卒業論文

異言語間の分散表現を用いた文化差検出

指導教官　　村上 陽平 准教授

立命館大学　情報理工学部
先端社会デザインコース　4 回生
2600170082-0

大井 也史

2020 年度（秋学期）卒業研究 3（CH）
令和 3 年 2 月 1 日

異言語間の分散表現を用いた文化差検出

大井也史

内容梗概

近年，機械翻訳の品質が向上してきており，機械翻訳を用いた多言語コミュニケーションや異文化コラボレーションが可能になってきている．このようなコミュニケーションでは，翻訳は合っているが文化差によってコミュニケーション齟齬が生まれてしまう場合がある．例えば，「卵」は，日本で生で食べる文化があるが，外国にはない．日本人が外国人に卵かけご飯などの卵を生で食べる料理の説明をする場合，加熱処理をしているとは言っていなくて，外国人は卵に何らかの加熱処理をしていると解釈するだろう．このようなコミュニケーションの齟齬を解消するために，画像検索を用いた対訳ペア間の文化差検出がある．しかしながら，この手法は画像検索の結果に依存するため，主要な画像しか検索されず，文化差のある画像が取得できない場合がある．また，画像として表現しにくい抽象的な概念では文化差を検出できない．

そこで，本研究では単語の分散表現を用い，異言語間で単語の分散表現をマッピングして文化差を検出する手法を提案する．テキスト情報に基づく手法のため，抽象的な概念の文化差も検出可能である．具体的には，言語ごとの Wikipedia コーパスを用いて各言語の分散表現空間を作成し，両空間のアライメントを行う．次に，概念辞書を用いて，対象概念(Synset)に紐付けされている各言語の単語群のベクトルを取得し，言語ごとの対象概念の統合ベクトルを作成する．その後，統合ベクトルを用いて文化類似度を算出し文化差の有無を判定する．本手法の実現にあたり，取り組むべき課題は以下の 2 点である．

文化類似度の算出

文化類似度の指標の一つとして，言語ごとの統合ベクトル同士の cos 類似度を算出する．また，異言語の対象概念の統合ベクトル間に差がなくても，関連する概念に異言語間で差がある場合もあるので，類義語も考慮した文化類似度の算出法が必要である．

文化差の基準となる閾値の同定

文化類似度がどの程度だと，人が文化差を感じるのかが明らかになっていないので，閾値を定める必要がある．文化差のある概念の検出が目的のため，評価指標として文化差有りの概念の再現率と適合率の F 値をも考慮する．

以上の課題を解決するために，本研究では異言語間で対訳になっている概念の統合ベクトルとその統合ベクトルの周辺に存在するベクトルを考慮して文化類似度を算出する．その後，概念の文化差の有無を人手で判定した結果に基づいて最適な閾値と文化差検出精度を求める．

具体的には，一つ目の課題である文化類似度の算出を行うために概念に含まれている複数の単語のベクトルを分散表現空間から取得し，統合ベクトルを作成する．作成した統合ベクトルの cos 類似度を異言語間で測定する．また，その際に日本語統合ベクトルの周辺にある類義語 50 単語を取り出し，異言語間で対訳となっている単語について評価する．統合ベクトルと類義語の評価値によって文化類似度を算出する．そして，二つ目の課題である閾値の同定をするために，日英で対訳となっている概念 200 個を用いて文化差検出を行う．概念の文化差有無の個数は同数とする．その際に文化差の有無を決める文化類似度の仮閾値を 0.00 から 1.00 まで，0.01 刻みで用いる．この結果と人手で判断した結果が一致した率を正確さ(Accuracy)とする．Accuracy を用いて最適な閾値を定める．

その後，導き出した最適な閾値を用いて本研究の文化差検出精度を検証する．閾値の同定で使用した概念とは別の概念 100 個を用いて評価を行い，提案手法の有効性を検証した．本研究の貢献は以下の通りである．

文化類似度の算出

統合ベクトルと類義語の評価値を用いることによって文化類似度を算出した．この算出方法を用いて実験データの文化差有の概念グループと文化差なしの概念グループの文化類似度の平均を算出した．この数値を用いて t 検定を行なったところ，2 群間の平均に有意差があるのを確認できた．

文化差の基準となる閾値の同定

仮閾値を 0.64 とした時に Accuracy の値が最大になる．よって，文化差の判定に用いる閾値は 0.64 とわかった．また，閾値の同定に使用した概念とは別の概念を用いて評価した．すると，Accuracy は 56.5%となり，閾値 0.64 は有効であることがわかる．

Detection of cultural differences using distributed expressions between different languages

Narifumi Oi

Abstract

Recently, the quality of machine translation has been improved, and multilingual communication and cross-cultural collaboration using machine translation have become possible. In such communication, though the translation is right, but communication discrepancy may be generated by the cultural difference. For example, "egg" is eaten raw in Japan, but not abroad. When Japanese explain to foreigners about raw egg dishes such as "tamago kake gohan", they will interpret that the eggs are heated in some way, even if they are not heated. There is “Cultural Difference Detection of Translation Using Image Feature Vector” in order to dissolve such discrepancy of the communication. However, since this method depends on the result of image retrieval, only the main image is retrieved, and there is a case in which the image with the cultural difference can not be acquired. And, the cultural difference can’t be detected by the abstract concept which is difficult to express as an image.

In this paper, we propose a method to detect cultural differences by mapping distributed expressions of words between different languages. Since the method is based on text information, cultural differences in abstract concepts can also be detected. Concretely, distributed expression space of each language is made using Wikipedia corpus of each language, and both spaces are aligned. Next, by using the concept dictionary, the vector of the word group of each language attached to the object concept (Synset) is acquired, and the integration vector of the object concept of each language is made. After that, the cultural similarity is calculated using the integration vector, and the existence of the cultural difference is judged. In the realization of this technique, following 2 points should be tackled.

calculation of cultural similarity

As an indicator of cultural similarity, we must calculate cosine similarity between integration vectors for each language. Also, even if there is no difference between integration vectors of object concepts of different languages, there may be a difference between related concepts of different languages. Therefore, a method of calculating cultural similarity considering synonyms is necessary.

Identification of thresholds for cultural differences

It is necessary to decide the threshold, because how much degree of the cultural similarity the human senses the cultural difference is not clarified. Because since the objective is to detect concepts with cultural differences,

the accuracy of detecting concepts with cultural differences is also considered as an evaluation index.

In this study, to solve the above problems, we calculate the cultural similarity by considering the integration vector of concepts which are bilingual between different languages and the vector existing around the integration vector. Then, the optimum threshold value and the cultural difference detection accuracy are obtained based on the result of judging the existence of the cultural difference of the concept manually.

Concretely, in order to calculate the cultural similarity which is the first problem, the vectors of plural words included in the concept are acquired from the distributed expression space, and the integration vector is made. We measure the cosine similarity of the synthesized integration vector between different languages. We also evaluate 50 synonyms around the Japanese integration vector for words that are bilingual between different languages. We calculate the cultural similarity based on the evaluation values of the integration vector and synonyms. To identify the second problem, we detect cultural differences using 200 concepts that are bilingual in Japanese and English. The number of concepts with or without cultural differences shall be the same. A tentative threshold of cultural similarity which determines the presence or absence of cultural differences is used from 0.00 to 1.00 in intervals of 0.01. The rate at which the result coincides with the result judged manually is defined as accuracy. The optimum threshold is determined by Accuracy.

Then, the derived optimal threshold is used to verify the accuracy of cultural difference detection in this study. The evaluation was carried out using 100 concepts different from the concept used in the identification of the threshold, and the effectiveness of the proposed method was verified. The following is the contribution.

calculation of cultural similarity

We calculated cultural similarity by using the evaluation values of the integration vector and synonyms. Used this calculation method, we calculated the average of cultural similarity between the concept group with cultural difference and the concept group without cultural difference in the experimental data. This value was used for the t-test.

There was a significant difference in the mean between the two groups.

Identification of thresholds for cultural differences

When the provisional threshold value was 0.64, Accuracy is maximum. The evaluation was performed using a different concept from the used to identify the threshold. The accuracy was 56.5%, indicating that the threshold value of 0.64 is effective.

異言語間の分散表現を用いた文化差検出

目次

第1章 はじめに

近年，コロナ禍により対面でのコミュニケーションが困難になってきている．特に,国境を越えた人の行き来が制限されているので，異言語での機械翻訳を用いたオンラインでのコミュニケーションが増えている．その時に,翻訳は合っているが文化差によってコミュニケーション齟齬が生まれてしまう場合がある．例えば,「卵」という概念がある．日本には卵を生で食べる文化があるが，外国にはない．日本人が外国人に卵かけご飯などの卵を生で食べる料理の説明をしても，加熱処理をしているとは言っていないのに外国人は卵に何らかの加熱処理をしていると解釈してしまうだろう．このようなコミュニケーションの齟齬を解消するために,画像検索を用いた対訳ペア間の文化差検出がある．しかしながら,画像検索では主要な画像しか取得できず，文化差のある画像が検索されない場合がある．また、画像として表現しにくい抽象的な概念では文化差を検出できない．

そこで,本研究では単語の分散表現を用い,異言語間で単語の分散表現をマッピングして文化差を比較する．単語ベースのため,抽象的な概念の文化差も検出しやすいと考えられる．具体的には言語ごとの分散表現空間を作成し,そこから概念辞書を用いて,対象概念(Synset)に紐付けされている各言語の単語群のベクトルを取得し,言語ごとの対象概念の統合ベクトルを作成する．分散表現空間の作成にはWikipediaの文章を使用する．Wikipediaの文章は言語ごとに書かれている内容が違うため，文化差の検出に適している．その後、統合ベクトルを用いて文化類似度を算出する．算出した文化類似度に基づいて文化差の有無を判定する．本手法の実現にあたり，取り組むべき課題は以下の2点である．

文化類似度の算出

文化類似度の指標の一つとして，言語ごとの統合ベクトル同士のcos類似度を算出する．また,異言語の対象概念の統合ベクトル間に差がなくても，関連する概念に異言語間で差がある場合もあるので，類義語も考慮した文化類似度の算出法が必要である．

文化差の基準となる閾値の同定

文化類似度がどの程度だと，人が文化差を感じるのかが明らかになっていないので，閾値を定める必要がある．文化差のある概念の検出が目的

のため，評価指標として文化差有りの概念の再現率と適合率のF値をも考慮する．

以下，本研究では 2 章において多言語間でのコミュニケーションにおける文化差を説明し，それに対する現状での文化差へのアプローチと問題点を説明する．続いて，3 章において単語分散表現を用いた文化差の検出方法を説明し，4 章において，3 章で説明するアプローチ課題となる文化差の基準となる閾値の同定についての説明を行う．そして，5 章では 3 章の文化差検出方法と 4 章で導出した最適な閾値を用いて，文化差を検出できるのか評価を行う．

第2章 多言語コミュニケーションにおける文化差

本章では, 本研究で取り扱う文化差について, 具体例を用いて説明する. また, 文化差検出の関連研究について記述する.

