辞書データ・Webデータを利用したクラスタリングによる単語難易度の推定

中西 聖明[1]　　小林 伸行[2]　　椎名 広光[3]

[1]岡山理科大学大学院 総合情報研究科 情報科学専攻

[2]山陽学園大学 総合人間学部 生活心理学科

[3]岡山理科大学 総合情報学部 情報科学科

NakanishiOus@gmail.co.jp[1], koba_nob@sguc.ac.jp[2], shiina@mis.ous.ac.jp[3]

1　はじめに

本研究では, 日本語単語の難易度を辞書とWebデータの記述から推定することを試みている．背景としては, 日本語を母語としない留学生などの外国語として学習する際に単語の難易度が提示できると, 学習者の語学学習に役立つほかに教師側にも問題作成やe-learningのシステム開発に必要と考えるからである．しかしながら，日本語に関しては漢字の難易度の指標はよく知られているが, 単語の難易度はあまり知られていない. 単語を構成する文字の難易度が単語の難易度と考えられるが, 日本語の学習者に提示するべきものとは違いが大きい．そして, 単語の難易度は, 日本語を母語とする学習者と日本語を母語としない学習者では, とらえられ方に相違があるのではないかと考えられる．加えて日本語の利用範囲や記述方法によっても単語の難易度のとらえられ方に相違点があるのではないかと考えられる．そこで本研究では，言語学習の基礎的なデータ収集として, 既知の日本単語の難易度の初期データから辞書のすべて見出し語のそれぞれで辞書とブログの記述を利用して難易度を推定する方法を提案する.

本提案手法では, 単語難易度を推定したい単語の周辺では同等な難易度の単語が使用されるという仮説を想定している．日本語単語の難易度の指標には, 日本語非母語者用の日本語能力試験 (JLPT)[1] の旧試験で利用されている4段階の難易度を利用する．日本語単語の難易度の推定手法としては, 機械学習を利用したクラスタリングを用いる．クラスタリングの学習アルゴリズムとしては, 教師付き学習による分類器である多クラスサポートベクタマシン (以下,SVM)[2], 難易度のグレード別の多峰性正規分布のマハラノビス距離によるクラスタリング (以下,MNDC) の場合, そしてこれらの3つを混合した手法の3種類の方法を使って推定を行う.

判別に利用する学習パラメータの素性としては, 辞書の意味，ブログの記述中の単語と推定単語との関係から作成したものを利用している．また，学習パラメータの素性を作成前に辞書データやブログデータの定型文の削除や選択のフィルタリングを行い，精度の向上も図っている.

2　難易度推定と処理概要

単語の難易度推定の処理概要 (図1) を以下に示す.

Step1:単語の難易度が判明している初期データ [3] の辞書またはブログの記述文を取得する.

Step2:単語に当てはめた難易度から学習パラメータを生成し, 見出し語の難易度を教師信号として分類器を学習する.

Step3:初期データにない見出し語は, 辞書またはブログの記述から学習パラメータを作り, 分類器を用いて難易度を推定する.

Step4:辞書の見出し語の難易度を, 辞書またはブログの記述文に当てはめる.

Step5:ブログの記述文から学習パラメータを再生成し, 分類器を再学習する.

初期データにない見出し語の難易度は, 意味記述から作られる学習パラメータが何度か再学習が行われて生成される．再学習を行わない単語”歴史”の例を図2(a)，再学習を行う単語”機会”の例を図2(b) に示す. 単語“機会”の例では, 意味記述中の単語“建造”の難易度が推定されてからもう一度学習パラメータを再生成し, 学習を行っている.

