令和6年 1月31日

修士学位論文

Biterm Topic Modelを用いた SNS上の商品レビューに対する 関連性評価

指導教員：相馬 隆郎 准教授

東京都立大学大学院

電子情報システム工学域

学修番号：22861651

氏名：西原涼介

論文要旨

近年, Amazon や楽天市場などの大手 EC サイトをはじめ, 数多くの EC サイトが普及しその利用者数は急速に増加している. EC サイトの利用者のうち, 約 70%が商品を購入する際に EC サイトのレビューを参考にしているとされ, その中でも特にレビューの信頼性を重要視している人が多いことが明らかになっている. また, 多くの企業にとっては EC サイトのレビューからユーザーの嗜好や意見を分析し，マーケティングに活用することが重要な課題となっている．そのため，EC サイトのレビューの信頼性や有用性を評価する評判分析や口コミ分析, レビューを様々なトピックに分類する文書分類に関する研究が多く行われている. 例えば, Amazon の商品レビューを機械学習を用いて分析し, 他のユーザーにとって参考になる順に並び替えるシステムの構築及びその評価に関する研究や, グルメサイトの口コミにおいて実名・顕名・匿名の違いがレビューの信頼性に与える影響についての研究などが挙げられる. また近年では, 従来の EC サイトや商品の Web ページだけではなく, YouTube のような動画投稿サイトや X(旧 Twitter) や Instagram などの SNS を自社製品やサービスの宣伝の場として利用している企業が増加している. これに伴い, 商品を購入する際に SNS や YouTube 上でその商品を宣伝している投稿を参考にしている人も増加している. そのため, SNS や YouTube 上の広告に対するユーザーのコメントも, 他のユーザーが商品の購入を検討する際の重要な判断材料になり得ると考えられる. つまり, SNS や YouTube 上での商品の宣伝に対するコメントは, EC サイトの商品レビューと同等の機能を持ち, その信頼性や有用性が重要視されるため, 同じく評判分析や文書分類の研究の対象になると考えられる. 例えば, SNS や YouTube は商品レビューのページとは異なり, 誰でも気軽にコメントを投稿できたり, その投稿内容も自由という特性上, 商品やサービスに関係ないコメントが多数存在するという問題がある.

そこで, 本研究では分析対象を YouTube 上で自社製品やサービスを宣伝している動画に対するユーザーのコメントとし, その動画で宣伝している商品やサービスに対しての関連性が高いコメントを抽出するシステムの作成, 及び作成したシステムの人手に対する精度の検証を目的としている. これにより, EC サイトの商品レビューと同様に他のユーザーが商品の購入を検討する際の重要な判断材料として扱うことが可能になると考えられる.

この目的の実現のため, 本研究ではトピックモデルの一種である Biterm Topic Model(BTM) を用いたトピック抽出, 及び各トピックの生成確率上位の単語を利用した手法を提案する. BTM は他のトピックモデルとは異なり, 文書全体のバイターム (2 単語の対) の共起性を利用してトピックを学習するため, YouTube のコメントのように一文が比較的短い場合でも適切にトピックを推定することが可能であると考えられる. 以下に本研究の提案手法の概要を述べる. はじめに, 商品やサービスの宣伝を行っている YouTube 動画に対するコメントを YouTube Data API を用いて取得し, クリーニング処理や分かち書きなどの前処理を施す. 次に, 前処理をしたコメント集合に BTM を適用し, 潜在的なトピックの推定と各トピックにおける生成確率上位の単語を抽出する. そして, 大規模言語モデルである GPT-4 を用いて, 各トピックから抽出した単語を基に文章を自動的に生成する. ここで生成した文章は BTM によって推定したトピックに出現しやすい単語を用いているため, 各トピックの代表的な文章であるという仮説を立てることができる. つまり, 生成した文章は元のコメント集合に対して代表的であり, 動画内容に関連している文章であるという仮説が立つ. その仮説を基に, 各トピックごとに生成した文章と元のコメント文との文章間の類似度を計算し並び替えることで, 本研究の目的である動画に対して関連性が高い文章の抽出が実現できると考えた. また, 本仮説の妥当性, 及びシステムの精度を検証するために, 人手でアノテーションしたデータセットとの比較分析を行った.