2.1 多言語コミュニケーションの齟齬

近年，様々な要因によって機械翻訳を用いた多言語での異文化コミュニケーションが活発化している．例えば，インターネットなどの通信技術の向上などがある．さらに，コロナ禍によって対面でのコミュニケーションを行うのが難しくなってきている．特に感染リスクが高くなる国境を越えた対面でのコミュニケーションは難しい．よってインターネットを介した異文化コミュニケーションが爆発的に増えている．しかしながら，誤訳や文化差によって自分が伝えたい情報をうまく相手に伝えることができず，コミュニケーションに齟齬が生じる場合がある．誤訳の問題は，固有名詞のような翻訳先言語に対応する単語や同音異義語のような曖昧性のある単語によって引き起こされる．一方，正しく翻訳されたとしても，文化差によって情報の受け手が情報の送り手の意図と異なる意味で解釈し，コミュニケーションの齟齬が生じる．

たとえば，卵の文化差がある．日本では卵を生で食べる食文化があるが，アメリカなどの外国にはこのような文化はない．日本人は「卵かけご飯」と聞くと，生卵をご飯の上にかけた料理を想像するだろう．しかし，卵を生で食べる文化のない国で生まれ育った人たちはガパオライスのように目玉焼きをご飯の上に乗

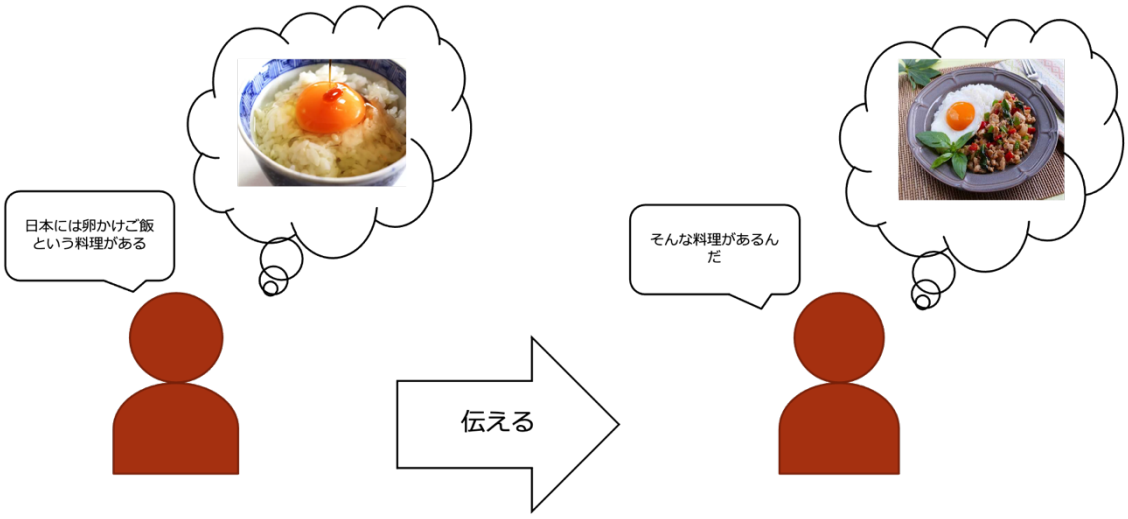

図 1:文化差によるイメージの違い

乗せた料理を想像するだろう．したがって，卵を食べない文化圏の人にとって卵料理は卵に何かしらの加熱処理をした料理となる．図 1 のように卵を生で食べる文化圏の人が卵を生で食べない文化圏の人に生卵を使った料理を伝えようとしても，その料理は卵に加熱処理をして食べるものだと勘違いしてしまう場合がある．

本研究では，このような異文化コミュニケーションをする時に齟齬を起こすような概念の違いを文化差とし，この文化差を検出することを目的とする．

2.2 関連研究

次に，多言語コミュニケーションにおける文化差の検出方法に関する既存の研究を示す．

既存の文化差を検出する手法として，画像特徴量を用いた対訳の文化差検出が存在する[1]．この手法は，単語で検出される画像の特徴量を用いて文化差の有無を自動判別する．具体的には，本研究と同じく概念辞書で同一概念に紐づけられている単語を用いる．それらの単語を用いて画像検索を行い，取得された画像の特徴ベクトルを生成する．生成されたベクトル間の類似度を計算し，その類似度に基づいた文化差を検出する．

他にも，吉野らによる Wikipedia を用いた文化差検出手法の提案がある．[5] まず，文化差の定義として第 1 種の文化差と第 2 種の文化差を定義している．第一種の文化差は Wikipedia の言語間リンクを利用したもので，リンクの有無で文化差を判定する．具体的には「だてマスク」という単語は日本語の新語のため，日本語版 Wikipedia には記事が存在するが，多言語版には存在しない．よって第 1 種の文化差は文化の有無である．第 2 種の文化差は Wikipedia の記事に含まれる国名，言語名の数を利用したものである．各言語版の国名，言語名が多い場合は，それぞれの国におけるその言葉の説明であるため，文化差があると考えられる．よって，第二種の文化差とは二つの国の文化差を比較する場合，どちらの文化圏にも存在するが，それぞれの文化圏で意味の異なるものである．例として日本にも中国にも存在する「醤油」が挙げられている．吉野らは上述した第 2 種の文化差の検出方法の提案を行なっている．日本語版 Wikipedia の検索語を「日本」とし，このような検索語を国名関連ごとする．各国版の Wikipedia の国名関連語の数の違いによって，文化差を判定している．言語 A と言語 B の Wikipedia を用いて文化差検出を行う場合，言語 A，B の記事において，両記事とも言語 A

の国名関連語が多い，もしくは両記事とも言語 B の国名関連語が多い場合，文化差有としている．

同じく，吉野らによる日本語版・中国版 Wikipedia を用いた文化差検出手法の提案[6]がある．日本人の学生から第 1 種の文化差と第 2 種の文化差のある語句をアンケートで収集する．それらから日本語版，中国語版両方に記述されている語句を抽出し，それらを日本に来ている中国人の留学生に文化差無，第 1 種の文化差，第 1 種または第 2 種の文化差の三つに分類した．このデータセットや Wikipedia のカテゴリなどを用いて文化差を検出できるのか確認している．

第3章 文化類似度の算出

本章では，文化差を検出するための本研究でのアプローチを説明する．文化差を検出するために本研究では単語分散表現を用いて文化差の有無を自動判別する手法を提案する．

具体的にはまず，テキストデータをもとに日英の単語分散表現空間を作成する．言語が違うと文法が違うので，同じ意味の文章であっても対訳関係にある単語の出現位置が変わってきてしまう．単語の出現位置が変わると単語分散表現も変わってくる．このズレを補正するために異言語間単語埋め込みを行う．次に，概念辞書で同一概念に紐付けされている日英それぞれの単語のベクトルを作成した単語分散表現空間から取り出す．取り出した単語を統合し，統合ベクトルを作成する．この際に日英の統合ベクトル間の類似度を計算する．そして，日本語統合ベクトルの周辺に存在する類義語 50 個を取得する．次に，取得した単語の対訳関係になっている単語を取得する．取得した単語の評価値と先程計算した統合ベクトル間の cos 類似度の加重平均を求める．この数値を文化類似度とする．図 2 は文化差の検出を行う際の内容をフローチャートにして表したものである．以下，それぞれのプロセスについて詳細を記述していく．

3.1 単語分散表現空間の作成

本手法では Word2Vec を用いて単語分散表現空間を作成する．単語分散表現とは，文字や単語をベクトル空間に埋め込み，その空間上の一つの点として捉える事である．Word2Vec とは，大量のテキストデータを用いて各単語の意味をベクトル化することで単語同士の意味の近さを計算できる手法である．本手法では言語ごとの特色がよく表れていて，なおかつ大規模である Wikipedia の文章を使用した．また，日英の二言語を用いて本実験を行なった．単語分散表現空間の次元数は 300 とした．

3.1.1 異言語間単語埋め込み

言語が違うと，同じ意味で対訳となっている文章であっても，対訳となる単語の出現位置が変わってしまう．単語の出現位置が変わると同じ意味の単語であっても分散表現に出現する位置も変わってしまう．なので，言語の違いによる分散表現のズレを補正する必要がある．このタスクを異言語間単語埋め込みという．本研究では Facebook が公開しているライブラリである MUSE を使用し，

