

卒業論文

画像特徴量を用いた対訳の文化差検出

指導教官 村上 陽平 准教授

立命館大学
情報理工学部
情報コミュニケーション学科 4 回生
2600160299-2

西村 一球

2019 年度（秋学期）卒業研究 3(2Q)

令和 2 年 1 月 31 日

画像特徴量を用いた対訳の文化差検出

西村 一球

内容梗概

近年、機械翻訳の翻訳精度が向上し、多言語間でのコミュニケーションの支援が出来つつある。その例として、NPO 法人パンゲアが開催する KISSY という異文化サマースクールが挙げられる。KISSY ではケニアやカンボジア、ジョージアといった様々な国から小学生から中学生の児童を集めて、日本人の児童たちとワークショップを行う。KISSY において、会話には様々な言語に対応した機械翻訳ツールを用いる。これによって子供たちの会話を可能にし、協力してワークショップに取り組むことができる。このように、機械翻訳を用いることによって、多言語でのコミュニケーションを行うことが容易になった。

しかし、機械翻訳を使っても会話が上手く成り立たない場合がある。その理由は、同じ概念の単語を機械翻訳で翻訳したときに、翻訳結果が合っている場合でも話者たちの育った環境や文化背景によってはその翻訳結果から想起されるものが異なるためである。その結果、会話がうまく成り立たない場合がある。例えば、日本でよく食べられる”ゴボウ”を機械翻訳で翻訳すると”great burdock”, “burdock”, “greater burdock”といった翻訳結果が得られる。日本人の多くは”ゴボウ”と聞くと根っこのような見た目の食べ物を想起する。しかし、海外では”great burdock”, “burdock”, “greater burdock”からイガのついた草木を想起することが多い。想起されるイメージは全く異なるものであるが、“ゴボウ”と”great burdock”, “burdock”, “greater burdock”は同じ種類の植物である。“ゴボウ”とは、“great burdock”, “burdock”, “greater burdock”で想起される草木の根っこの部分である。日本ではこの根の部分を食べる習慣が持っているため、根っこの見た目を想起してしまう。しかし、海外では”ゴボウ”を食べる習慣がないため、イガのついた草木を想起してしまう。

このような文化差を解消するためには、話者がお互いの文化に関する多くの知識を持つ必要があるため、容易ではない。したがって、文化差の有無を調べ、自動的に検出することが求められる。

そこで、本研究で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定する手法を提案する。具体的には、概念辞書で同一概念に紐づけられている日英のそれぞれ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

基づいて文化差を検出する。本手法を実現するために、本研究で取り組んだ課題は以下の2点である。

文化差の基準となる閾値の同定

既存の手法では、文化差の有無を判断する閾値が明確に出来ていないため、検出した結果をもとに人が確認して文化差を判断する必要がある。よって、文化差の検出精度を最適化する閾値を定める必要がある。

最適閾値を用いた文化差検出精度の検証

現在は、文化差の有無の判断をある程度、人が確認しなければならない。自動的に文化差を検出するためには人が文化差の有無を判断した結果に比べて、同定した閾値を用いた結果がどれほど正確に文化差を検出することができるかを確認する必要がある。

これらの課題に対し、本研究では、提案手法を用いた際に取得された画像を人に文化差の有無を判断してもらう。そして、人手で判断した結果と提案手法の結果と比較し、閾値の値に応じて、提案手法の検出精度がどのように変化するかを調べ、最適な閾値を求める。

具体的には、一つ目の課題である最適な閾値を求めるために同一概念とされる概念を800個分、提案手法を用いて文化差を検出する。そして、仮の閾値を0.00と定め、800個の概念の文化差の有無を判断する。これを仮閾値が1.00になるまで0.05ごとに増やして繰り返していく。それに対し、同じ800個の概念から取得された画像を人手で文化差の有無を判断する。人手で判断した結果に閾値を用いた手法の結果が一致している率を文化差検出の正確さ(Accuracy)として、仮閾値0.00から1.00までの間で最もAccuracyが高かった値を最適な閾値として定める。

その後、別の概念200個分を用いて、最適な閾値を閾値としたときの提案手法の評価を行い、有効性を検証した。本研究の貢献は以下のとおりである。

文化差の基準となる閾値の同定

仮閾値を0.4とした時、Accuracyが77.5%で最も高かったため、文化差と判断するための基準となる閾値は0.4であると分かった。

最適な閾値を用いた文化差検出精度の検証

最適な閾値より閾値を0.4として、概念200個で提案手法を評価した。結果はAccuracyが77%であった。

Cultural Difference Detection of Translation Using Image Feature Vector

Ikkyu NISHIMURA

Abstract

Nowadays, machine translation-mediated multilingual communication is becoming popular because the quality of machine translation is improved. For example, NPO Pangea holds KISSY, Kyoto Intercultural Summer School for Youth, where elementary school and junior high school students from various countries such as Japan, Kenya, Cambodia, and Georgia collaboratively create an artwork. During the collaboration, children can talk about their own ideas each other using machine translation system.

However, the communication is sometimes failed even though machine translation system can translate correctly. This is because images recalled by the same concept are different due to the cultural differences. For example, machine translation translates “ゴボウ (*gobou*)” in Japanese into “great burdock”, “burdock,” and “greater burdock” in English. Japanese imagine a root food from “ゴボウ (*gobou*),” and foreigners imagine a flower from “great burdock”, “burdock,” and “greater burdock.” Those images are different, but they represent same species. The image Japanese imagine from “ゴボウ (*gobou*)” is a root of “great burdock,” “burdock” and “greater burdock” because Japanese often eat a root of burdock. On the other hand, foreigners imagine that flower because foreigners do not eat a burdock but see it. It is not easy that people solve this cultural difference because they need to learn many knowledges about both cultures.

Therefore, we propose a method that detects a cultural difference using image feature vector automatically. Specifically, an image search is performed using Japanese and English words linked to the same concept dictionary, and feature vectors of the acquired images are generated. And then, it calculates the similarity between the generated vectors and detects cultural differences based on the similarity. To this end, we address the following two issues.

Threshold value to decide cultural difference

We need to find the best threshold value detecting cultural differences

because the criteria for judging whether cultural differences exist are not obvious in the existing researches.

Evaluating accuracy of the proposed method using the optimal threshold value

Only people can judge whether cultural differences exist or not. To evaluate accuracy of the proposed method, we need to compare the results from our method with the human judgements.

In this research, we collect human judgements by asking people to judge whether they recognize a cultural difference between images searched by the proposed method. And then we compare human judgements and our detection results with various threshold values, and we find the optimal threshold value.

Specifically, we have targeted on 800 concepts to automatically detect cultural differences using the proposed method. While increasing a threshold value from 0.0 to 1.0 by 0.05, cosine similarities between every pair of feature vectors of the concepts are compared with the threshold value to judge whether cultural differences exist. Second, we have shown people the images of the same 800 concepts, and collected human judgements. After that, we compared our detection results and human judgements to optimize threshold value.

Finally, to validate the optimal threshold value, we applied it to other 200 concepts. The following is the contribution.

Threshold value to decide cultural difference

When threshold value is 0.45, the proposed method achieved the highest accuracy, 71%. This result indicates the optimal threshold value is 0.45 for cultural difference detection.

Evaluating accuracy of the proposed method using the optimal threshold value

To validate the optimal threshold, we have applied our method to 200 concepts to detect cultural differences. As a result, the accuracy of our method achieved 69.8%, which shows the optimal threshold is valid.