実際の YouTube コメントを本手法に適用し実験を行なった結果, BTM によって商品に関連し

ているトピック及び単語を適切に抽出することができた. また, 人手でアノテーションしたデータと本手法で類似度計算を行ったデータから Confusion Matrix を算出し, 様々な指標で本手法の性能を評価した. その結果, Precision(適合率) が 0.7 程度の値を示したことから, 本手法により商品との関連性が高いと予測したコメントのうち約 7 割が人手の評価に対して正しく予測できていることが分かった. この結果より, 本研究で提案した SNS や YouTube コメントに対する新たな分析手法の有効性を示すことができた.

目 次

第 I 部
はじめに

1　研究背景

近年, Amazon や楽天市場などの大手 EC サイトをはじめ, 数多くの EC サイトが普及し, その利用者も急増している. そして, 商品を購入する際に EC サイトのレビューを参考にしている利用者の割合は約 70%と言われていて, その中でもレビューの信頼性を重要視している人が多いことが明らかになっている. また, 多くの企業にとって, EC サイトのレビューからユーザーの嗜好や意見を分析し，マーケティングに活用することが重要な課題となっている. そのため，EC サイトのレビューの信頼性や参考になるかどうかを評価する評判分析や口コミ分析, レビューを様々なトピックに分類する文書分類に関する研究が多く行われている. 例えば, 関連研究の項で詳しく紹介する「機械学習を用いた自然言語処理による商品レビューの評価」[1] では, Amazon の商品レビューを機械学習を用いて参考になる順に並びかえるシステムの構築, 及びその評価に関する研究を行っている。また近年では, 従来の EC サイトや商品の Web ページ以外にも, YouTube のような動画投稿サイトや X(旧 Twitter) や Instagram などの SNS で自社製品・サービスの宣伝を行う企業が増えてきている. それにつれて, 商品を購入する際に SNS や YouTube 上でその商品を宣伝している投稿を参考にしている人も増加している. そのため, SNS や YouTube 上の広告に対するユーザーのコメントも, 他のユーザーが商品の購入を検討する際の重要な判断材料になり得ると考えられる. つまり, SNS や YouTube 上での商品の宣伝に対するコメントは, EC サイトのレビューと同等の機能を持ち, その信頼性や参考になるかどうかが重要になるため, 評判分析や文書分類の研究の対象になると考えられる. ここで, SNS や YouTube は商品レビューのページとは異なり, 誰でも気軽にコメントを投稿できたり, その投稿内容も自由という特性上, 商品やサービスに関係ないコメントが多数存在する.

そこで, 本研究では分析対象を YouTube 上で自社製品やサービスを宣伝している動画に対するユーザーのコメントとし, トピックモデルの一種である Biterm Topic Model による商品に関するトピック抽出を用いて, その動画に対するユーザーのコメントから, 宣伝している商品やサービスに対して関連性が高いコメントを抽出するシステムの作成, 及び作成したシステムの人手に対する精度の検証を行った.

本論文の第 I 部では, EC サイトのレビューにおける評判分析やトピックモデルを用いた文書分類に関する関連研究の紹介, また本研究の研究目的を明確に説明する. 第 II 部では, 本研究で用いる二つのトピックモデルの説明, 及び提案手法のシステムや実装方法について説明する. 第III部では, 実際の YouTube 上の動画に対するコメントを用いた実験結果を述べる. 第IV部では, 実験結果をもとに考察した提案手法の有効性や将来性について述べる.

2　関連研究

本研究を進めるにあたり, 研究テーマの方向性決めや研究課題の発見, 及び本研究で用いている技術に関して参考にした論文を 4 つ紹介する.

2.1　機械学習を用いた自然言語処理による商品レビューの評価 (市川 (2021)[1])

市川 (2021) は, ユーザーが商品レビューを読んで参考になったかどうかを評価する機能が備わっていない EC サイトの場合に, 数多くあるレビューから参考になる情報を探す必要がある問題に着目し, 機械学習を用いた自然言語処理の手法で分析, 評価を行い, レビューを参考になる順番に並び替えるシステムの構築を目的としている. そして並び替えた順番が正しいかどうかを評価するために, クイックソートを利用した新しい評価法である QE 法を提案している.