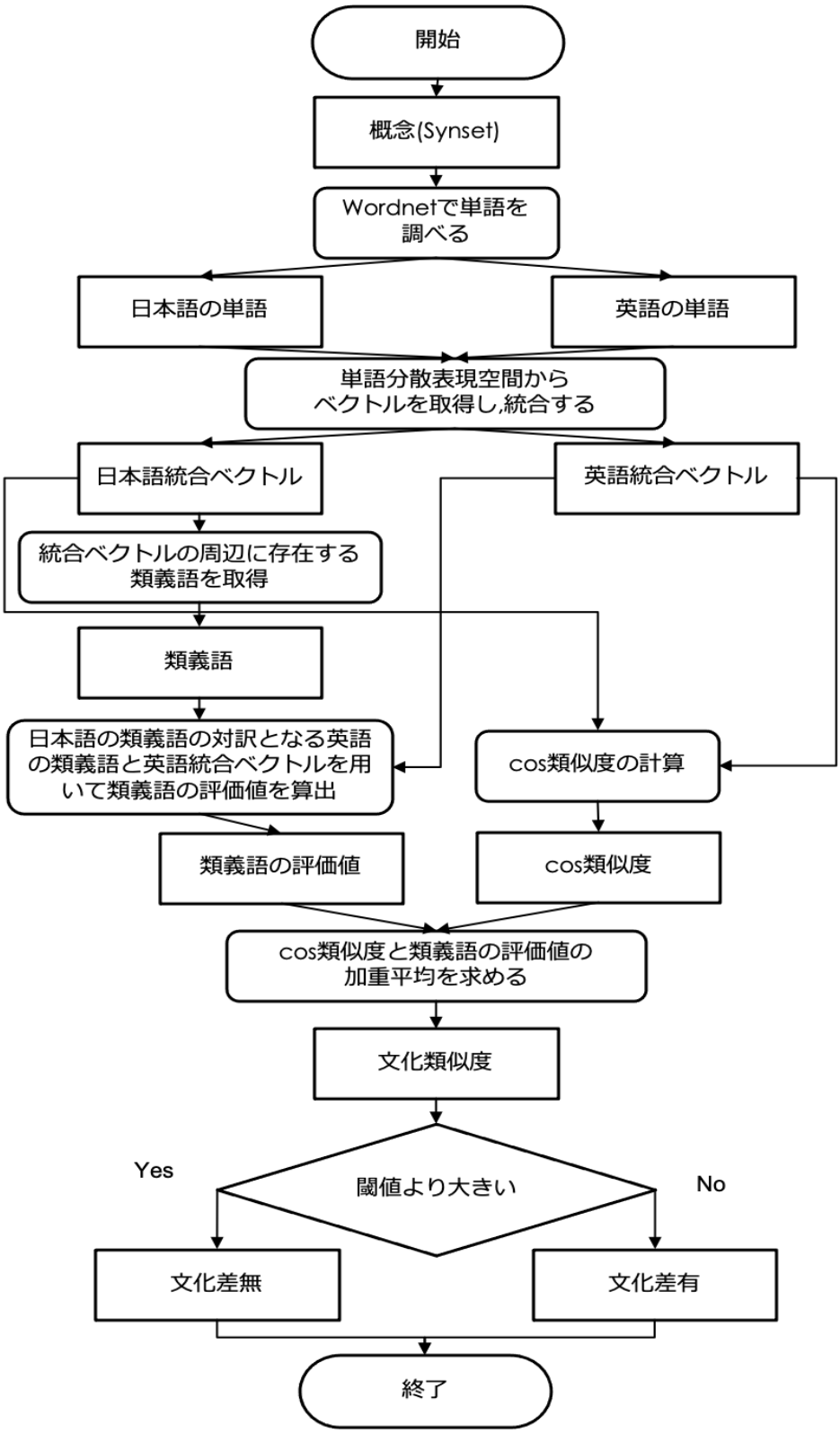

図 2:提案手法のフローチャート

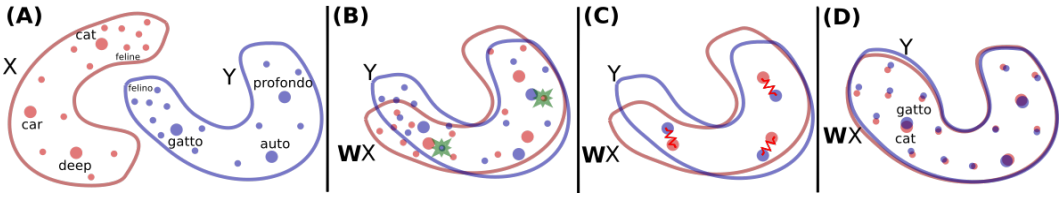

図 3 ： 異言語間単語埋め込みの図解[2]

英語の分散表現空間を日本語の分散表現空間に近づける形で異言語間単語埋め込みを行なった.

まとめると，日本語の単語分散表現空間と英語の単語分散表現を日本語の単語分散表現に近づけるように異言語間単語埋め込みを行なった単語分散表現の二つを用いて本手法の評価を行なった.

3.1.2 統合ベクトルの作成

本研究では,概念に対して文化差の有無を判定する．表 1 のような日本語 Wordnet から同じ概念と定義されている日本語の単語と英語の単語が複数含まれている Synset を取得する．Wordnet とは，プリンストン大学によって作られた英語の概念辞書である．日本語 WordNet は日本語版の概念辞書であり，英語版 WordNet をもとに作られたものである．表 1 は日本語 WordNet で「卵」というワードを検索した例である.

その後，本章で作成した日英の単語分散表現空間からそれらの単語のベクトルを取得する．取得した単語のベクトルを平均化することによって，統合ベクトルを作成する.

以下，「卵細胞，玉子，卵」の Synset の例を用いて統合ベクトル作成の説明をする．まず，Synset に含まれている単語をそれぞれs_1, s_2, s_3と表す.

$$s_1 = \textit{卵細胞のベクトル}$$
$$s_2 = \textit{玉子のベクトル}$$
$$s_3 = \textit{卵のベクトル}$$

以下の式により，これらのベクトルを平均することによって，統合ベクトル c とする.

$$c = s_1 s_2 s_3 / 3$$

表１：「卵」と検索した時の WordNet の Synset の例

日本語の Synset	英語の Synset
御玉, 玉子, 玉, 鳥の子, 卵子, お玉, 鶏卵, 卵	egg, eggs
卵細胞, 卵子, 卵	egg, cell, ovum
卵細胞, 玉子, 卵	egg

3.2 統合ベクトルの類義語の測定

異言語の対象概念の統合ベクトル間に差がなくても，関連する概念に言語間で差がある場合もあると考えられるので，類義語も考慮した文化類似度の算出法が必要である．

本手法では，本章で作成した単語分散表現空間内において 3.1.2 節で作成した日本語統合ベクトルの周辺に存在する類義語 50 単語を取得する．取得した類義語から対訳関係にある英語の単語を取得し，英語の統合ベクトルとの cos 類似度を測定する．図 4 は統合ベクトルの類義語の測定方法を表したフローチャートである．

3.2.1 類義語の対訳関係の判定方法

日英の統合ベクトルの周辺から取得した類義語が対訳関係となっているのかの判定には日本語 WordNet を使用する．具体的にはまず，日本語の概念から作成した日本語の統合ベクトルの類義語が含まれている Synset を WordNet で検索する．次にこの Synset に対応づく英語の Synset を見つける．そして，その Synset から統合ベクトルを作成する．この統合ベクトルを類義語の対訳関係の単語とする．

3.2.2 対訳関係の類義語の評価方法

3.2.1 節で取得した日本語の類義語の対訳関係にある統合ベクトルと．英語の概念の統合ベクトルの cos 類似度を測定する．50 単語分の cos 類似度が算出されるので，それらの平均を取る．平均を取った数値を類義語の評価値とする．

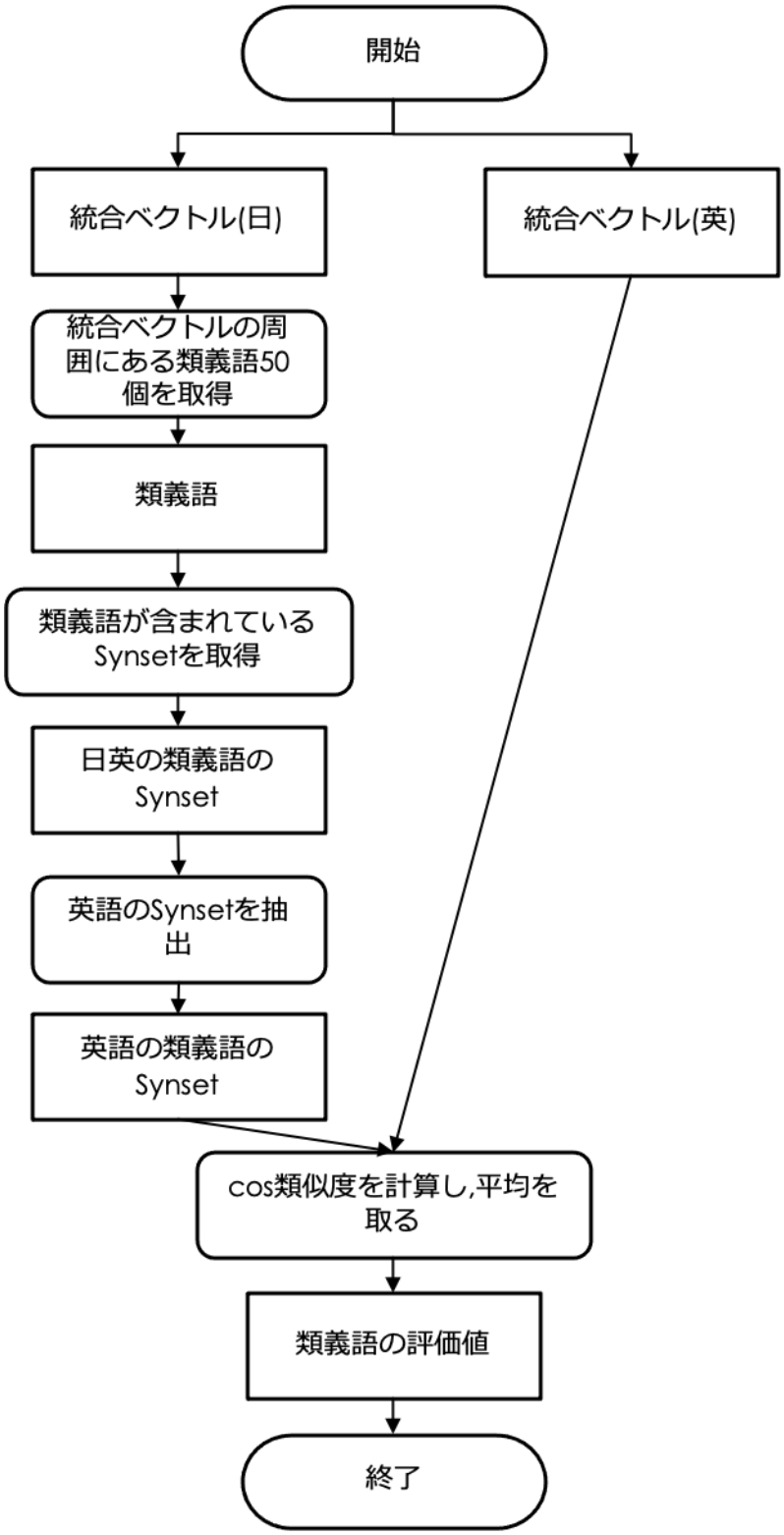

図 4: 統合ベクトルの類義語の測定方法のフローチャート

3.3 文化類似度の算出

文化類似度の指標の一つとして，言語ごとの統合ベクトル同士の cos 類似度を算出した．もう一つの指標として統合ベクトルの周辺に存在する類義語の測定を行なった．文化類似度はこの二つの指標を組み合わせたものである．具体的には統合ベクトルの cos 類似度に対訳関係と測定された類義語評価値の加重平均である．重みは 1:1 とする．算出したこの数値を文化類似度とする．文化類似度は以下の式で算出する．