卒業論文タイトル

目次

第 1 章 はじめに	1
第 2 章 多言語コミュニケーションにおける文化差	3
2.1 多言語コミュニケーションの齟齬	3
2.2 関連研究	5
第 3 章 文化差の検出手法	7
3.1 画像特徴量を用いた検出方法	7
第 4 章 閾値の最適化	12
4.1 文化差検出の正確さ	12
4.2 最適な閾値	13
4.2.1 閾値を最適化する方法	13
4.2.2 人手での判断	14
4.2.3 人手での判断と提案手法の比較	17
第 5 章 評価	19
第 6 章 おわりに	21
参考文献	24
付録：ソースコード	1

第1章 はじめに

近年, Google 翻訳に代表される機械翻訳の翻訳精度が向上し, 異なる言語を話す人同士のコミュニケーションの支援が出来つつある. その例として, 認定 NPO 法人パンゲアという組織が毎年京都で開催する KISSY という異文化サマースクールが挙げられる. KISSY ではケニアやカンボジア・ジョージアといった様々な国から小学生から中学生の児童が集まり, 日本の児童たちとともにクレイアニメを創作するワークショップを行う. それらの子供たちは英語のような共通の言語を話すことができるわけではなく, それぞれが自らの国の言葉話すため, 普段通りに会話を行うことができない. そこで KISSY では英語やクメール語やロシア語など 5 カ国の言葉に対応したパンゲア独自の機械翻訳ツールを使用して会話をしている. これによって, 異なる言語を話す子供たちが協力してワークショップを行うことができている. このように, 多言語間でのコミュニケーションにおいて機械翻訳を用いることで, 異なる文化の話者たちと会話することが容易になった.

しかし, 異なる国の人との会話において機械翻訳を使用しても会話がうまく成立しない場合が存在する. その理由として, 同じ概念の単語を機械翻訳した際, 翻訳結果が合っている場合でも話者たちの育った環境や文化背景によってはその翻訳結果から想起されるものが異なる場合がある. その結果, 会話が上手く成り立たない場合がある.

このような文化差を解消するには, 話者がお互いの文化に関する幅広い知識を持ち, それを踏まえた上で会話する必要があるため, 容易ではない. したがって, 文化差の有無を調べ, 自動的に検出することが求められる.

そこで, 本研究で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定する手法を提案する. 具体的には, 概念辞書で同一概念に紐づけられている日英のそれぞれ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て文化差を検出する. 本手法を実現するために, 本研究で取り組んだ課題は以下の 2 点である.

文化差の基準となる閾値の同定

既存の手法では, 文化差の有無を判断する閾値が明確に出来ていないため, 検出した結果をもとに人が確認して文化差の有無を判断する必要がある.

自動的に文化差を検出するために、文化差の検出精度を最適化する閾値を定める必要がある。

最適閾値を用いた文化差検出精度の検証

現在は、文化差の有無の判断をある程度、人が確認しなければならない。自動的に文化差を検出するためには人が文化差の有無を判断した結果に比べて、同定した閾値を用いた結果がどれほど正確に文化差を検出することができるかを確認する必要がある。

この課題に対し、既存の手法では文化差の有無を判断する閾値が存在しない。それでは、何を基準に文化差が存在すると判断して良いかわからないため、文化差の有無を判断するための基準として閾値を定める必要がある。本研究では、既存研究で行われていなかった、文化差の有無を人の判断で確認する。そして、提案手法の結果と比べて、より詳しく文化差の有無を調査するとともに文化差の検出精度を最適化する閾値を定めることを目的としている。

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、画像の特徴ベクトルから類似度を計算することで文化差を検出するアプローチを説明し、4章において、3章で説明するアプローチの課題となる、最適な閾値を定めることの説明を行う。そして、5章では3章でのアプローチ方法と4章での最適な閾値によって、文化差を検出することができるか評価を行う。最後にこれからの課題について述べる。

第2章 多言語コミュニケーションにおける文化差

2.1 多言語コミュニケーションの齟齬

最近、異なる言語の話者や違う国の人と話す機会が増えている。その理由には、以下のような理由が挙げられる。

- グローバル化
- インターネット・遠隔通信技術の向上

グローバル化により、国を跨いだ人の往来が増えている。また、国外での仕事も増えている。そして、インターネット・遠隔通信技術の向上・普及で海外の人と通話やチャットなどコミュニケーションの手段が増えている。これらのことから、異なる言語の話者と話す機会が増えている。その時には、英語などお互いに共通する言語を用いて会話をする。たとえ、共通の言語がなくとも、語学が達人でなくとも、近年では機械翻訳の精度向上により、コミュニケーションの支援ができてきている。

しかし、このような多言語のコミュニケーションにおいて、話し手が伝えたいニュアンスと異なる形で聞き手に伝わることもある。理由として、以下のようなケースが考えられる。

- 翻訳の際に齟齬が生じる場合
- 発音により齟齬が生じる場合
- 文化や環境により齟齬が生じる場合
- 翻訳結果の誤訳により齟齬が生じる場合

一つ目の翻訳の際に齟齬が生じる場合とは、翻訳した結果の単語が多義語であった場合などに齟齬が生じる場合である。二つ目の理由とは、発音により正しく伝わらないことや同音異義語により上手く伝わらないケースなどである。三つ目の理由として、翻訳結果は合っているが、話者たちの育った環境や文化によって想起するものが異なるため齟齬が生じる場合である。そして、四つ目の理由として、そもそも翻訳結果が間違っている場合である。

本研究ではこのうち三つ目に述べている話者たちの育った環境や文化によって、想起されるものが異なる場合を多言語コミュニケーションにおける文化差とし、この文化差を検出することを目的とする。

次に、多言語コミュニケーションにおける文化差の具体的な例を紹介する。

“ゴボウ”



“great burdock” “burdock” “greater burdock”



図 2 : “ゴボウ”と“great burdock”, “burdock”, “greater burdock”の例

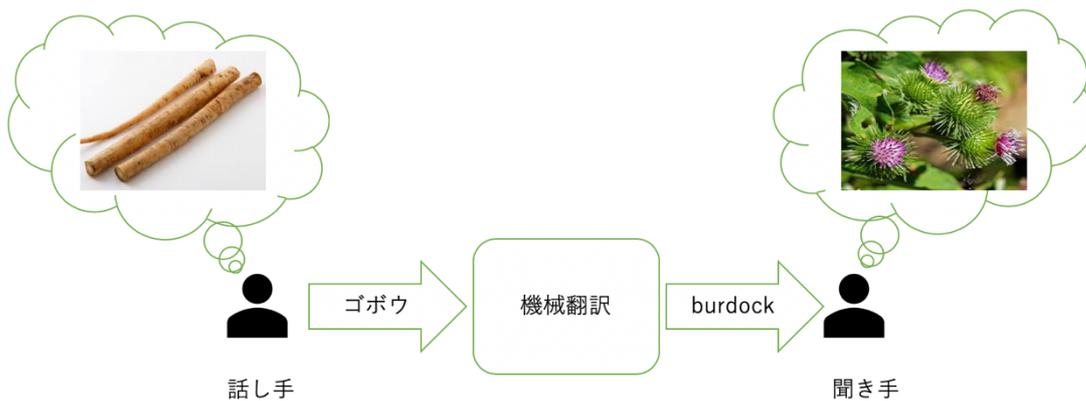


図 1 : 文化差による想起されるイメージの違い

例として、日本でよく食べられる”ゴボウ”が挙げられる。機械翻訳で”ゴボウ”を翻訳すると”great burdock”, “burdock”, “greater burdock”といった翻訳結果が得られる。

日本人の多くは”ゴボウ”と聞くと根っこのような見た目(図 1 の左)の食べ物を想起する。しかし、海外では”great burdock”, “burdock”, “greater burdock”からはイガのついた草木(図 1 の右)を想起することが多い。これらの言葉に対する想起されたイメージはどちらも正しい。なぜなら、どちらも同じ種類の植物であり、日本で言う”ゴボウ”は”great burdock”, “burdock”, “greater burdock”の根っこの部分である。日本では、この部分を食べる文化を持っているため、根っ