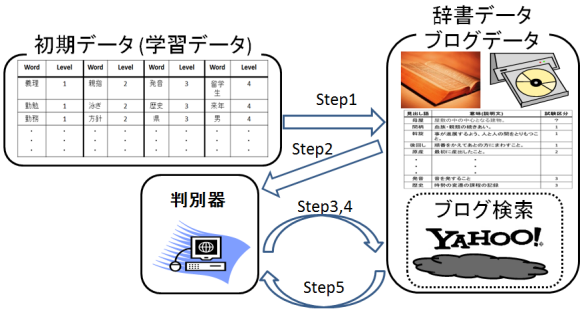

図 1: 判別器の学習過程の概要

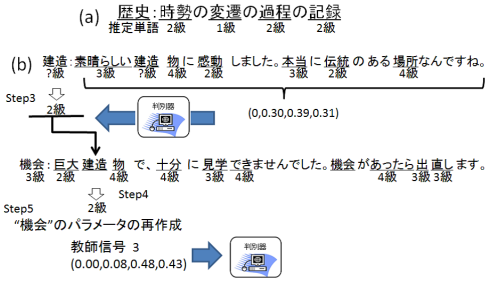

図 2: 学習パラメータとパラメータの再生成

3 学習パラメータの生成

本研究では辞書データ, ブログデータを利用する推定のそれぞれに対して学習パラメータを作成する. 以下に学習パラメータの作成方法を示す.

3.1 辞書データに対する学習パラメータの生成

辞書データを用いて推定を行う場合, 辞書の意味記述中の単語難易度分布を利用する. 本研究ではある見出し語単語と対応する意味記述との関係として, 見出し語と同一文中で利用されている単語の難易度は, 近い難易度になっているという仮説を想定している. この仮説のもとで, 辞書記述に現れる単語のグレード別難易度分布を, その見出し語の単語の難易度に相関があるとして, 学習パラメータとしている. これについて以下に示す.

難易度が L 区分あるとしたときに, 見出し語 w の意味記述中に現れる単語の難易度グレードごとの頻度を $D(l,w), l=1,\cdots,L$ とすると, 正規化した難易度頻度分布 $DR(l,w)$ は, 次の式で定義する.

$$DR(l,w)=\frac{D(l,w)}{\sum_{i=1\ldots L}D(i,w)}$$

学習パラメータは, 単語 w の難易度と意味記述中の難易度分布の L 個からなる (w の難易度, $DR(1,w)$, $DR(2,w)$, ..., $DR(L,w)$) としている.

図 2(a) の日本語辞書の見出し語「歴史」の例では, 意味記述は「時勢の変遷の過程の記録」となっており, その時の単語難易度ごとの頻度から難易度分布 $DR(l,w)$ が求められる. この文中には 1 個の 1 級, 3 個の 2 級, 0 個の 3,4 級が意味記述中に現れ, 1 級から 4 級の難易度のグレード別の比率を並べて学習パラメータ $(\frac{1}{4},\frac{3}{4},\frac{0}{4},\frac{0}{4})$=(0.25, 0.5, 0.00, 0.00) とする.

3.2 ブログデータに対する学習パラメータの生成

ブログデータを用いて推定を行う場合, ブログ中の各難易度の単語の位置関係を利用する. ここではある見出し語単語と同一ブログ中の単語との関係として, 見出し語と近い距離の単語ほど, 近い難易度になっているという仮説を想定する. この仮説のもとで, 同一ブログ中に現れる単語の距離に応じたグレード別難易度分布を, その見出し語の単語の難易度に相関があるとして, 学習パラメータとしている. 学習パラメータの定義を以下に示す.

(1) ブログ内の文中に含まれる平均単語数を偏差 σ, 見出し語の出現する位置を中心とする正規分布を作成する.

(2) 距離は見出し語単語を基準とした各単語との間の単語数に 1 を加算したものとする.

(3)$GR(l,w)$ を w からの距離に応じた l 級単語の確率密度の和とする.

(4)$$E(l,x)=\begin{cases}1 & w\text{ から距離 }x\text{ の単語が }l\text{ 級}\\ 0 & \text{上記でない場合}\end{cases}$$

$$G(l,w)=\sum_{x=firstword}^{x=lastword}\frac{1}{\sqrt{\pi}\sigma}e^{-\frac{x^2}{2\sigma^2}}\cdot E(l,x).$$

(5) グレード別の比率で求めた

$$GR(l,w)=\sum_{i=1,2\ldots L}\frac{G(l,w)}{G(i,w)}$$

を学習パラメータとする.

図 3 の単語“毛布”の例では, 文中の平均単語数は 3.0, 2 級の単語が距離 1,2,2,3 に, 3 級の単語が距離 1 に, 4 級の単語が距離 3 の位置に現れるので, 各単語の確率密度の和をとり 1 級から 4 級のグレード別の比率を並べてた学習パラメータ (0.00, 0.67, 0.20, 0.13) と初期学習パラメーターとしている. また, 学習パラメータの再生成によって (0.08, 0.48, 0.43, 0.00) となり, 教師信号 3 とあわせて, 分類機に学習させている.