言語Ｌ１のある Synset S^{L1}に含まれる単語w^{L1}の分散表現 s を以下の(1)の式で表現する．

$$w^{L1}{}_i \in S^{L1}(1 <= i <= n) \cdots\cdots (1)$$

言語Ｌ１の統合ベクトルc^{L1}の算出は以下の(2)の式で行う．

$$c^{L1} = \Sigma s_i / n \cdots\cdots (2)$$

上記の(1)，(2)の式により，言語 L1 と言語 L2 の統合ベクトルの cos 類似度は以下の(3)の式で算出する．

$$sim(c^{L1}, c^{L2}) = c^{L1} \cdot c^{L2} / |c^{L1}| \times |c^{L2}| \cdots\cdots (3)$$

統合ベクトルの近傍 50 件N_cは以下の(4)の式で表す．

$$N_{c=}\{w | sim(c, w) \text{の上位} 50 \text{件}\} \cdots\cdots (4)$$

近傍間の対訳関係にある単語ベクトル(類義語の統合ベクトル)の集合$Tr_{N_{c^{L1,}}}$は以下の(5)の式で表す．$translation(w1, w2)$は w1 と w2 が対訳関係であるということを表す述語である．

$$Tr_{N_{c^{L1,}}} = \{(w^{L2} | w^{L1} \in N_{c^{L1}}, w^{L2}, translation(w^{L1}, w^{L2})\} \cdots\cdots (5)$$

よって，上記の式により，文化類似度は加重平均を用いて以下の式で算出する．$get(Tr_{N_{c^{L1,}}}, i)$は$Tr_{N_{c^{L1,}}}$の i 番目の要素を取得する述語である．

$$文化類似度 = \frac{1 \cdot sim(c^{L1}, c^{L2}) + 1 \cdot \sum_{i=1}^{50} sim\left(c^{L2}, get(Tr_{N_{c^{L1}}}, i)\right)/50}{1+1}$$

第4章 文化差の基準となる閾値の同定

本章では提案手法によって文化差の有無を判定するための閾値の説明や，同定方法を説明する．

4.1 評価指標

まず，提案手法が正確に判定できているか，いないかの 4 パターンを示した表を下記の表 2 に示す．

表 2 に示した通り，人手での文化差判定結果と提案手法の文化差判定結果が一致した場合，判定成功とする．逆に人手での文化差判定結果と提案手法の文化差判定結果が一致しない場合，判定失敗とする．提案手法によってどれだけの割合で文化差判定に成功しているかが正確さ(Accuracy)である． 下記に Accuracy の計算方法を示す．

$$Accuracy = \frac{True\text{に該当した個数}}{synset\text{の個数}} \quad \cdots\cdots(1)$$

文化差の有無を調査した概念に対して，下記の表の True に該当する概念の個数を数える．Accuracy は調査した概念の個数を分母にし，提案手法で判定した結果が True になる概念の個数を分子にすることによって，提案手法がどのくらいの割合で文化差の有無を判定できているのかを示す．

提案手法において重要なのは Accuracy だけではない．本研究は文化差有の概念を見つけることなので，文化差有の検出精度も重要である．よって，文化差検出の評価指標として文化差有の概念の再現率，適合率，再現率と適合率の F 値を用いる．下記に文化差有の概念の再現率，適合率，再現率と適合率の F 値の計算方法を示す．

表 2:人手の判断と提案手法の判断の比較パターン

		提案手法	
		文化差有	文化差無
人の判断	文化差有	True	False
	文化差無	False	True

$A =$ *人手で文化差有と判定したの概念の個数*

$B =$ *提案手法で文化差有と判定した概念の個数*

$C =$*人手で文化差有と判定かつ提案手法で文化差有と判定した概念の個数*

$$recall = \frac{C}{A} \quad \cdots\cdots\cdots (2)$$

$$precision = \frac{C}{B} \cdots\cdots\cdots (3)$$

$$F\text{値} = \frac{2}{\frac{1}{recall} + \frac{1}{precision}} \cdots\cdots (4)$$

文化差ありの概念いついて，再現率とは上記で示した(2)の式の通り，分母を人手で文化差有と判定した概念の個数，分子を人手で文化差有と判定かつ提案手法で文化差有と判定した概念の個数にすることによって全文化差ありの概念のうち何割を検出できたかを示す.

文化差ありの概念について，適合率とは上記で示した(3)の式の通り，分母を提案手法で文化差有と判定した概念の個数，分子に人手で文化差有と判定かつ提案手法で文化差有と判定した概念の個数にすることによって提案手法で文化差有と検出したもののうち何割があっていたのかを示す.

文化差ありの概念について，F 値とは上記(4)の式で，再現率と適合率の調和平均である.

閾値とは，提案手法で算出した文化類似度に対して文化差の有無を決めるための一定の値である．まず，閾値を最適化する方法を説明する．閾値を最適化するための仮閾値を用意する．値は 0 から 1.0 まで 00.1 刻みで設定する．200 個の概念を使用し，提案手法で文化差の有無を検出する．検出した結果の Accuracy と文化差有の概念の適合率，再現率，適合率と再現率の F 値を総合的に考慮して最適な閾値を決定する．図 5 は最適な閾値を決定するフローチャートである．

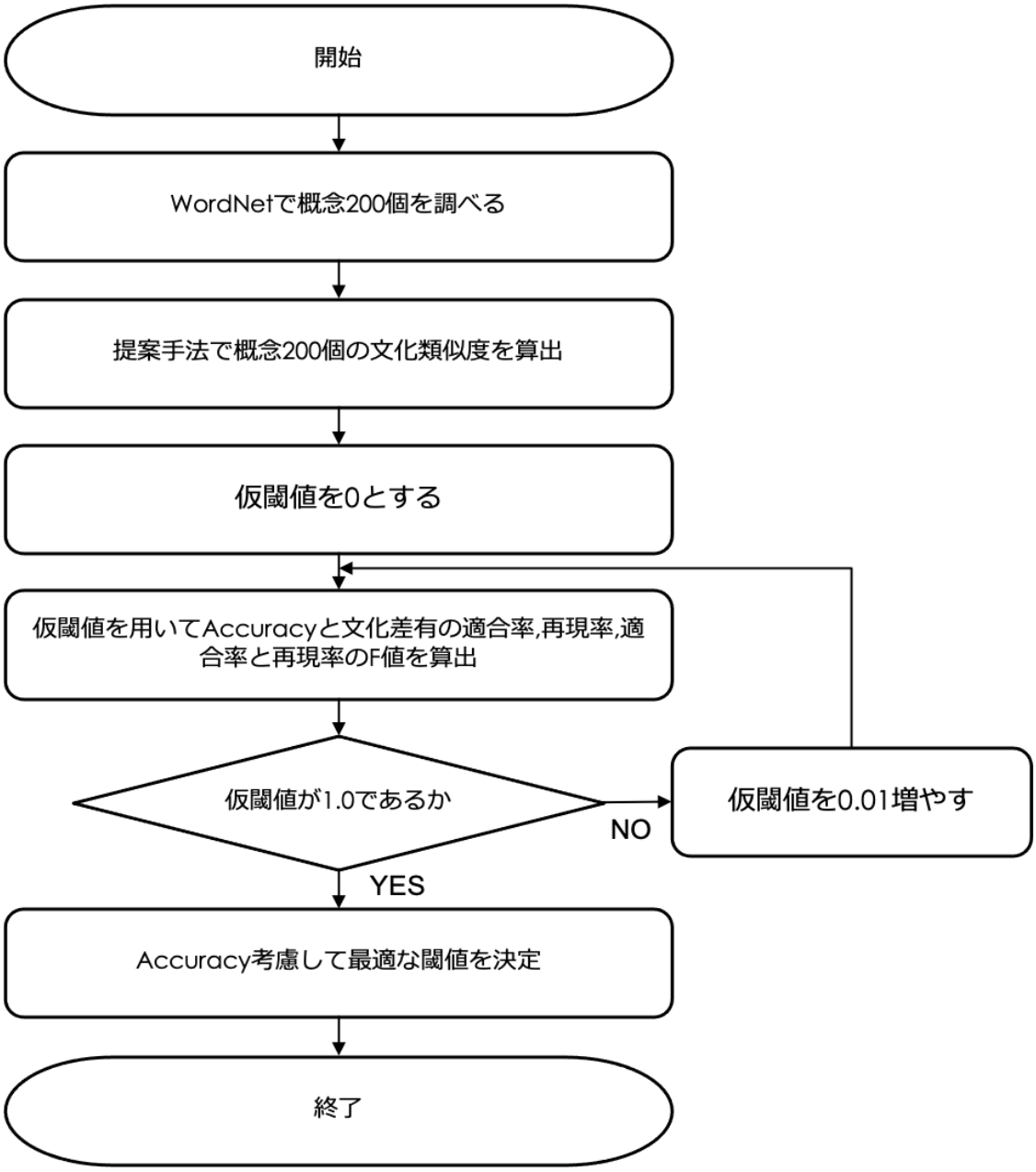

図 5:最適な閾値を求めるフローチャート

4.2 閾値の同定

4.1 節で述べた最適な閾値の同定方法を用い，人手で判定した文化差の有無の判定結果と仮閾値ごとに提案手法で文化差の有無を判定した結果を用いて Accuracy，文化差ありの適合率，再現率，適合率と再現率の F 値を求めた．結果は以下の通りである．図 6,図 7 のグラフの縦軸は割合を，横軸は文化類似度を示している．