この見た目の食べ物(図1の左)を想起してしまう。しかし、海外では、“ゴボウ”を食べる文化がないため、イガがついた草木(図1の右)を想起してしまう。

2.2 関連研究

次に、多言語間のコミュニケーションにおける文化差に関する既存の研究を示す。

まず、**Wikipedia** を利用した文化差を検出する研究が存在する[1]。この研究は吉野らによる研究で事前にアンケートによって、文化差が存在すると人手で判断された言葉に対して、**Wikipedia** のカテゴリなどを用いて文化差を検出できるか確認するものである。加えて、重要度を計算することで、文化差を検出する精度を向上させるものである[2]。

同じく、吉野らは文化差検出手法を用いた多言語間のチャットコミュニケーションアプリを開発する研究を行っている。この研究では、チャットコミュニケーションアプリを作り、吉野らが提案する文化差検出手法が多言語コミュニケーションにおいて、効果的であるかを調べたものである[3]。

Pituxcoosuvarn らによる研究では、日本語の単語とその単語の翻訳結果の英語の単語をキーワードにして、画像検索を行う。そして、検索された画像から画像の特徴ベクトルを抽出し、類似度計算を行うことにより文化差を検出する手法を提案した研究である[4]。

Cho らによる研究で、絵文字を用いて文化差を調査する研究がある。これはネットワークを介して異文化の話者が会話するときに言語を使わない意思伝達のツールとして絵文字に着目した研究である。アメリカと日本の人にアンケートで 120 種類の絵文字についての印象を調査したもので、絵文字は文化によって解釈のされ方が違うことを検出する手法を提案した研究である[5]。

Koda らによる研究では、近年のネットワークコミュニケーションでよく利用されるアバタに着目して、アバタの表情の解釈において文化差が存在するかを分析した研究である。アジアと欧米 8 カ国間のアバタの表情の解釈内容を比較して分析したところ、否定的な内容の解釈には文化差がなかったが、肯定的な内容の解釈には文化差が存在することを発見している[6]。

高崎らによる研究で、多言語の話者に知識を伝える時のコミュニケーションに着目したものがある。この研究では、ベトナムの児童を介して、日本の稲作の専門家からベトナムの農業事業者に農業の知識を伝えるプロジェクトを基にし

ている。知識の伝達の際に、仲介者を挟むことにより、専門家が意図していないニュアンスでベトナムの事業者に伝達される問題があったことを述べている[7].

第3章 文化差の検出手法

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。

3.1 画像特徴量用いた検出方法

文化差を検出するために本研究で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定する手法を提案する。

具体的には、概念辞書で同一概念に紐づけられている日英のそれぞれの単語

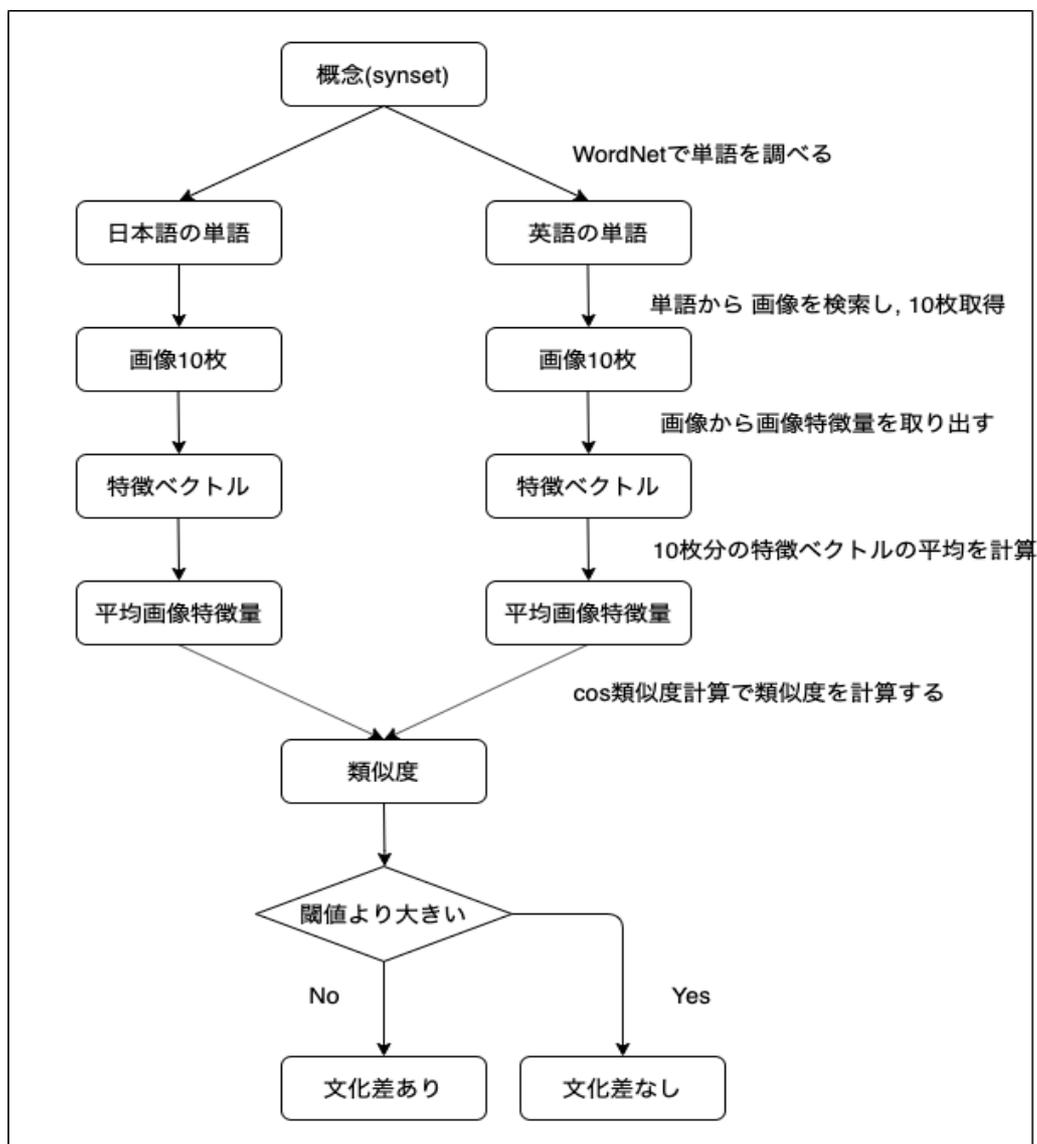


図 3：提案手法のフローチャート

11924445-n	ゴボウ, 牛蒡 burdock, clotbur	ユ を
11924849-n	ゴボウ great burdock, greater burdock, cocklebur, Arctium lappa	ハ 生

図 4 : WordNet の例

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て文化差を検出する。

上記の図 3 は、提案手法をフローチャートにしたものである。図 3 のようにまず、日本語 WordNet から同じ概念と定義された日本語の単語と英語の単語を取得する。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 WordNet の日本語版の概念辞書であり、英語の WordNet をもとに人手で日本語を対応づけて作られたものである。図 4 は、日本語 WordNet の例である。11924445-n という数字は概念を表してる番号である。その横に書いてある”ゴボウ, 牛蒡”というのが同じ概念で対応づけられた日本語の単語であり、その下の”burdock, clotbur”というのが同様の概念に対応づけられた英語の単語である。次に、取得された日本語と英語の単語をそれぞれキーワードにして、画像検索を行う。この際に、それぞれの言語に複数の単語がある場合は、複数の単語をキーワードにして AND 検索をする。例えば、図 4 の”ゴボウ, 牛蒡”であれば、“ゴボウ”と”牛蒡”に分けるのではなく、“ゴボウ 牛蒡”というキーワードで画像検索する。