図 1 は市川 (2021) で提案されている, レビューを参考になる順番に並び替えるシステムの概要図である. はじめに, インターネット経由で Amazon の商品レビューのデータ取得し, 学習用データと評価データに分ける. 学習段階では, レビュー文章の正規化や各前処理を施し, 教師データとして準備する. この研究では, 全角数字やアルファベットを半角に変換したり, 数字は全て 0 に置換, アルファベットは全て小文字に変換などの正規化を行っている. また, 日本語形態素解析システムである MeCab を用いて形態素解析を行い, 品詞ごとに “_” で分割する. その後, 活用語の原型への変換, 及びストップワード除去を行っている. 例えば,「ロボットは２４時間働けるのでＡＩに仕事をとられる.」という文章の場合, 正規化と前処理を施すことで,「ロボット_0_働ける_ai_仕事_とる.」となる. この一連の処理を学習用データに施した後, 機械学習の際に用いる素性の抽出を行う. 市川 (2021) はレビューを参考になる順序に並べ替えることが目的のため, 素性には単語

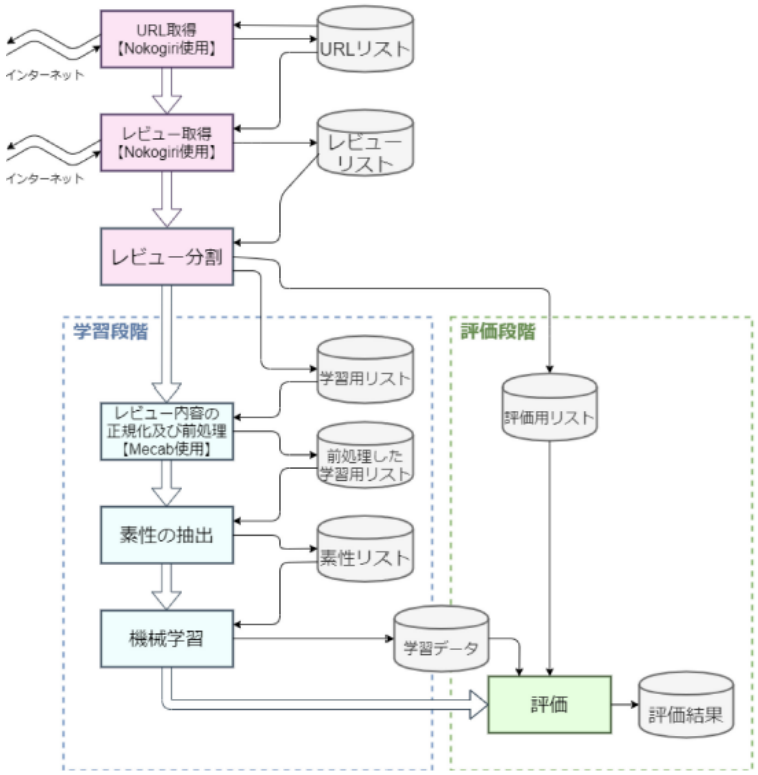

図 1: システム概要図 (出典：市川 [1] p.85)

の出現頻度を用いている. 目的変数をレビューが参考になる確率 P とし, 抽出した素性を用いてロジスティック回帰により学習する. ロジスティック回帰のモデル式は式 (1) で示される. θ_i は素性の重み, N は素性の数を表している.

$$P = \frac{1}{1 + exp(\theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_N x_N)} \tag{1}$$

次に学習したモデルを用いて, 評価用データに対して実験を行ない, 提案システムの精度を検証している. この研究の提案システムの精度の評価は, 実際の商品ページのレビューの並び順との一致率で評価している. 正解の並び方を L_R, 提案システムによる並べ方を L_P としたとき, それぞれの要素の一致率を P_{match} としている. 例えば, 以下の並び方のとき, $P_{match} = 100\%$ となり最も良い結果となる.