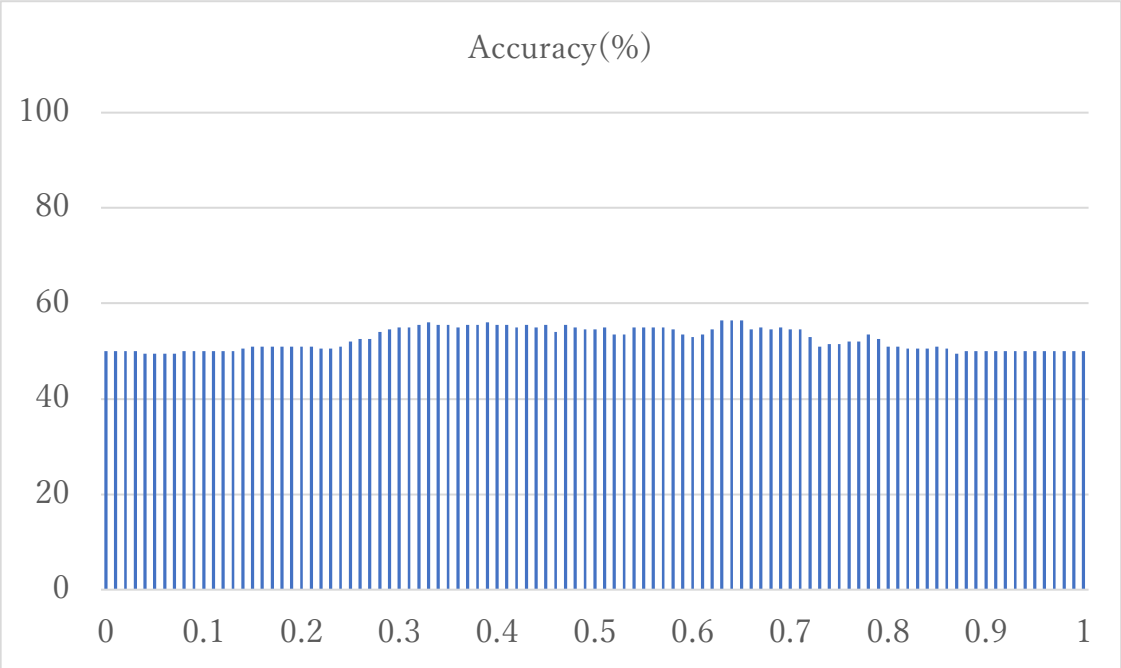

図 6:Accuracy

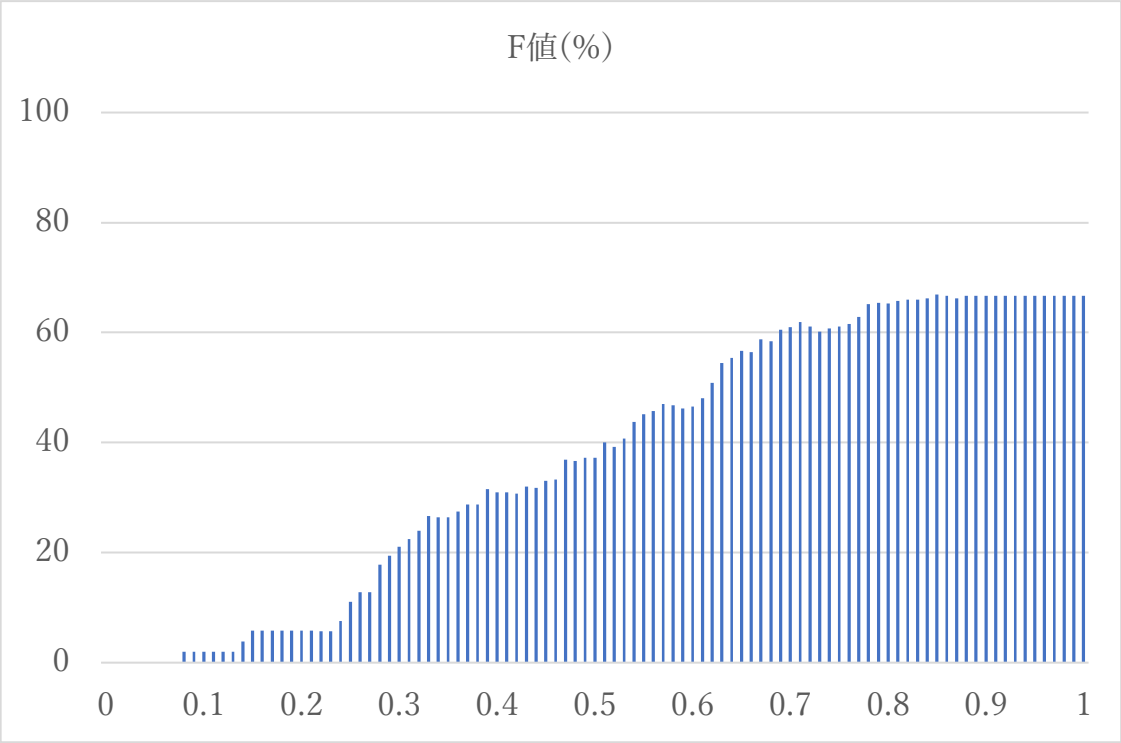

図 7:文化差ありの概念の再現率と適合率の F 値

図 6 は文化類似度による仮閾値ごとの Accuracy である．200 個の概念を用いて評価したため，(1)の式に当てはめると，分母は 200 になる．分子は仮閾値ごとの文化差判定に成功している概念の個数である.仮閾値 0.64 の時に 56.5%で最も高い数値となった.

図 7 は文化差有の概念に関しての再現率と適合率の F 値である．再現率は(2)の式に当てはめて算出すると，文化差有の概念の個数は 100 個なので，分母は 100 となる．分子は人手で文化差有と判定かつ提案手法で文化差有と判定した概念の個数であり，仮閾値によって変動する．適合率は(3)の式に当てはめると分母は提案手法で文化差有と判定した個数で，仮閾値によって変動する．分子は再現率を求めた際と同じ人手で文化差有と判定かつ提案手法で文化差有と判定した概念の個数である.

閾値の同定には Accuracy の値を用いる．閾値 0.64 の時に Accuracy が最も高くなったので，文化差判定における文化類似度の最適な閾値は 0.64 である．閾値 0.64 の時の F 値は 55.4%となった.

4.3 テストデータの t 検定の結果

最適な閾値の同定に使用した文化差有の概念 100 個と文化差無の概念 100 個

表 3:t 検定の結果

	文化差有の概念	文化差無の概念
平均	0.66620655	0.73325145
分散	0.04637145	0.03287843
観測数	100	100
プールされた分散	0.03962494	
仮説平均との差異	0	
自由度	198	
t	-2.3815869	
P(T<=t) 片側	0.00909289	
t 境界値 片側	1.65258578	
P(T<=t) 両側	0.01818577	
t 境界値 両側	1.97201748	

の 2 群の文化類似度を t 検定にかけた．有意水準は 0.05 とした．表 3 に検定結果を示す．文化差有の概念の文化類似度は約 0.67，文化差無の概念の文化類似度の平均は 0.73 となっており，文化差無の概念の文化類似度の平均のほうが高い値になっている．また,P(T<=t)両側の値が 0.01818577 となり，0.05 より低かったので 2 群間の文化類似度の平均値に有意差があることが分かった．よって，文化類似度の算出方法は妥当であることが分かる．

第5章 評価

5.1 評価結果

本章では，最適な閾値を用いて文化差の検出制度の評価を行う．4 章で導き出した通り，最適な文化類似度の閾値は 0.64 であった．この数値を閾値として提案手法で文化差の有無の判定を行う．評価データは 4 章で使用したものとは違う，新たな概念 100 個を用いる．文化差有の概念の個数と文化差無の概念の個数は同数とする．図 8,図 9 のグラフの縦軸は割合を，横軸は文化類似度を示している．図 8 を見てみると，閾値 0.64 の場合，Accuracy は 59%であり，概念 100 個のうち,59 個の概念の文化差の有無の判定に成功したことがわかる．つまり，提案手法は 6 割近い概念の文化差有無に成功した．また，図 9 を見てみると閾値 0.64 の時，文化差ありの概念の F 値は 60.8%となったことがわかる．

Accuracy は 6 割程度の精度となっている．また，閾値 0.64 の時の文化差有の概念に関しての再現率と適合率の F 値は 60.8%と低い値になった．この数値は決して高いと言える数値ではないので，5.2 節で述べる課題を解決し，手法を改善して精度を上げる必要があると考えられる．

5.2 検出誤りの分析

課題として，多義語への対応がある．現状単語分散表現では多義語は全て一つの分散表現として表現されているため，多義語を区別することができない．よって，テキストデータの多い概念に単語ベクトルが大きな影響を受けていて正確な文化差判定に支障をきたしていると考えられる．以下の表 4 はテキストデータの多い概念に単語ベクトルが大きな影響を受けている例である．「高い」という単語は表 4 の Synset のように価格が高いという意味があるが，背が高いなどの物理的に「高い」という意味もある多義語である．表 4 の cos 類似度を見てみると，「高い」という単語とその他の単語を比べたときの cos 類似度だけ低くなっているのがわかる．つまり，「高い」という「単語は価格が高い」という意味以外の意味に影響を大きく受けていることがわかる．