検索された画像から各言語につき、上位 10 件を取得する。この際に余分な情報をなくして、類似度計算を行いやすくするために、取得した画像をグレースケール化する。グレースケール化した画像から Keras の VGG16 を使って画像の特徴ベクトルを取り出す。VGG16 とは、オックスフォード大学の VGG というチームが開発した、畳み込み 13 層と全結合層 3 層の計 16 層からなる畳み込みニューラルネットワークである。VGG16 は人の判断結果に近い画像認識を行うこ

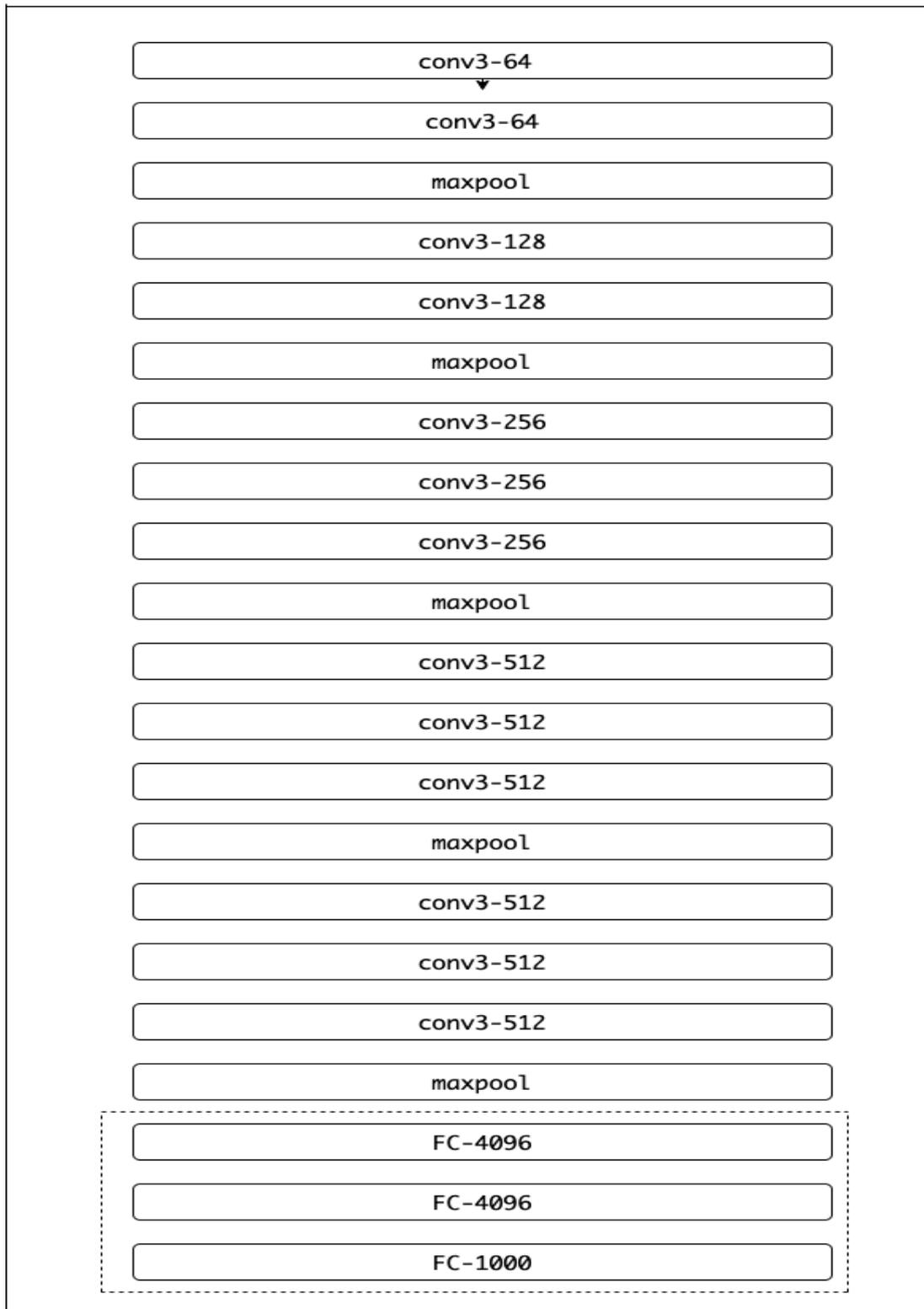


図 5 : VGG16 の構造

とができる. 図 5 は VGG16 の構造を表したものである. 図 5 のうち conv という

畳み込み層では、畳み込みという手法によって、画像から特徴を抽出している。そして、maxpool というプーリング層では、複数の畳み込み層で畳み込んだ特徴量のサイズを半分にする。この畳み込み層 13 層とプーリング層を 5 層重ねた後、最後の FC という結合層では畳み込み層とプーリング層の結果から画像認識のなどを行う。本研究では、画像認識を行うのではなく、画像の特徴量を利用するだけなので結合層を除いた畳み込み層とプーリング層のみを使用する。

畳み込み層を用いて、画像からそれぞれ画像の特徴ベクトルを抽出している。抽出された特徴ベクトルは画像 1 枚につき、4 行 512 次元の配列が 4 組得られる。4×4×512 の形では平均画像特徴量をとることや類似度計算ができないため、特徴ベクトルを 1 行 8192 次元の形に整形する。図 6 はその際の整形を表したものである。そして、画像 10 枚から得られた特徴ベクトルを用いて平均画像特徴量を求める。その後、それぞれの言語から得られた平均画像特徴量を cos 類似度計算によって類似度を計算する。

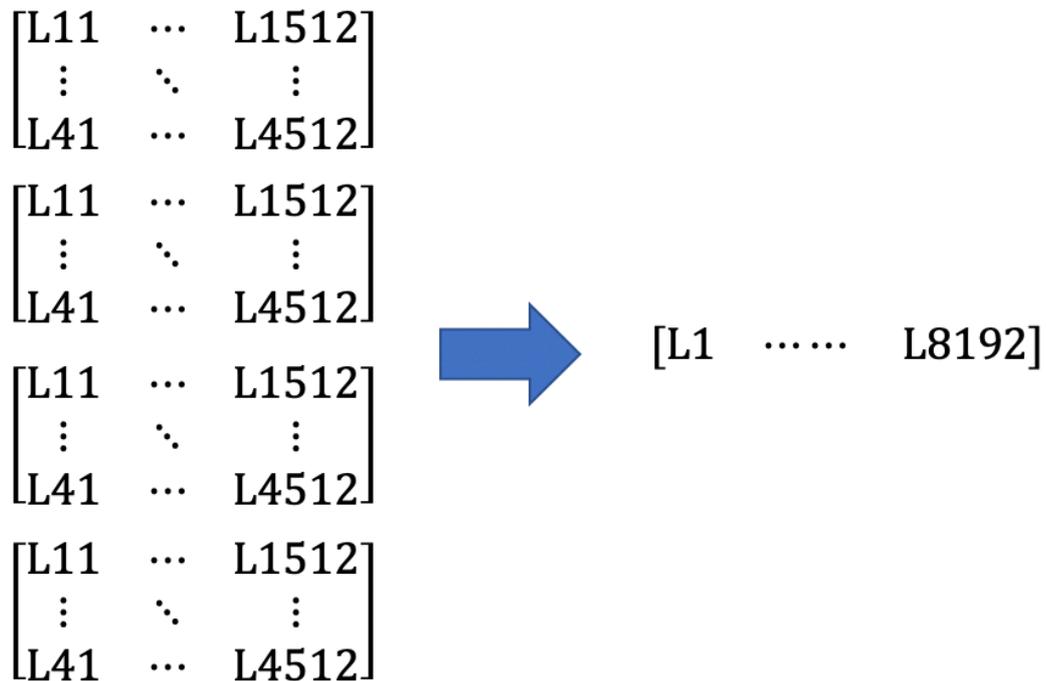


図 6 : 特徴ベクトルの形の整形

$$\text{sim}(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}$$

類似度計算によって得られた類似度と閾値を用いて文化差の有無を評価する。類似度が閾値を下回った場合、文化差が存在し、閾値を上回った場合は、文化差が存在しないと判断する。