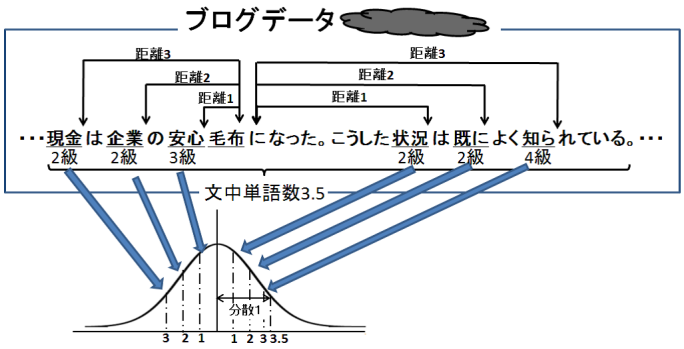

図 3: ブログパラメータ

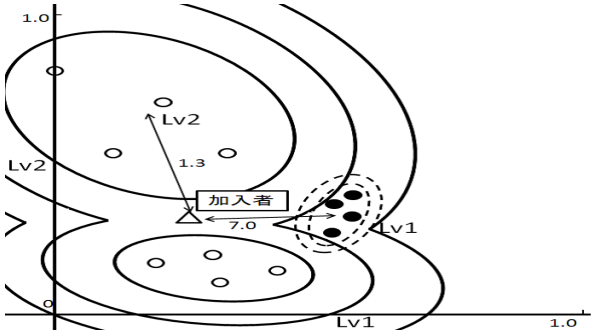

図 4: マハラノビス距離による境界面

表 1: 定型表現の出現回数と分散

定型表現	出現回数	平均からの分散
国際オリンピック委員会	3	0.3
大山脈.	1	—
の別称.	126	11.5
の転.	426	38.7
〔古い言い方で〕	224	20.4
メートル法で，面積を	2	0.1
より丁寧な言い方.	7	1.3

4 日本語単語の難易度推定

4.1 SVM を利用した推定

本研究では SVM の中でも多クラス SVM を利用し, 難易度のグレードを分類クラスに対応させ, 教師信号としている．本研究では難易度のグレード 4 段階をそれぞれクラスに分類している．

4.2 MNDC を利用した推定

難易度のグレード別の混合正規分布のマハラノビス距離によるクラスタリングを行う．教師データである初期データを難易度のグレード別に, 学習パラメータが出現する確率密度関数を混合正規分布で近似する．各難易度グレードの混合正規分布の中心からのマハラノビス距離が等しくなる境界を分離面とする．図 4 は, 学習パラメータの素性のうち 1 級, 2 級の素性の 2 つに削減して, 2 次元座標に 2 つのグレードの学習パラメータのプロットしたものである．●は 1 級, ○は 2 級である． 2 級は, 中心から広がっている． 1 級は中心に集中していて, マハラノビス距離が 2 の境界で分離している．単語 △“加入者”は, 1 級の中心から 7. 0，2 級単語の中心から 1． 3 のマハラノビス距離にあり，2 級と推定する．

4.3 SVM と MNDC の合成手法

混合手法では SVM と MNDC のクローズドテストの学習結果を用いて推定する．クローズドテストにおいて, 2 つの推定結果における各級の正しい級別の割合を確信度とし，最も確信度の高い級別として推定する．

5 データのフィルタリング

本研究では辞書データ，ブログデータのそれぞれに対してあらかじめフィルタリングを行っている. それぞれのフィルタリングについて以下に示す.

5.1 辞書データ中の定型文削除

辞書で用いられる表現は辞書特有の表現があるため, 難易度分布に偏りが出ることがある． そこでパターン化した表現を削除するフィルタリングを行っている. 以下にその手順を示す.

Step1:定型文の定義として品詞をもとに，意味記述の文頭，文末から 1 から 10 品詞でかつ辞書中に 2 回以上するものを定型文とする.

Step2:定型文の頻度に正規分布を近似させ，分散の上位 2 以上である定型文の削除を行い推定する.

本研究において定型表現とするものの例を表 1 に示す．表 1 においては「の別称。」,「の転。」,「[古い言い方で]」,「より丁寧な言い方」が平均からの分散が上位 2 以上であるため削除を行っている.