$$L_R : \{1, 2, 3, 4, 5\}$$
$$L_P : \{1, 2, 3, 4, 5\}$$

しかし, 以下のように並び方の評価としては良い結果と言える場合でも, 5 件のレビュー中 1 件のみ一致していることになり, $P_{match} = 20\%$ と低い結果になる.

$$L_R : \{1, 2, 3, 4, 5\}$$
$$L_P : \{4, 1, 2, 3, 5\}$$

このように正しい評価が行えない場合を解決するため, この研究ではクイックソートを利用した新しい評価法の QE 法 (Quicksort Evaluation method) を提案している. QE 法ではピボットを中央値とし, 昇順にするために要素を入れ替えた回数 S_{count} と, 要素数における最大の入れ替え回数 S_{max} を用いた式 (2) により, 評価値 P_{QE} を求めている. なお, S_{max} は全ての要素が逆順の場合にクイックソートで昇順に入れ替えた回数である.

$$P_{QE} = 1 - \frac{S_{count}}{S_{max}} \tag{2}$$

実際の商品レビュー 52,403 件を取得し, そのうち 51,403 件を学習用データ, 1,000 件を評価用データに分けて実験を行い, 提案システムの精度を評価した結果を表 1 に示している. ここで, 登場回数 F とは学習の素性とするか決定するための単語の出現回数である. 表 1 から, $F = 5000$, 学習率 $\eta = 1.7$ のときに評価値 $P_{QE} = 0.814$ と最大になる. したがって, この論文で提案しているシステムはレビューを参考になる順序に並び替える手法として有効であると言える.

しかし, 市川 (2021) では Amazon の商品レビューを分析の対象としていて, 素性には単語の出現頻度を用いているため, 提案システムが成り立つにはしっかり商品をレビューしている文章を学習させる必要がある. そのため, この論文で提案されているシステムでは YouTube で商品を宣伝している動画や, SNS の投稿に対するコメントを学習させた場合に上手く学習できなかったり, 精度が悪くなってしまうことが考えられる. なぜならば, YouTube の動画や SNS の投稿に対するコメントというのは誰でも気軽にでき, 内容も自由であるため, 商品のレビューのようなコメントの数が Amazon の商品レビューに比べると少ないからである. また, 一文の長さも短いことが多く, 素性となり得る単語の抽出も難しいと考えられる. そこで, 本論文ではそのような問題を解決するための手法を第Ⅱ部で提案する.

表 1: 登場回数と学習率の組み合わせごとの評価値 P_{QE} (出典：市川 [1] p.91)

登場回数 F	素性数 N	学習率 η										
		1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
100	2083	0.788	0.783	0.800	0.806	0.796	0.766	0.750	0.762	0.769	0.747	0.768
200	2083	0.788	0.783	0.800	0.806	0.796	0.766	0.750	0.762	0.769	0.747	0.768
500	1472	0.774	0.748	0.781	0.782	0.769	0.769	0.763	0.751	0.787	0.774	0.726
1000	1058	0.728	0.794	0.746	0.781	0.758	0.813	0.792	0.762	0.795	0.776	0.784
2000	701	0.782	0.756	0.781	0.712	0.737	0.734	0.722	0.800	0.795	0.769	0.718
5000	363	0.759	0.773	0.774	0.801	0.764	0.763	0.772	0.814	0.757	0.765	0.755
10000	207	0.795	0.804	0.795	0.809	0.789	0.782	0.794	0.781	0.741	0.787	0.759

2.2 単語の出現頻度と類似性に基づいたトピックモデル洗練化手法 (東 (2019)[2])

東 (2019) では, 第 II 部で後述するトピックモデルの一種の Latent Dirichlet Allocation(以下 LDA) を自然言語文書に適用する際の改善案を提案している. 通常, トピックモデルを自然言語文書に適用する際には, 前処理として分類に不必要なストップワードの除去を行うことが多いが, 一般的にストップワードリストに含まれている単語を除去するだけでは, 特定の文書にのみ頻出する特徴的な単語を除去することが出来ず, トピックモデルの精度に影響を及ぼすという問題が存在する. また, トピックモデルによって分類したトピックには, 類似したトピックが複数出現し, 分類の精度が下がるという問題も存在する.