第4章 閾値の最適化

本章では提案手法により正確に文化差の有無を検出するための閾値の説明を行う。

4.1 文化差検出の正確さ

まず、提案手法が正確に判断できている場合を下記の表1に示す。

表1で示したように人が文化差の判断をした結果に対して、提案手法が同様の判断をできるかが提案手法による文化差検出の正確さ(Accuracy)である。

提案手法が文化差の有無を正確に検出できていると確認できる場合は2パターン考えられる。

- 人手で文化差があると判断したものを提案手法も文化差があると判断した場合
- 人手で文化差が無いと判断したものを提案手法も文化差が無いと判断した場合

表1でTRUEと表記している箇所が上記の2パターンに該当する。これをもとにAccuracyの式を以下に示す。

$$\text{Accuracy} = \frac{\text{TRUEに該当した個数}}{\text{synsetの個数}} \dots \dots \dots (1)$$

文化差の有無を調査する概念に対して、表1でTRUEと表記されているパターンに該当した概念の個数を数える。AccuracyはTRUEに該当した概念の個数を分子にし、調査した概念(synset)の個数を分母にしたときに算出される確率で表す。

表1：人の判断と提案手法での判断の比較パターン

	人の判断	提案手法	提案手法が正確な判断しているか
文化差の有無	有り	無し	FALSE
	有り	有り	TRUE
	無し	無し	TRUE
	無し	有り	FALSE

4.2 最適な閾値

4.2.1 閾値を最適化する方法

閾値とは文化差の有無を人手で判断した結果に対して、提案手法が正確に文化差の有無を検出するための一定の値とする。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に仮閾値を設ける。概念 800 個分を使って、提案手法で文化差の有無を検出する。仮閾値の値に応じて Accuracy が最も高かった時の仮閾値の値を最適な閾値とする。このとき、各仮閾値において Accuracy を算出するための概念は同じ 800 個を使用する。以下に閾値を最適化する手順

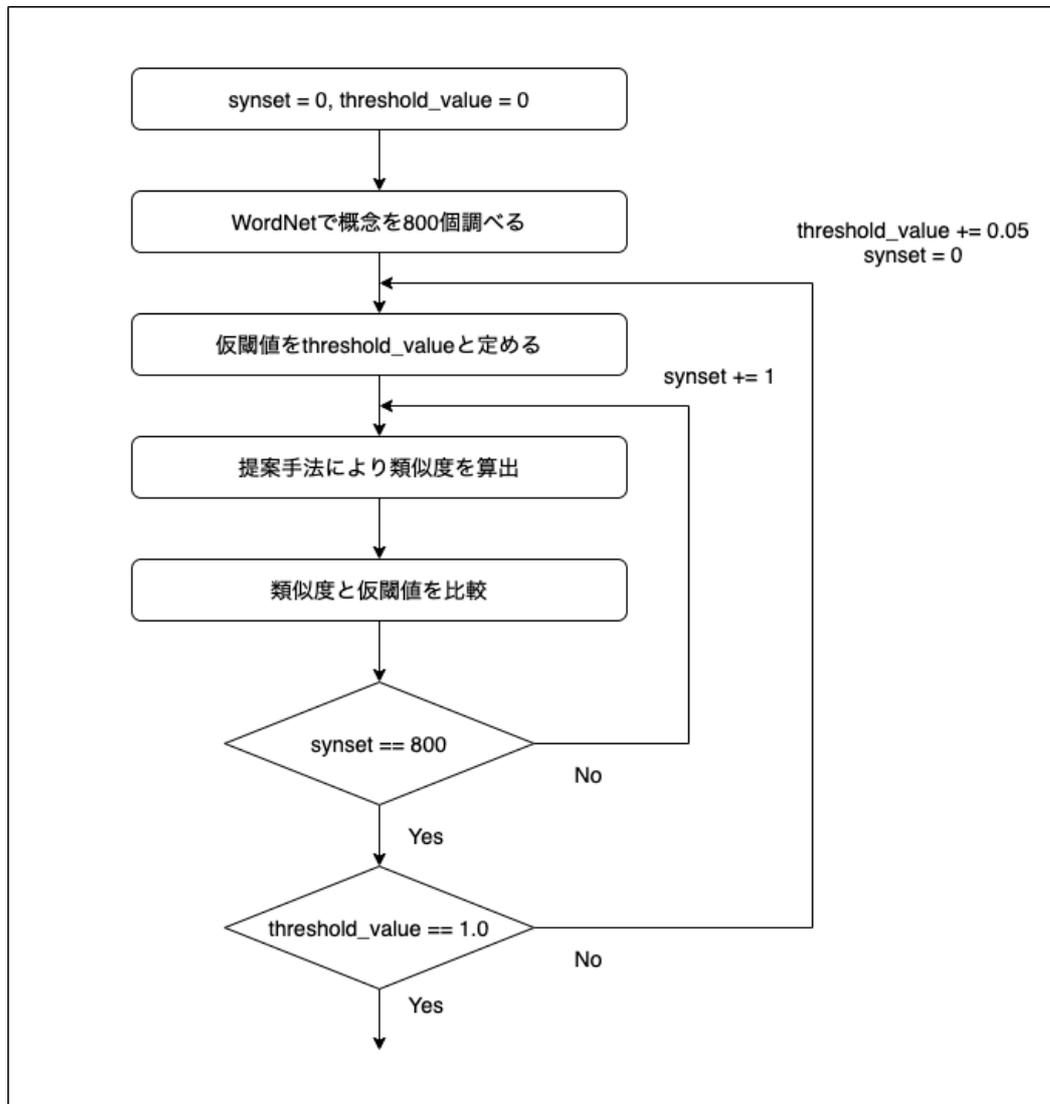


図 7：閾値を最適化する方法

を示す.

閾値の最適化の手順を説明する. まず, 最初に $\text{synset} = 0$, $\text{threshold_value} = 0.0$ とする. synset とは概念の個数のことであり, threshold_value とは仮閾値の値である. 今回, 仮閾値は $0.0 \sim 1.0$ までの範囲で 0.05 ごとに検証する. 次に, 日本語 WordNet で概念を 800 個用意する. 次に, threshold_value を仮閾値と定める. 提案手法により synset の類似度を算出し, 仮閾値と比較し, 文化差の有無を調べる. この手順で用意した概念 800 個になるまで繰り返し, Accuracy を求める. 次に, 仮閾値が 1.0 になるまで 0.05 ごとに同じ 800 個の概念を使用して, 同様の手順を行う. 仮閾値の値ごとに求めた Accuracy を比較し, 最も Accuracy が高かった時の仮閾値を最適な閾値として定める.

4.2.2 人手での判断

人手で文化差の有無を判断した場合, 人によって, 文化差の感じ方が異なるため, 本研究では概念一つにつき, 五人に文化差の有無を判断してもらった. このため, 上記の表 1 において, 人の判断は 5 人分の判断があるため, 多数決し, 人

...