5.2 ブログ文のデータ選択

本研究ではでは各単語につき複数のブログ文を取得し単語難易度推定を行っている． しかしながら, ブロ

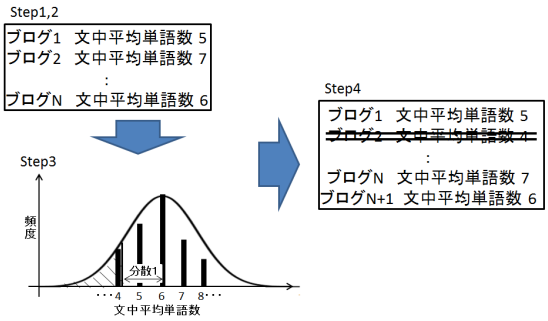

図 5: ブログ文のデータ選択

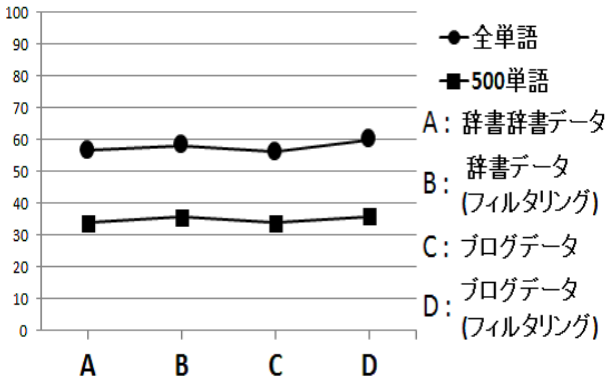

図 6: 各推定の正答率

グ文には極端には文中の単語数が少ない物が存在し, これらのブログ文は正確な日本語表現で書かれていないと考えられる． そこで, 本手法では極端に文中の単語数が少ないブログに関しては除去を行い, ブログの再取得を行う．

Step1:各単語につき N 個のブログ文を取得する．

Step2:Step1 で取得したブログ文中の平均単語数を求める．

Step3:それぞれのブログの文中平均単語数の頻度に対して正規分布で近似を行い，平均からの分散が下位 1 以下であったブログは除去する．

Step4:Step3 で除去したブログ数と同数のブログ文を取得する． ただし，Step4 の分散距離が下位 1 以上であるようにする．

初期データとブログの記述と分類器の学習の過程を図 5 に示す．また，本研究では N の値を求めるために 1 から 10 件のブログを利用した． ここで正答率が最も高かったのは各単語につき 9 つのブログを利用した場合であったため，以後 9 つのブログを利用し推定を行う．

5.3 評価

本研究において，学習データを対象とし，500 単語, 全単語を利用した 2 つのクローズドテストの推定精度を図 6 に示す．本研究において辞書，ブログ文を用いた双方の推定の正答率は同程度となっている．しかしながら，辞書を用いた場合には多くの単語が難解であると推定されているのに対し，ブログ文を用いた場合には若干易しく推定される傾向がある．また，辞書, ブログを用いた推定の双方において，フィルタリングを行うことで正答率の改善されている．フィルタリングによって辞書においては難解な定型表現が，ブログ文においては文中の単語数が少ない表現が削除された結果と考えられる．

6 おわりに

本研究では辞書データ，ブログデータを利用して日本語単語の難易度評価を行った． また，初期学習データとしては各言語の学習者とその言語を母国語とする人にとっての難易度を利用し，SVM，MNDC, 混合手法による単語の難易度評価を行った． 結果として日本語においては辞書の意味とブログ文の難易度評価の差異が小さくなった． 現状では使用する辞書の性質が難易度に大きく影響しており，このことが各評価間の差異に繋がった一因と考えられる． 単語評価においても初期学習データの対象年齢等に差異があるためこれらの差異を考慮しなければならない． 今後は難易度の精密化のために, 教師データの構築やデータ収集の様々な違いを調査する予定である．また, 日本語単語以外の英語や中国語についても拡大したいと考えている．

参考文献

[1] 日本語能力試験公式ウェブサイト, http://www.jlpt.jp

[2] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[3] 徳弘康代，日本語学習のためのよく使う順漢字 2100, 三省堂,2008.

[4] Yahoo Japan Developer Network, http://developer.yahoo.co.jp/