そこで東 (2019) では, 前処理として分析対象としている文書から適切なストップワードリストを作成する方法を提案している. また, トピックモデルを適用後の後処理として, トピックを構成している単語の類似度からトピック間の距離を算出し, 類似しているトピックを統合することでより正確なトピック分類を可能にする手法を提案している. 図 2 は提案手法の全体像である.

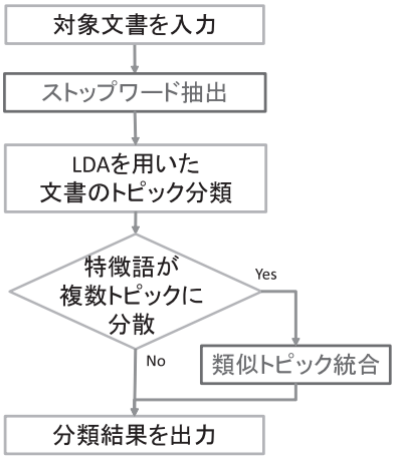

図 2: 提案手法の流れ (出典：東 [2] p.27)

ストップワードリストを作成する手順を図 3 で示している. この手法では, まず対象としている文書全体に対して出現率が高い単語をストップワードとして抽出する. 出現率の算出にはは DF(Document Frequency) を用いている. DF とは, 文書全体に対してある単語 T が含まれる文書数のことであり, 事前に設定した閾値よりも高い DF 値を持つ単語をストップワードリストに加える. 次に, 抽出した単語と意味的に類似している単語をさらにストップワードとしてリストに加え

る．word2vec を用いて文章中の各単語を周辺の単語から学習し, 単語の分散表現を得て単語間の類似度を算出する．それによりある単語 T の類似単語を抽出することができ, ある閾値以上の類似度を示した単語を全てストップワードリストに加える．これにより, DF 値が高くない場合でも文書の特徴を表しにくい単語をストップワードリストに加えることが可能になる．

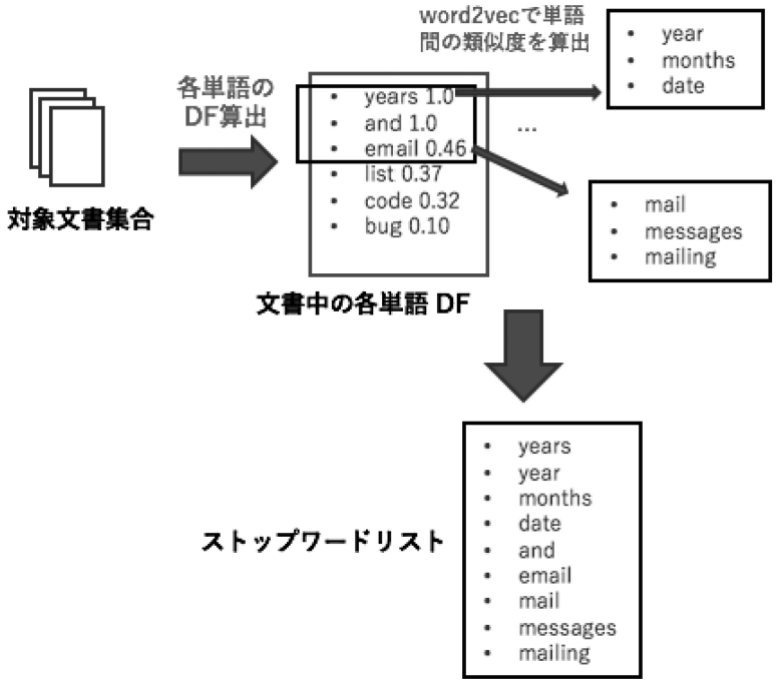

図 3: ストップワード抽出手法の流れ (出典：東 [2] p.27)