A (1) ラジオボタン



キーワード: 壁

A B 両方 該当なし

図 8 : アンケート(画像 A)

B (1)



図 10 : アンケート(画像 B)

アンケート1~25

今から25組の画像群のセットを見てもらいます。
キーワードを見て、“A”と“B”のどちらの画像群がより連想できるか選んでください。
もし、両方とも連想できた場合は“両方”を選び、どちらも連想できない場合は“該当なし”を選んでください。
※()内の数字は質問の番号です

図 9 : アンケート(文言)

の判断を一つに定めている。

上記の図 8~10 は人手で文化差を判断してもらうときに用いたアンケートである。アンケートは概念 1000 個から 25 個ずつに区切って、40 個のアンケートを作成した。図 10 のようにアンケートでは、キーワードと画像群 A と画像群 B の三つの情報を解答者に与えている。図 8 と図 9 には A と B の画像群と選択の択を載せている。択には、“A”、“B”、“両方”、“該当なし”の四つの選択肢があり、ラジオボタンを使用し、択の選択の重複を不可能にしている。

このアンケートでは、それぞれの画像群を見て、キーワードから連想できる画像群はどちらかを選んでもらう。A の画像群が連想できた場合は“A”、B の画像群が連想できる場合は“B”、両方とも連想できる場合は“両方”を選択してもらう。また、どちらの画像群もキーワードから連想できない場合は“該当なし”を選択してもらう。

次に、これらのアンケート結果から多数決を行い、人手で文化差を判断した結果とする。表 2 はアンケート結果と多数決の例である。表 2 の行は一つの

synsetを表している。そして、アンケート結果の列はそれぞれ同一の解答者によ

表2：アンケート結果の多数決の例

類似度	アンケート結果					多数決の結果
0.4421	A	A	両方	両方	両方	両方
0.2753	A	両方	A	両方	両方	両方
0.4329	A	A	A	両方	両方	AB
0.3221	A	A	A	両方	両方	AB
0.7506	B	両方	両方	両方	両方	両方
0.7325	A	両方	両方	両方	両方	両方
0.4905	A	A	A	B	両方	AB
0.8203	両方	A	両方	両方	両方	両方
0.8069	A	B	該当なし	両方	両方	両方
0.6575	両方	A	両方	両方	両方	両方
0.6163	B	両方	両方	両方	両方	両方
0.778	両方	両方	両方	両方	両方	両方
0.3419	A	A	A	両方	両方	AB
0.5665	A	両方	両方	両方	両方	両方
0.6933	A	A	A	両方	A	AB
0.5016	A	A	該当なし	A	A	AB
0.5694	B	B	B	B	両方	AB
0.5441	B	B	両方	両方	両方	両方
0.5497	B	両方	両方	両方	両方	両方
0.4749	A	A	A	A	A	AB
0.7492	A	両方	両方	両方	B	両方
0.825	両方	B	B	両方	両方	両方
0.4359	A	A	両方	A	両方	AB

る解答である。

多数決では”両方”や”該当なし”が最も多い場合はそのまま”両方”と”該当なし”になり、“A”や”B”が最も多い場合は、“AB”という結果になっている。本研究では、文化差を検出したいため、“両方”の場合は人が見て文化差がないと判断したと考える。また、“AB”となる時は”A”か”B”のどちらか一方しか連想できなかった

ということであるため、文化差があると判断できる。また、“該当なし”が最も多かった場合は画像検索の結果が悪いなど様々な理由が考えられるため、文化差の有無の判断に加えない。

4.2.3 人手での判断と提案手法の比較

多数決から得られた人出での判断の結果と提案手法の結果を比較する。

4.2.1 節で述べた手法を用いて、概念 800 個を用いて提案手法を検証し、アンケートの結果と比較して最適な閾値を調査した結果が下の図 11 である。

800 個の概念を検証して、(1)の式に当てはめた時、概念の数は 800 個を使用したため、分母は 800 となる。上記の値はそれぞれ人手での判断と提案手法の判断を比較した場合に、文化差検出できた数である。この値を(1)の式の分子にあてはめて、それぞれ Accuracy を計算した場合、仮閾値を 0.4 にした時が 77.5% で最も高かった。このため、文化差を検出するための最適な閾値は 0.4 である。

また、図 11 の仮閾値を 0.0 にした時、提案手法が正確に検出できた数は 800 個中 560 個であった。同じく、仮閾値を 1.0 にした時、提案手法が正確に検出できた数は 800 個中 212 個であった。仮閾値を 0.0 にしたときとは提案手法は全て文化差なしと判定しているということである。仮閾値を 1.0 にしたときとは提

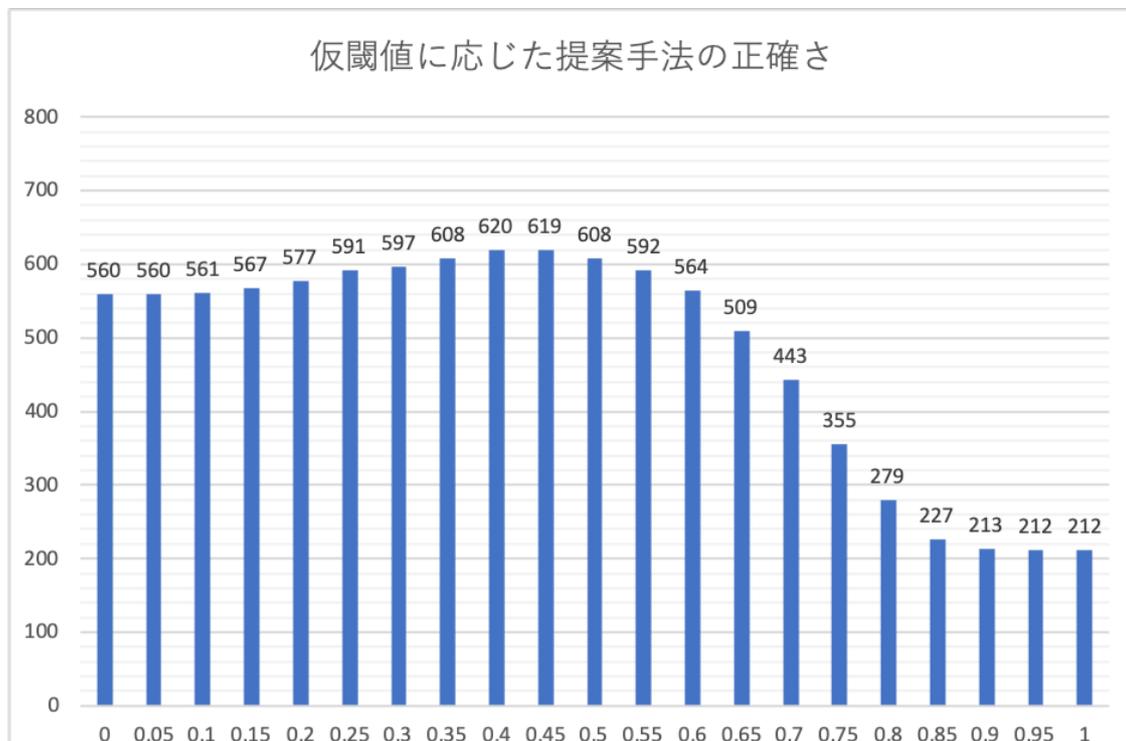


図 11：仮閾値ごとの検出数

案手法は全て文化差ありと判断しているということである. このことから, 使った概念 **800** 個のうち **560** 個は実際に文化差がないということであり, **212** 個は, 実際に文化差があるということが分かった.