他にも，文化差の有無の基準として文化類似度を用いたが，化類似度は cos 類似度に大きく依存するものである．cos 類似度による分散表現に比較はかな

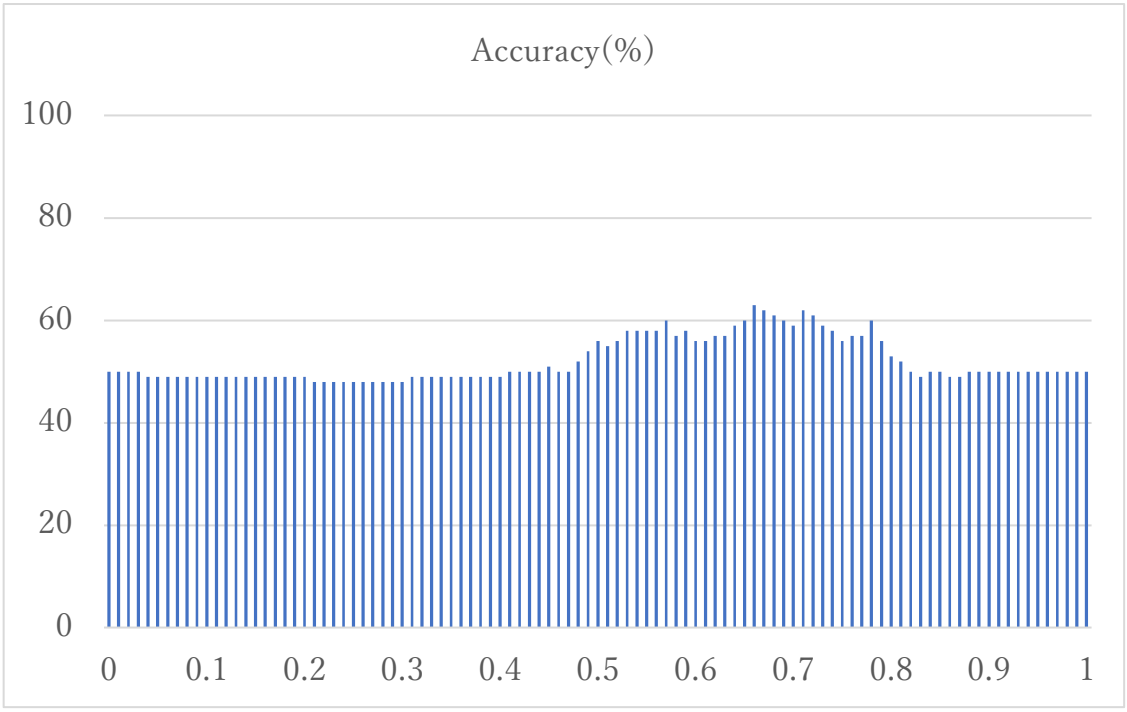

図 8 :Accuracy

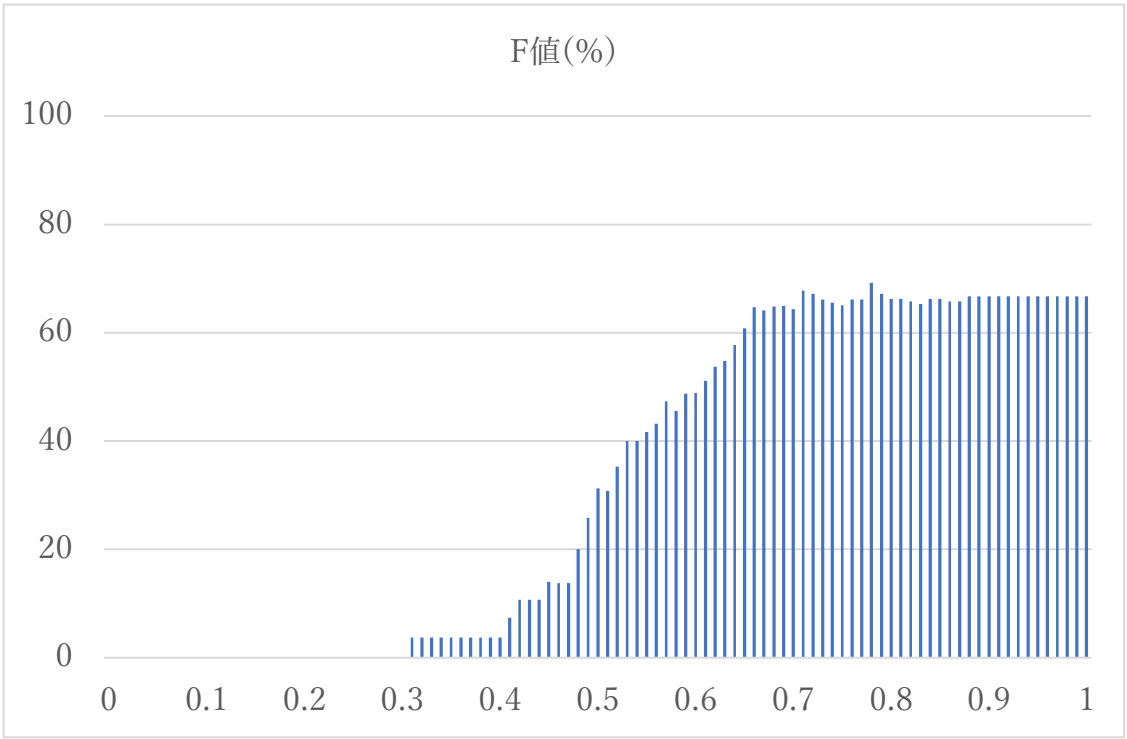

図 9:文化差ありの概念の再現率と適合率の F 値

表 4:多義語の例

Synset	高額,割高,高価,高い
「高い」と「高額」の cos 類似度	0.32247284054756165
「高い」と「割高」の cos 類似度	0.2865821123123169
「高い」と「高価」の cos 類似度	0.3378903269767761
「高額」と「割高」の cos 類似度	0.7336506843566895
「高額」と「高価」の cos 類似度	0.7467356324195862
「割高」と「高価」の cos 類似度	0.697924792766571

り大味なものになってしまっている．よって，サポートベクタマシンなどを用いて分散表現の比較をより詳細に行なっていく必要がある．

言語の違いによる単語分散表現空間のズレを解消するために異言語間単語埋め込みを行なった．異言語間単語埋め込みを行うために，CrossLingualWordEmbedding というツールを用い，ベクトルのアライメントを取るための対訳辞書をデフォルトのものを使用したが，対訳辞書を改良してアライメントの精度を高める必要がある．具体的には評価データで使用した 100 個の概念のうち，9 個の概念の統合ベクトルの類似度が 0.4 以下となっており，正確にアライメントを取れていないと思われる．

また，提案手法は単語分散表現空間を作成した際に使用するテキストデータに依存する．今回は国ごとの特色がよく現れていて尚且つ大規模なテキストデータである Wikipedia の文章を使用した．だが，分散表現が作成されていない単語が存在したり，単語の出現回数が少なすぎて上手く分散表現が作成できていないと思われる単語も存在したので，さらにテキストデータを収集する必要があると考えられる．

5.3 文化差有の検出例

文化差有の概念に対して，文化差検出が成功した例と失敗した例について説明する．表 5 には成功例を，表 6 には失敗例をそれぞれ日本語の Synset，英語の Synset，日本語の Synset の類義語，英語の Synset の類義語，統合ベクトルの cos 類似度を記した．

成功例

表 5 にある概念は「残業」という概念の Synset である．日本語の Synset の類義語には給料関係の単語が，英語の Synset の類義語はスポーツ関係の単語が見られる．日本では時間超過に対して労働時間の概念が強いが，英語圏ではスポーツの時間に対する概念が強いということが推察できる．また，統合ベクトルの cos 類似度も低くなっている．よって，この概念には文化差があるのがわかる．

失敗例

表 6 にある概念は日本には和式便所が，英語圏にはトイレと風呂が同じ場所にあるという文化差があると考えたが，cos 類似度や Synset の類義語ともに文化差は見られなかった．日本語の Synset の類義語に浴槽や浴室といった単語が存在するところを見ると，日本にもトイレと風呂が同じ場所にあるというケースも一般的であることが分かる．よって，この概念には文化差がないということが分かった．

表 5:文化差有の概念の判定成功例

日本語の Synset	オーバータイム, 超過勤務, 超勤, 残業
英語の Synset	overtime
日本語の Synset の類義語	残業, 出勤, 賃金, 割増, 給与, 欠勤, 給料, 未払い, 正社員, 夜勤
英語の Synset の類義語	game-winning, double-overtime, game-winner, triple-overtime, game-tying, playoff, 17-14, powerplay, 3-pointer, 24-21
統合ベクトルの cos 類似度	0.2197721302509308

表 6:文化差有の概念の判定失敗例

日本語の Synset	大壷,WC,トワレット,手洗,幼児用便器,室内便器,トワレ,便所,閑所,W.C.,便器,お手洗い,トイレ,憚り,御手洗い,手洗い,室内用便器,トイレット
英語の Synset	pot,toilet,throne,can,potty,commode,stool,crapper
日本語の Synset の類義語	トイレ,便所,手洗い,水洗トイレ,浴室,便座,水洗,便器,用便,浴槽
英語の Synset の類義語	washroom,restroom,lavatory,bathroom,toilet,cubicle,washrooms,lavatories,room,downstairs
統合ベクトルの cos 類似度	0.5740739107131958

5.4 画像ベースの文化差検出方法との比較

関連研究で示した画像ベースの文化差検出方法である画像特徴量を用いた対訳の文化差検出[1]との比較を述べる.

画像ベースが有利なケース

画像ベースが本手法のテキストベースに比べて文化差検出がしやすいケースは，人が概念を想起した時に，その見た目が違うケースである.画像ベースは文字通り画像を元に文化差検出するので，文化差が見た目に反映される文化差を判定しやすい.

テキストベースが有利なケース

本手法のテキストベースが画像ベースに比べて文化差検出がしやすいケースは,「楽しい」などの画像として表すことのできない概念である.テキストベースでは画像で表すことのできる概念(単語)も単語分散表現空間上にベクトルが生成されているので，文化差検出において画像として表すことのできないケースは有利だと考えられる.