トピックモデル適用後の後処理では, 分類結果に似たよったトピックが存在する場合にそれらのトピックを統合する処理を行うことを提案している．類似トピックの判断基準には, TF-IDF cos 類似度推定法が用いられている．これは, cos 類似度の計算に使用するベクトルの成分を TF-IDF で算出したものにした手法である．この研究では, 分類した各トピックの単語集合に対して TF-IDF cos 類似度を利用したクラスタリングを行い, その結果に従ってトピックを統合する手法を提案している．

以上の提案手法を LDA のよるメーリングリストのトピック分類に適用し, 評価項目に基づいて比較することで提案手法の有効性の評価を行っている．評価項目として, 一般的なストップワードリストを用いた手法と提案手法を比較している．また, 後処理として類似トピックの統合を行った場合についても比較を行っている．トピック分類の正確さを評価する指標として, 適合率, 再現率, それらの調和平均の F 値を採用している．実験結果を表 2 に示している．表 2 より, LDA を自然言語に適用したトピック分類では, 前処理としてストップワード除去を行うことで分類精度が大幅に向上していることが分かる．また, 一般的なストップワードリストを用いた場合と提案手法によるストップワードリストを用いた方法を比較すると F 値が向上している．また, 後処理を行った場合の F 値はさらに向上していることから, トピック分類の正確性を向上させる提案手法の有効性を確認できている．

本研究ではトピックモデルの分類精度よりも抽出する単語の質を重視しているため, 東 (2019) で提案されているストップワードリスト作成方法を用いると抽出したい単語をストップワードリストに加えてしまう恐れがある．また, 本研究で分析の対象としている YouTube のコメントは, 書

表 2: ストップワード数（$\#N$）, 適合率, 再現率, F 値の結果 (出典：東 [2] p.30)

	$\#N$	適合率	再現率	F 値
ストップワードなし	なし	0.047	0.497	0.086
Fox ストップワードリスト	425	0.237	0.397	0.297
Poisson ストップワードリスト	1238	0.355	0.422	0.385
RAKE	500	0.091	0.423	0.149
提案手法 (前処理)	761	0.411	0.418	0.414
提案手法 (前処理+後処理)		0.489	0.453	0.470

き言葉よりも話し言葉で書かれていたり, 若者言葉が使われていることが多く, 一般的なストップワードリストだけでは不要な単語を除去することが困難である. そのため, 前処理で形態素解析を行い, 特定の品詞のみ抽出してストップワードとして除去する, または特定の品詞のみでトピック分類を行う手法で実験を行った.

2.3 テキストマイニングを用いた口コミ分析による点数評価の信頼性確認手法 (谷口 (2017)[3])

谷口 (2017) は商品に対して点数評価を加えたレビューを行える口コミサイトに関して, 平均値や点数の分布が示されているなどのメリットがあることに対し, 点数の信頼性に疑問を抱く購入検討者が多い問題に着目している. そこで, カメラを購入した顧客からの点数評価とレビューを用いて, 点数評価の信頼性を確認する手法を提案している.

谷口 (2017) はまず Sony の製品サイト上のカメラに関するレビューから, 5 段階の総合評価の点数を集計している. 次に, カメラの特徴 (画質・機能・デザインなど) に関する点数評価を集計し, その結果を表 3 に, 総合評価で 5 を付けた人の各項目の点数評価の集計結果を表に示している. 谷口 (2017) は, この集計結果より総合評価 5 を付けている人でも全ての項目で満足しているわけで

表 3: 各項目の点数評価 (出典：谷口 [3] p.2)

Score	PICTURE QUALITY	FEATURES	DESIGN	EASE OF USE
5	1,075	937	865	653
4	271	397	458	579
3	28	34	59	126
2	9	14	11	21
1	10	10	7	20
0	13	14	5	7
Total	1,406	1,406	1,405	1,406
5	76.46%	66.64%	61.52%	46.44%
4	19.27%	28.24%	32.57%	41.18%
3	1.99%	2.42%	4.20%	8.96%
2	0.64%	1.00%	0.78%	1.49%
1	0.71%	0.71%	0.50%	1.42%
0	0.92%	1.00%	0.36%	0.50%