第5章 評価

本章では、最適な閾値を用いて、文化差を検出できるかの評価を行う。4章での検証から最適な閾値は0.4とわかった。これを閾値として定めて、提案手法を検証する。検証する概念の数は200個でこれらの概念は、閾値を最適化の際に使用された概念とは別の概念を使う。アンケートは最適化する時と同様に5人に行い、多数決によって人の判断を定めた。概念は200個を使用して検証を行なった。結果は、200個のうち154個が人の判断と一致していた。Accuracyは77%で人が文化差を判断した場合に比べて、提案手法は8割近く正確に判断することができた。

本研究の提案手法では、画像をもとに文化差を検出しているため、具体的な形のあるものを表す言葉・概念は正確に文化差の判定が可能である。しかし、形のないものを表す言葉・概念の文化差を判定する場合は、具体的な形がないため様々な画像が取得される。結果、文化差の判定において正確に判断できない場合がある。

また、WordNet から日英の単語をもとに画像検索するとき、取得した単語と同じタイトルの本・映画などが存在すると画像検索の結果が映画の写真や本の表紙が多く取得されるので、正確に文化差を検出できない場合がある。

これらのように本研究の提案手法では、よくも悪くも画像検索で取得された

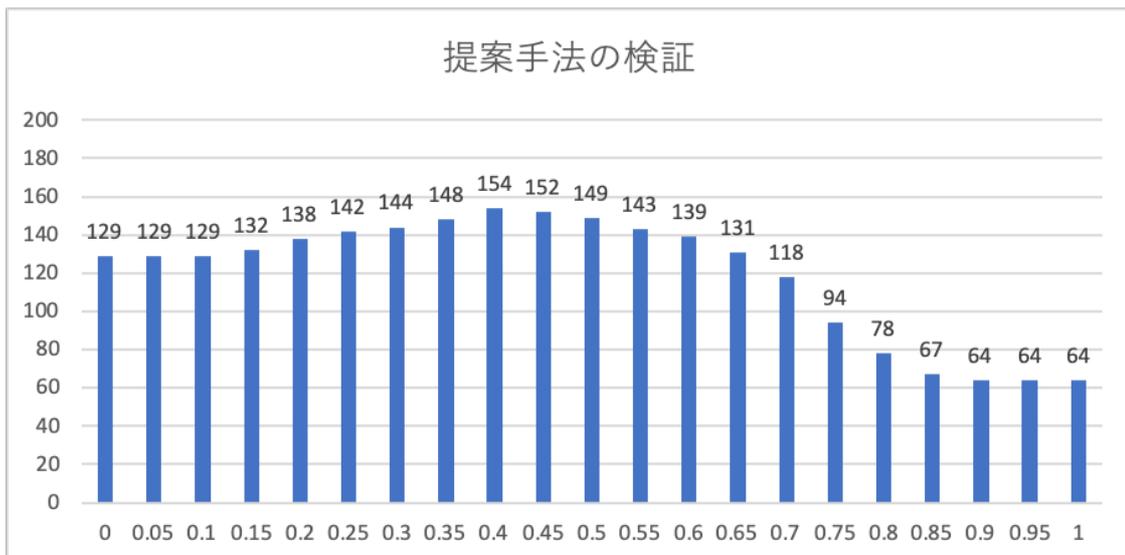


図 12 : 同定した閾値の検証

画像に依存した結果となる。今後は画像以外から文化差を判断する手法と複合させ、複数の視点から文化差を検出することが今後の課題であると考えられる。

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、本研究では画像から得られる特徴ベクトルを用いることで文化差検出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。

最適な閾値の同定

閾値を最適化するために仮の閾値を定める。800 個の概念を用いて、それぞれの仮閾値で提案手法の正確さ(Accuracy)を調べた。仮閾値を 0.4 とし、定めた時に提案手法の正確さ(Accuracy)が 77.5%で最も高かった。この結果から文化差を検出のための最適な閾値は 0.4 であると分かった。

閾値を使った文化差検出精度の検証

Accuracy が最も最大化する閾値を求めることによって、最適な閾値は 0.4 であると分かった。定められた最適な閾値を検証するために、最適な閾値を求める時に用意した概念とは異なる概念を 200 個用意し、閾値の文化差検出精度を検証した。検証の結果、用意した 200 個の概念でも Accuracy は 77%で閾値は 0.4 が最も高かった。このことから、文化差検出のための最適な閾値は 0.4 であると確認できる。

また、検証の結果から 8 割近い精度で人が文化差の有無を判断した場合と同様の判断をすることができた。本研究の提案手法は文化差検出において、有効であると言える。

文化差を検出するために、画像の特徴ベクトルの類似度を計算することで、単語から想起されるイメージをもとに文化差を検出することができた。画像を使うことは言葉だけでは伝わらない情報やイメージを踏まえて文化差が存在することを検出できるため、有効な方法であると考えられる。

また、文化差が存在すると判断する基準として、閾値 0.4 を定められたことにより、多くの文化差を検出することができたため、提案手法が文化差検出において有効であることを示した。

一方で課題として、本研究の提案手法では、画像検索で取得された画像に依存した結果となるため、他の画像以外の要素を含めて、文化差を判断する必要があると考えられる。

今後は、本研究の提案手法である画像を用いた手法に加えて、画像以外の要素から文化差を検出する手法を複合することで多角的な視点から文化差検出で

きるようにすることが必要であると考えられる。

謝辞

本研究を行うにあたり、ご指導くださりました村上陽平准教授に深く感謝申し上げます。また、アンケート等でご協力くださいました皆様に感謝を申し上げます。

参考文献

- [1] 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)
- [2] 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
- [3] 諏訪智大, 宮部真衣, 吉野孝:文化差検出手法を用いた異文化間チャットコミュニケーションシステムの開発と評価(人工知能と知識処理), 電子情報通信学会技術研究報告書
- [4] Mondheera Pituxcoosuvann, Donghui Lin, and Toru Ishida:A Method for Automated Detection of Cultural Difference Based on Image Similarity:Collaboration Technologies and Social Computing, 25th International Conference, CRIWG+CollabTech 2019, September, 4-6, pp.129-143.
- [5] CHO Heeryon, 稲葉利江子, 石田享, 高崎俊之, 森由美子: 絵文字コミュニケーションにおけるセマンティクス, 情報処理学会2006-ICS-145(1)
- [6] 神田智子, 石田享:アバタ表情解釈の異文化間比較, 情報処理学会論文誌, Vol47 No3, 731-738(Mar 2006)
- [7] 喜多香織, 高崎俊之, 林冬恵, 中島悠, 石田享:多言語知識コミュニケーションの分析と改善, 情報処理学会 第75回全国大会講演論文集pp169-170 (2013)
- [8] Karen Simonyan, Andrew Zisserman:Very Deep Convolutional Networks for Large-Scale Image Recognition,Computer Vision and Pattern Recognition
- [9] Mondheera Pituxcoosuvann, Toru Ishida, Naomi Yamashita, Toshiyuki Takasaki, and Yumiko Mori: Machine Translation Usage in a Children's Workshop

付録：ソースコード

提案手法を実装するにおいて、プログラムの実装は Python で行った。

ソースコード 1 :image_feature_vector_improve.py

```
from keras.applications import VGG16
from keras.applications.vgg16 import VGG16, preprocess_input,
decode_predictions
import os, zipfile, requests, argparse, json, urllib, urllib.request, collections
import numpy as np
from keras.preprocessing.image import ImageDataGenerator, load_img
from keras import models, layers, optimizers
from PIL import Image
from bs4 import BeautifulSoup
import openpyxl

def write_result(classes,result):
    filename='../image_feature_vector.xlsx'
    book = openpyxl.load_workbook(filename)
    sheet = book.worksheets[2]
    max_row = sheet.max_row
    sheet.cell(row = max_row + 1, column=2).value = classes[0]
    sheet.cell(row = max_row + 1, column=3).value = classes[1]
    sheet.cell(row = max_row + 1, column=4).value = result
    # 保存
    book.save(filename)
    # 終了
    book.close()

def cos_sim(v1, v2):
    return np.dot(v1, v2) / (np.linalg.norm(v1) * np.linalg.norm(v2))
```

```

def imread_japanese(filename): ##画像を赤一色に単色化
    gamma22LUT = [pow(x/255.0, 2.2)*255 for x in range(256)] * 3
    gamma045LUT = [pow(x/255.0, 1.0/2.2)*255 for x in range(256)]
    #img = Image.open(filename)
    #img = Image.open("data/TrainData/test1/" + filename)
    img = Image.open("dataset/trainData/downloads/魔物 御化け 妖精 変化 お化け 化生/" + filename)
    img_rgb = img.convert("RGB") # any format gym mat to RGB
    img_rgbL = img_rgb.point(gamma22LUT)
    img_grayL = img_rgbL.convert("L") # RGB to L(grayscale)
    img_gray = img_grayL.point(gamma045LUT)
    #pil_img.save('data/TrainData/redscale/test/' + filename)
    img_gray.save('dataset/trainData/grayscale/魔物 御化け 妖精 変化 お化け 化生/' + filename)