第6章 おわりに

多言語コミュニケーションにおける文化差を解消するために，本研究では単語分散表現から得られるベクトルを多言語の単語分散表現空間から取得し，比較するアプローチを提案してきた．本研究の貢献は以下の通りである．

文化類似度の算出

統合ベクトルと類義語評価値を用いて文化類似度を算出した．この算出方法を用いて実験データの文化差有の概念グループと文化差なしの概念グループの文化類似度の平均を算出し，t 検定を行なったところ 2 群間の平均に有意差があるのを確認できた．

文化差の基準となる閾値の同定

200 個の概念と仮閾値を用い，提案手法で文化差の有無の判定を行なった．それぞれの仮閾値における提案手法の正確さ(Accuracy)を算出した結果，最適な文化類似度の文化差を検出するための最適な閾値は 0.64 とわかった．導き出した閾値が最適か検証するために，実験で使用したものとは別の概念 100 個を用いて，閾値の文化差検出制度を検証した．検証した結果，59 個の概念の文化差の有無の判定に成功した．約 6 割の概念の文化差の有無の判定に成功したが，本研究の提案手法が文化差検出において有効であるというのは難しい数値である．

文化差を検出するために単語分散表現から算出する文化類似度を用いることで，国ごとに記述されているテキストデータから文化差を検出することができた．だが，文化差検出制度は約 60%と低い値にある．本手法を異文化コミュニケーションの場で使用するには，文化差検出制度をさらに高める必要がある．

謝辞

本研究を行うにあたり，ご指導していただいた村上陽平准教授に深く感謝を申し上げます.

参考文献

[1] 西村一球: 画像特徴量を用いた対訳の文化差検出

[2] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, Herve Jegou: Word Translation Without Parallel Data, ICLR2018

[3] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean: Efficient Estimation of Word Representations in Vector Space, Proceedings of Workshop at ICLR2013

[4] Mozhi Zhang, Keyulu Xu, Kenichi Sawarabayashi, Stefanie Jegelka, Jordan Boyd-Graber: Are Girls Neko or Shojo? Cross-Lingual Alignment of Non-Isomorphic Embeddings with Iterative Normalization, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp3180-3189

[5] 宮部真衣, 吉野孝:Wikipediaを用いた文化差検出手法の提案, 情報処理学会論文誌55(1),pp257-266

[6] 諏訪智大, 宮部真衣, 吉野孝:日本語版・中国版Wikipediaを用いた文化差検出手法の提案, 情報処理学会論文誌, Vol55 No.1,257-266(Jan 2014)