表 4: 総合評価 5 の各項目の点数評価 (出典：谷口 [3] p.2)

Score	PICTURE QUALITY	FEATURES	DESIGN	EASE OF USE
5	848	773	742	580
4	78	150	186	317
3	2	3	6	29
2	1	1	1	5
1	0	2	0	3
0	10	10	3	5
Total	939	939	938	939
5	90.31%	82.32%	79.02%	61.77%
4	8.31%	15.97%	19.81%	33.76%
3	0.21%	0.32%	0.64%	3.09%
2	0.11%	0.11%	0.11%	0.53%
1	0.00%	0.21%	0.00%	0.32%
0	1.06%	1.06%	0.32%	0.53%

はないことや, 逆に評価項目にはない部分で製品に対して満足している可能性などに言及している. この点数評価の信頼性を確認するため, 製品に対するレビューをテキストマイニングにて分析し, Positive, Neutral, および Negative の表現を抽出する感性評価を行っている. 総合評価の各点数ごとのレビューから各感性の出現頻度をカウントした結果が表 5 である. 表 5 より, 総合評価 4 では Neutral に分類されている文章が 9 割を超えていることや, 総合評価 5 では半数が Neutral を示し, Negative と判定された口コミが 0 件であることが分かる. この結果より, 谷口 (2017) は総合評価は妥当な点数であるといえることや, 総合評価が満点であっても全ての項目に対して満足しているわけではない場合もあることを明らかにしている.

総合評価 5 のとき Negative が 0 件であることから点数評価の妥当性を判断できるが, Neutral に分類された文書が多い総合評価 4 に妥当性があるかどうかは判断が難しいと感じた. また, 谷口 (2017) の研究は点数評価に対する信頼性を評価しているため, 文書自体の信頼性については確認されていない. 本研究では, Positive, Negative などの評価表現の有無に依存せずに文書の分類 (商品との関連性の有無) を行える手法を提案する.

表 5: 総合評価と口コミ分析の感性評価 (出典：谷口 [3] p.2)

Score	5	4	3	2	1
Positive	497	4	29	9	2
Neutral	442	363	9	3	16
Negative	0	21	13	4	2
Total	939	389	45	13	20
Positive	52.93%	1.03%	64.44%	69.23%	10.00%
Neutral	47.07%	93.32%	20.00%	23.08%	80.00%
Negative	0.00%	5.40%	28.89%	30.77%	10.00%

2.4　グルメサイトにおけるクチコミの信頼性確保に関する一考察 (吉見 (2014)[4])

吉見 (2014) はグルメサイトの「食べログ」で所謂やらせのレビューが行われていた問題によって口コミの信頼性が揺らいでいる問題に着目し, 実名・顕名・匿名といった口コミの差異がその信頼性に与える影響について検討している. 各項目は表 6 の定義に従って分類されている. 吉見 (2014) は以下のリサーチ・クエスチョンについてテキストマイニングの手法を用いた分析を行っている.

> 実名のグルメサイトは「長期的関係による評判」を重視し, 匿名・顕名のグルメサイトは「不完備情報による評判」を重視している. (吉見 2014, p.3)

さらに, 実名の特徴として評判の種類が「長期的関係」であること, 匿名と顕名については「不完備情報」であると仮定している (吉見 2014, 表 3). 匿名・顕名のグルメサイトとして「食べログ」を, 実名のグルメサイトとして「Retty グルメ」を対象に実験を行った結果, 投稿数と 1 投稿あたりの文字数は顕名>匿名>実名の順に小さく傾向があることを明らかにしている. また, 全体の 1 割以上の投稿に現れている単語を対象に共起ネットワーク分析を行った結果, 同様に顕名>匿名>実名の順に密から疎に変化していることが分かり, この結果からも実名の投稿が簡素であるという結果を明らかにしている (図 4, 図 5).

以上の実験結果より, 吉見 (2014) は実名の投稿が「長期的な関係による評判」に依拠していて, レビュアーへの信頼を補完しているため詳細なクチコミを必要としていないと考察している. また匿名・顕名の投稿は「不完備情報による評判」に依拠していて, 他のユーザーからの信頼を獲得するために比較的詳細なクチコミを求められていると考えている. そのため, リサーチクエスチョンは支持されると結論付けている.

