーンリーダーに読み上げさせれば、数式の本来的意味に従った正しい読み上げが可能になる。特に 3.1 で挙げたような複数

の意味を持つ数式による混乱を防ぐことができる。

5.2 他システムとの連携

本研究で変換したコンテンツ形式の MathML は論理的構造を持っている。そのため、他システムへの利用が容易である。例えば以下のような利活用が期待できる。

数式処理システム

数式処理システムとは、数学をコンピュータで扱うソフトウェアであり、数式を近似値ではなく数式のままで処理できる特徴がある。よく知られているものに、Mathematica や Maxima などがある。

Mathematica は、Wolfram Research 社が提供する数値計算のアプリケーションである。Mathematica コードで記述した数式を入力情報とし、その計算結果が出力される。

Wolfram 言語では XML がサポートされているため、Mathematica の入力時に本研究を利用することで、TeX 形式の文書を利用した Mathematica への入力が可能になる。そうすれば Mathematica に数式を一から打ち込む手間が省くことができ、入力が楽になる。

数学オンライン評価システム

e-Learning システムには、数式を含む設問に対応した数学オンライン評価システムがある。例として STACK 等がある。

STACK とは、ある数式の設問に対してユーザーが返した回答の正誤評価を行う e-Learning システムである。2004 年にバーミンガム大学の Sangwin らによって開発が始められ、2008 年から Moodle と連携している。また、ポテンシャル・レスポンス・ツリーという機構を用いて、ユーザーの回答に対するコメントを与えることも可能で、教育効果が高まることに繋がると期待されている。

STACK では数式処理にオープンソースソフトウェアである Maxima が採用されており、Maxima の書式に従った数式を入力情報とするので、複雑な数式の入力が大変である。

本研究の MathML 変換を利用することで、これまで TeX で蓄積されている多くの問題を STACK へ取り込むことができる。ひいては採点を重ね不正解だった問題をたくさん集めれば、それらに共通する学習範囲を提示することで、復習のフィードバック提供が可能になる。

6. まとめ

視覚障害者のスクリーンリーダーを利用した数式のオンライン学習を支援するため、数式の TeX コードをコンテンツ形式の MathML コードに変換するアプリケーションの構築を目指している。現在数式の前後の文脈によってその数式がどのような意味をもつのか自動で判断を行えるようにしている。このようにして変換の精度を高めつつ、スクリーンリーダー以外にも、自動採点システム等の他システムと

の連携を含め、本研究の幅広い利活用について考えていきたい。

参考文献

- [1] 内閣府「平成 26 年版障害者白書（概要）」第 3 章障害者の状況(基本的統計より)<<http://www8.cao.go.jp/shougai/whitepaper/h26hakusho/gaiyoyu/h03.html>>(2017/2/8 閲覧)
- [2] 大武信之, 視聴覚障害者高等教育支援のための文書処理システムの開発, お茶の水女子大学大学院 人間文化研究科博士(理学) 学位論文 ; 博乙第 174 号(2002)
- [3] 東京ヘレン・ケラー協会「点字を学ぶ-点字の歴史-」<<http://www.thka.jp/tenji/braille.html>>(2017/2/8 閲覧)
- [4] 社会福祉法人「点字図書館-録音図書について-」<<http://www.nittento.or.jp/about/scene/recording.html>>(2017/2/8 閲覧)
- [5] 厚生労働省「平成 18 年度身体障害児・者等実態調査」
- [6] 国立特別支援教育総合研究所「視覚障害者のパソコン・インターネット・携帯電話利用状況調査」<http://www.nise.go.jp/kenshuka/josa/kankobutsu/pub_d/d-267/d-267_5.pdf>
- [7] 日本学習支援機構「障害学生受入促進研究委託事業一障害のある生徒の進学の促進・支援のための高大連携の在り方に関する調査研究—」第 1 章障害学生の大学進学の実況<http://www.jasso.go.jp/gakusei/tokubetsu_shien/chosa_kenkyu/ko udarairei/_icsFiles/fieldfile/2015/11/19/jasso2211.pdf>