```

```

def imread_english(filename): ##画像を赤一色に単色化
    gamma22LUT = [pow(x/255.0, 2.2)*255 for x in range(256)] * 3
    gamma045LUT = [pow(x/255.0, 1.0/2.2)*255 for x in range(256)]
    #img = Image.open(filename)
    #img = Image.open("data/TrainData/test1/" + filename)
    img = Image.open("dataset/trainData/downloads/hob hobgoblin goblin/" + filename)
    img_rgb = img.convert("RGB") # any format gym mat to RGB
    img_rgbL = img_rgb.point(gamma22LUT)
    img_grayL = img_rgbL.convert("L") # RGB to L(grayscale)
    img_gray = img_grayL.point(gamma045LUT)
    #pil_img.save('data/TrainData/redscale/test/' + filename)
    img_gray.save('dataset/trainData/grayscale/hob hobgoblin goblin/' + filename)

```

```
new_eng_dir_path = 'dataset/trainData/grayscale/hob hobgoblin goblin'
```

```
new_jap_dir_path = 'dataset/trainData/grayscale/魔物 御化け 妖精 変化
```

お化け 化生'

```
jap_dir_exist = os.path.exists(new_jap_dir_path)
eng_dir_exist = os.path.exists(new_eng_dir_path)
if jap_dir_exist == False:
    os.mkdir(new_jap_dir_path)
if eng_dir_exist == False:
    os.mkdir(new_eng_dir_path)
    #return arr.flatten()

filenames_japanese = os.listdir('dataset/trainData/downloads/魔物 御化け
妖精 変化 お化け 化生')
filenames_english = os.listdir('dataset/trainData/downloads/hob hobgoblin
goblin')
#filenames = os.listdir('data/TrainData/test1')
X = np.array([imread_japanese(filename_jap) for filename_jap in
filenames_japanese])
Y = np.array([imread_english(filename_eng) for filename_eng in
filenames_english])
model = VGG16(include_top=False, weights='imagenet')
model.summary()
classes = {'魔物 御化け 妖精 変化 お化け 化生','hob hobgoblin goblin'}
batch_size = 32

image_data_generator = ImageDataGenerator() #rescale=1.0/255)
train_data = image_data_generator.flow_from_directory(
    'dataset/trainData/grayscale',
    #target_size=(224, 224),
    target_size=(150, 150),
    batch_size = batch_size,
    class_mode='categorical',
    classes = classes,
    shuffle=False
)
```

```

n_train_samples = 1
# train_features = []
# feature_vector_average = np.zeros(4*4*512)
counter = 0
label_list = []
for input, label in train_data:
    counter += 1
    label_list.append(label)
    if batch_size * counter >= 20:
        break
#print(label_list)
#c = count(label_list)
#c = collections.Counter(tuple(count_label) for count_label in label_list)
str_label_list = [str(i) for i in label_list[0]]
c = collections.Counter(str_label_list)
train_features_dict = {i:[] for i in c.keys()}
feature_vector_average_dict = {i:np.zeros(4*4*512) for i in c.keys()}

#norm = []
bottleneck_feature_train = model.predict_generator(train_data,
n_train_samples, verbose=1)
#print(type(bottleneck_feature_train))
#print(bottleneck_feature_train.shape)
"""results = decode_predictions(bottleneck_feature_train, top=5)[0] #上位
5個出力
#print(results)
for result in results:
    print(result)
exit()"""

for i in range(len(bottleneck_feature_train)):

```

```

        #         train_features.append(np.reshape(bottleneck_feature_train[i],
4*4*512))
        #print(i)

train_features_dict[str_label_list[i]].append(np.reshape(bottleneck_feature_train[i], 4*4*512))

#train_features_dict[str_label_list[i]].append(np.reshape(bottleneck_feature_train[i], 1000))
        feature_vector_average_dict[str_label_list[i]] +=
train_features_dict[str_label_list[i]][-1]

num = 1
print(classes)
for i in feature_vector_average_dict:
    print(str(num)+".画像特徴量の総和\n",feature_vector_average_dict[i])
    #print(c[i])
    feature_vector_average_dict[i] /= c[i]
    print(str(num)+".特徴量の平均値\n",feature_vector_average_dict[i])
    num += 1
# feature_vector_average /= len(bottleneck_feature_train)
# print("特徴量の平均値\n",feature_vector_average)
#print(feature_vector_average_dict["[1. 0.]"])
cos_similarity = cos_sim(feature_vector_average_dict["[1. 0.]"],
feature_vector_average_dict["[0. 1.]"])
classes = list(classes)
write_result(classes,cos_similarity)
print("cos 類似度は : ",cos_similarity)

```

ソースコード 2 :Accuracy_cal.py

```
import openpyxl
```

```

threshold_value_list = [0.0, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45,
0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95, 1.0]
filename='../image_feature_vector.xlsx'
book = openpyxl.load_workbook(filename)
Cul_Def_count = []
for threshold_value in threshold_value_list:
    #cul_not_counter = 0
    cul_counter = 0
    #print("閾値は", threshold_value)
    for sheet_number in range(4):
        sheet = book.worksheets[sheet_number]
        #print("シートナンバーは", sheet_number)
        min_row = sheet.min_row
        max_row = sheet.max_row
        #print("一番上は", min_row)
        #print("一番下は", max_row)
        for row in range(min_row + 1, max_row + 1, 1):
            #print(row)
            sim_result = sheet.cell(row = row, column=4).value
            if sim_result == None:
                print("空欄",row)
            elif sim_result >= threshold_value:
                both_num = sheet.cell(row = row, column=11).value
                if both_num == "両方":
                    cul_counter += 1
                    #print(cul_not_counter)
                    #B_num = sheet.cell(row = row, column=6).value
            elif sim_result < threshold_value:
                AB_num = sheet.cell(row = row, column=11).value
                if AB_num == "AB":
                    cul_counter += 1
                    #print(cul_counter)

```

```

        #print(sheet.cell(row = row, column=4).value)
        #exit()
print("閾値", threshold_value,"の文化差の検出数は", cul_counter)
Cul_Def_count.append(cul_counter)
# 保存
book.save(filename)
# 終了
book.close()

```

ソースコード 3 :Majority_decision.py

```

import openpyxl

filename='../image_feature_vector.xlsx'
book = openpyxl.load_workbook(filename)
#Cul_Def_count = []
for sheet_number in range(5):
    sheet = book.worksheets[sheet_number]
    #print("シートナンバーは", sheet_number)
    min_row = sheet.min_row
    max_row = sheet.max_row
    #print("一番上は", min_row)
    #print("一番下は", max_row)
    for row in range(min_row, max_row + 1, 1):
        both_counter = 0
        AB_counter = 0
        none_counter = 0
        for ans_col in range(5):
            result_num = sheet.cell(row = row, column=ans_col+5).value
            if result_num == "両方":
                both_counter += 1
            elif result_num == "A":
                AB_counter += 1

```

```

elif result_num == "B":
    AB_counter += 1
elif result_num == "該当なし":
    none_counter += 1
if both_counter >= AB_counter:
    if both_counter >= none_counter:
        sheet.cell(row = row, column=11).value = "両方"
    elif both_counter < none_counter:
        sheet.cell(row = row, column=11).value = "該当なし"
elif both_counter < AB_counter:
    if AB_counter >= none_counter:
        sheet.cell(row = row, column=11).value = "AB"
    elif AB_counter < none_counter:
        sheet.cell(row = row, column=11).value = "該当なし"

# 保存
book.save(filename)
# 終了
book.close()
#print(Cul_Def_count)

```