[7] 諏訪智大, 宮部真衣, 吉野孝:異文化コミュニケーションにおける重要度を考慮した文化差検出手法の提案, 情報処理学会関西支部 支部大会 講演論 文集

付録:グラフ

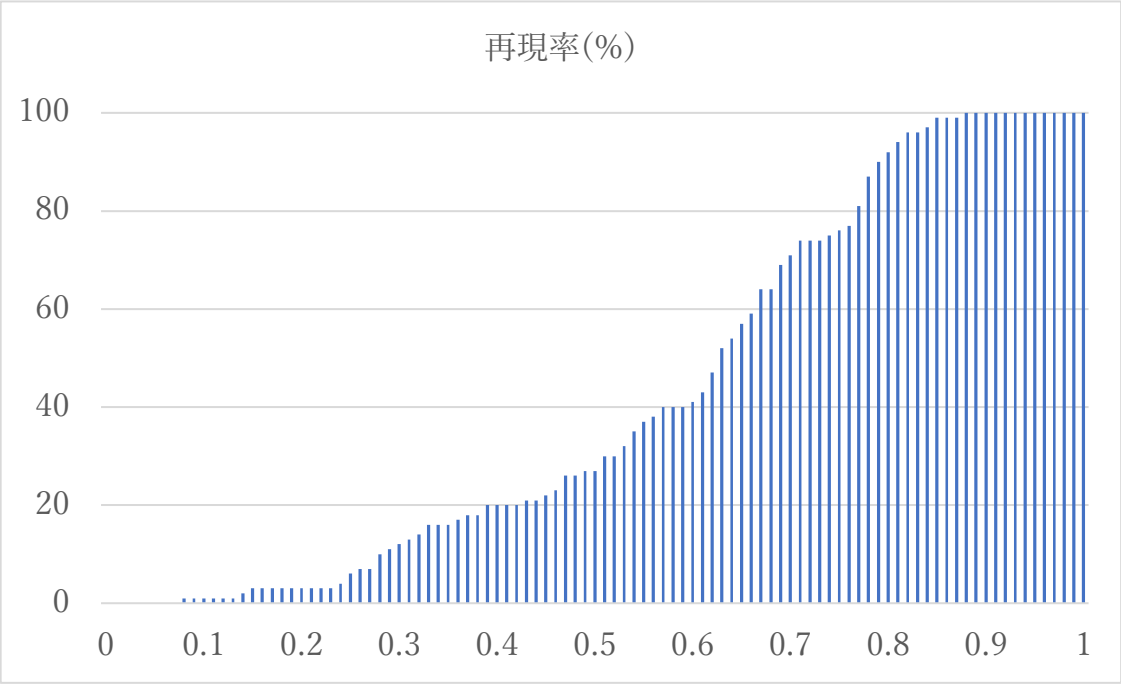

図 10:4 章における文化差有の再現率

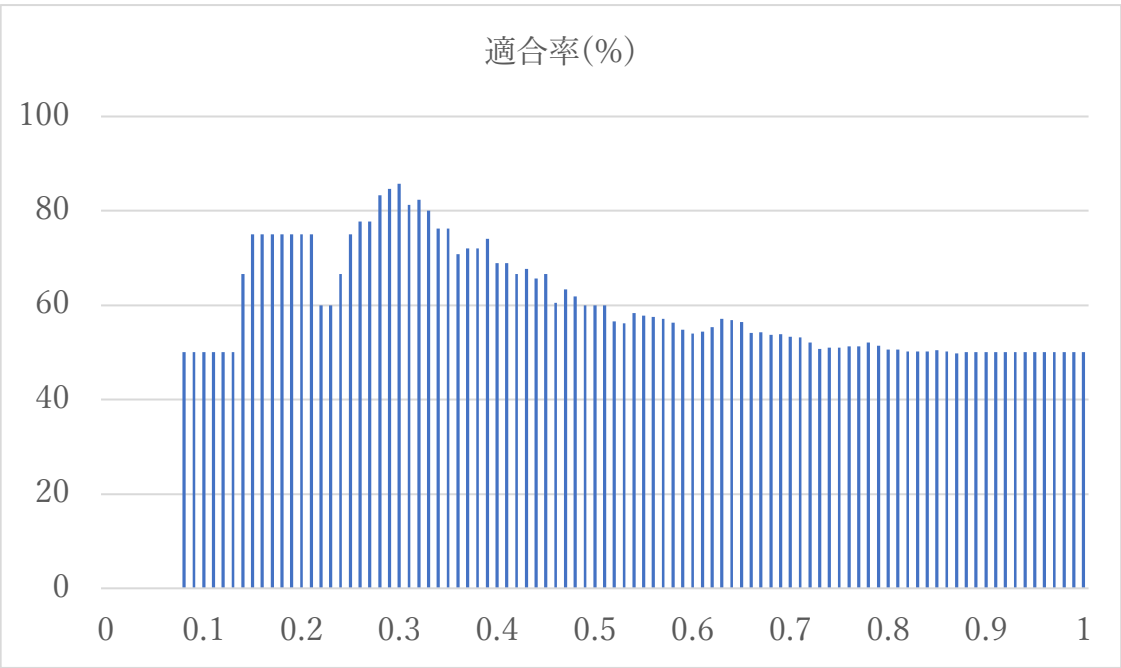

図 11:4 章における文化差有の適合率

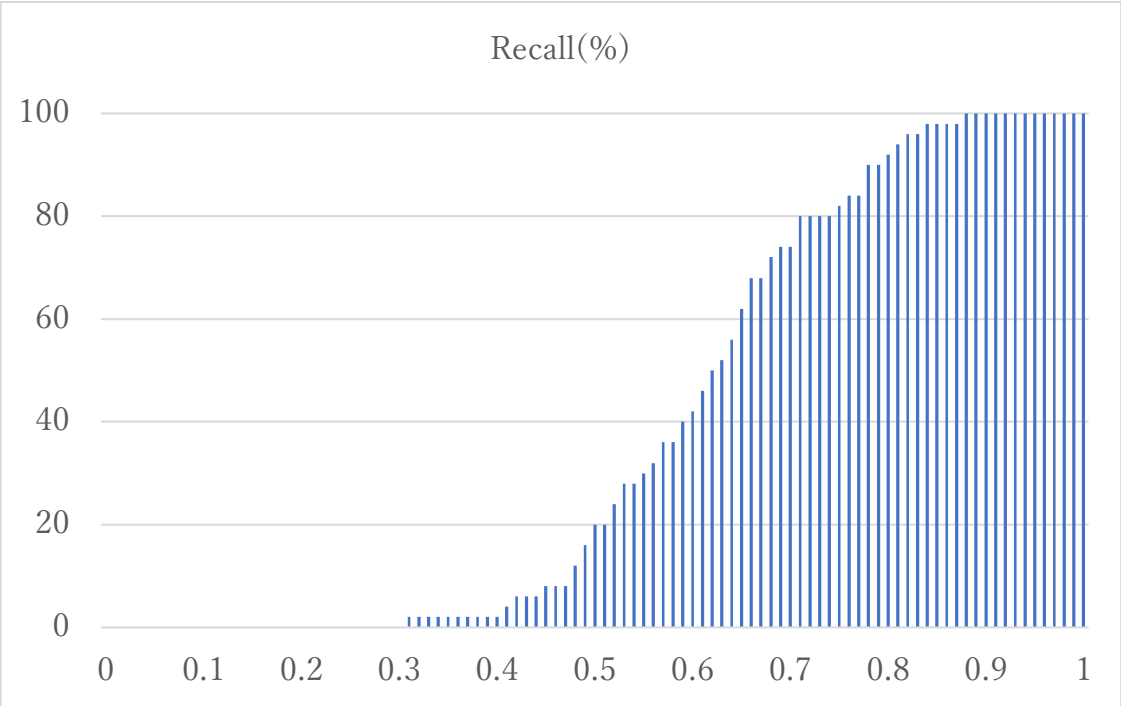

図 12:5 章における文化差有の再現率

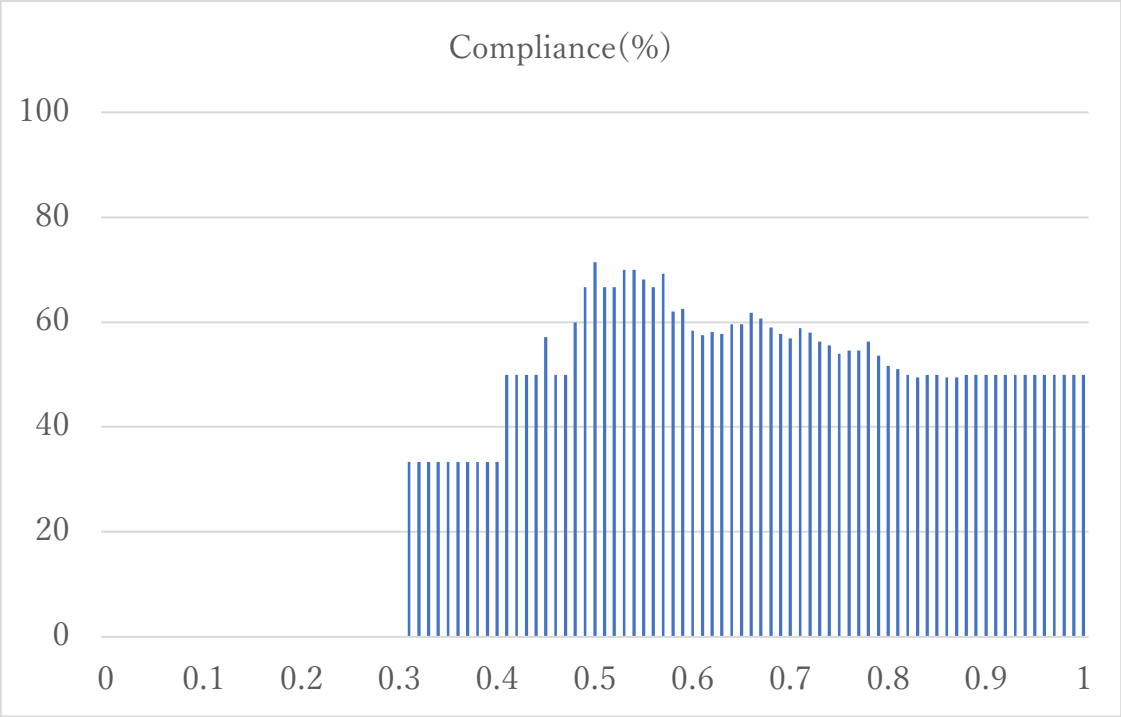

図 13:5 章における文化差有の適合率

付録:ソースコード

1,統合ベクトルの作成のソースコード

```
    import gensim
    import numpy as np
    from scipy import spatial

    #単語分散表現空間の読み込み
    word2vec_model_en =
gensim.models.KeyedVectors.load_word2vec_format('data/vectors-en(en-jp).bin',
binary=True)
    word2vec_model_jp =
gensim.models.KeyedVectors.load_word2vec_format('data/jp_wiki.bin', binary=True)

    #関数 avg_feature_vector は統合ベクトルを作成する
    def avg_feature_vector(sentence, model, num_features):
        #Synset の単語は,_で区切られているので,_で区切る
        l = sentence.split(',_')
        o = []
        #複数の単語が_で区切られて,一つの単語になっているものを区切る
        for i in l:
            m = i.split("_")
            for t in m:
                j = t.replace('¥n','')
                o.append(j)
        k = []
        #unicode を削除
        for i in o:
            m = i.replace(u',¥xa0',' ')
            k.append(m)
```

```
        # 特徴ベクトルの入れ物を初期化
        feature_vec = np.zeros((num_features,), dtype="float32")
        n = []
        #以下,特徴ベクトルを future_vec に入れて,平均化する
        for word in k:
                try:
                        feature_vec = np.add(feature_vec, model[word])
                        n.append(word)
                except:
                        pass
        if len(words) > 0:
                feature_vec = np.divide(feature_vec, len(n))
        return feature_vec

 #sentence_similarity は Synset を読み取り,関数 avg_feature_vector に渡し,日英の
  Synset の cos 類似度を計算する関数
  def sentence_similarity(sentence_1, sentence_2):
      # 今回使う Word2Vec のモデルは 300 次元の特徴ベクトルで生成されているので、num_features も 300 に指定
        num_features=300
        sentence_1_avg_vector = avg_feature_vector(sentence_1, word2vec_model_jp, num_features)
        sentence_2_avg_vector = avg_feature_vector(sentence_2, word2vec_model_en, num_features)
      # 1からベクトル間の距離を引いてあげることで、コサイン類似度を計算
        return  1 - spatial.distance.cosine(sentence_1_avg_vector, sentence_2_avg_vector)

  #Synset を読み込む
  jp = open("data/jp_y100_n100.txt", "r")
  jp_word = jp.readlines()
  en = open("data/en_y100_n100.txt", "r")
```

```
    en_word = en.readlines()

    for e, j in zip(en_word, jp_word):
        result = sentence_similarity(j,e)
        print(result)
```

2.類義語の評価値を計算するためのソースコード

```
    import gensim
    import numpy as np
    from scipy import spatial
    import sqlite3
    import math
    import re
    #英単語を識別するためのもの
    p = re.compile('[a-z]+')
    #Synset のデータベースを読み込む
    conn = sqlite3.connect("data/wnjpn.db")
    #単語分散表現空間の読み込み
    word2vec_model_en =
gensim.models.KeyedVectors.load_word2vec_format('data/vectors-en(en-jp).bin',
binary=True)
    word2vec_model_jp =
gensim.models.KeyedVectors.load_word2vec_format('data/jp_wiki.bin', binary=True)

    #関数 SearchSimilarWords は入力した単語が含まれている Synset を検索する
    def SearchSimilarWords(word):
        # 問い合わせしたい単語が Wordnet に存在するか確認する
        cur = conn.execute("select wordid from word where lemma='%s'" % word)
        word_id = 99999999
        for row in cur:
            word_id = row[0]
```

```
        # Wordnet に存在する語であるかの判定
        if word_id==99999999:
            return
        else:
            pass

        # 入力された単語を含む概念を検索する
        cur = conn.execute("select synset from sense where wordid='%s'" % word_id)
        synsets = []
        for row in cur:
            synsets.append(row[0])

        # 概念に含まれる単語を検索して返す
        for synset in synsets:
            cur1 = conn.execute("select name from synset where synset='%s'" %
synset)
            cur3 = conn.execute("select wordid from sense where (synset='%s' and
wordid!=%s)" % (synset,word_id))
            words = ",_"
            for row3 in cur3:
                target_word_id = row3[0]
                cur3_1 = conn.execute("select lemma from word where
wordid=%s" % target_word_id)
                for row3_1 in cur3_1:
                    ans = row3_1[0] + ",_"
                    words += ans
            ans = words + word
            return ans

    #関数 avg_feature_vector は統合ベクトルを作成する
    def avg_feature_vector(sentence, model, num_features):
```

```
        l = sentence.split(',_')
        o = []
        for i in l:
                m = i.split("_")
                for t in m:
                        j = t.replace('¥n','')
                        o.append(j)
        k = []
        for i in o:
                m = i.replace(u',¥xa0',' ')
                k.append(m)

        feature_vec = np.zeros((num_features,), dtype="float32")

        for word in k:
                try:
                        feature_vec = np.add(feature_vec, model[word])
                        n.append(word)
                except:
                        pass

        if len(words) > 0:
                feature_vec = np.divide(feature_vec, len(n))
        return feature_vec

    def sentence_similarity(sentence_1, sentence_2,mun):
        # 今回使う Word2Vec のモデルは 300 次元の特徴ベクトルで生成されているので、num_features も 300 に指定
        num_features=300
        #統合ベクトル作成
        sentence_1_avg_vector = avg_feature_vector(sentence_1, word2vec_model_jp, num_features)
```

```
        sentence_2_avg_vector = avg_feature_vector(sentence_2, word2vec_model_en,
num_features)
        #日本語の類義語50個取得
        ans_jp = word2vec_model_jp.most_similar(positive=[sentence_1_avg_vector],
topn=50)
        list_jp = []

        for i in ans_jp:
                j = list(i)
                k = j[0].split("¥'")
                list_jp.append(k[0])

        num = 0
        jp_num = 0
        cos_synset = 0
        eng_list = []
        eng_tango = ",_"
        for i in list_jp:
                jp_num += 1
                try:
                        #日本語の類義語の synset 取得
                        j = SearchSimilarWords(i)
                        #Synset は英語と日本語が混ざっているので,英語のみを
抽出
                        k = j.split(',_')
                        k.pop(0)
                        for e in k:
                                if p.fullmatch(e):
                                        eng_list.append(e)
                                else:
                                        pass
                        #データ形式を整える
```

```
                    for t in eng_list:
                        eng_tango = eng_tango + t + ",_"

            except:
                    pass
            try:
                    en_num = 0
                    en_synset_vec = avg_feature_vector(eng_tango,
word2vec_model_en, num_features)
                    #「synset の英単語の統合ベクトル」と「概念の英語の統合
ベクトル」の cos 類似度を算出
                    cos_ans = 1 - spatial.distance.cosine(en_synset_vec,
sentence_2_avg_vector)
                    if math.isnan(cos_ans):
                        print("nan です")
                    else:
                        cos_synset += cos_ans
            except:
                                pass
        cos_synset_num2 = cos_synset / 50
        with open("syn_y100_n100.txt", "a") as g:
                        print(cos_synset_num2, file = g)


jp = open("data/jp_y100_n100.txt", "r")
jp_word = jp.readlines()
en = open("data/en_y100_n100.txt", "r")
en_word = en.readlines()
n = 0
for e, j in zip(en_word, jp_word):
    n += 1
    result = sentence_similarity(j,e,n)
```