本研究で扱う YouTube の動画に対するコメントは匿名・顕名であり, 信頼を獲得するにはある程度詳細な情報が求められるが, YouTube の特性上, 誰でも自由に投稿できるため商品に対する詳細なレビューはかなり少ない. そのため, 本研究では比較的文章が短い文書集合に対してでも適切に分析することが可能であると考えらえれる Biterm Topic Model(Xiaohui Yan 2013) を用いる手法を提案している.

表 6: 匿名・顕名・実名の分類 (出典：吉見 [4] p.2)

匿名	・レビュアー（口コミ主）の同一性が保持されていない状態 ・コミュニティからの離脱・再参入は容易
顕名	・レビュアー（口コミ主）の同一性が何らかの形である程度保持されている状態 （例：「食べログの電話番号認証」） ・コミュニティからの離脱は容易であるが, 再参入はやや困難
実名	・レビュアー（口コミ主）の同一性がかなりの程度で保持されている状態 （例：実名 SNS における投稿） ・コミュニティからの離脱も再参入も容易ではない

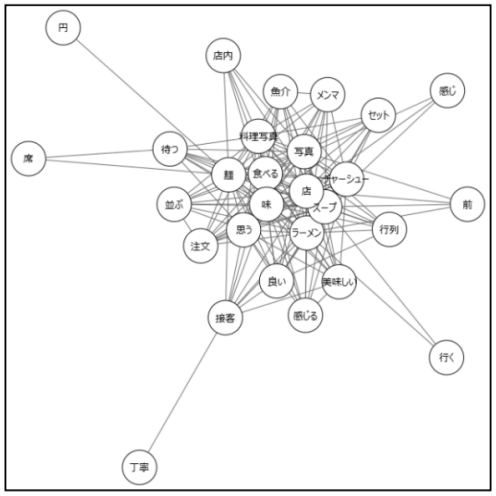

図 4: 顕名のクチコミ (出典：吉見 [4] p.4)

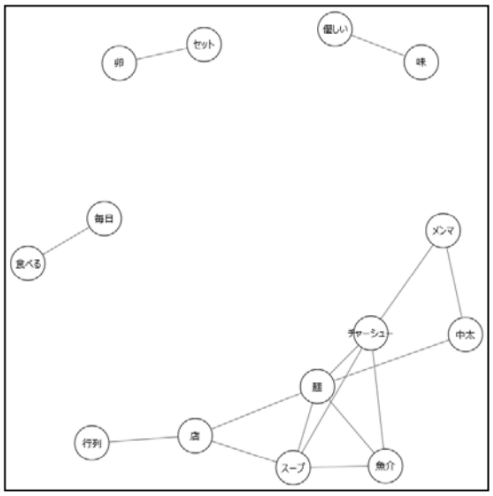

図 5: 実名のクチコミ (出典：吉見 [4] p.4)

3　研究目的

前項までの関連研究を踏まえた上で, 本研究の研究目的を明らかにする. 本研究では YouTube 上で商品やサービスを宣伝している動画に対する視聴者のコメントを分析対象とする. 対象としているコメントのうち, その動画で紹介している商品・サービスと関連性が高いコメントを抽出するシステムの作成を目的としている. [1] でレビューを参考になる順序に並び替えているように, 本研究では後述する提案手法により商品との関連性が高い順にコメントを並び替え, 他のユーザーがその商品の購入を検討する際に参考にするコメントを取得しやすいシステムの作成を目指す. また, 作成したシステムにより抽出した商品との関連性が高いコメントが, 人手の評価に対してどれほどの精度で抽出できているかを評価し, 最終的な提案手法の精度としている.

第 II 部
提案手法

第Ⅱ部では本研究の提案手法, 及び提案手法で用いている主要な技術について説明する.

4　トピックモデル

提案手法の説明に先立ち, 本提案手法において主要な技術であるトピックモデルについて説明する. トピックモデルとは一つの文書が複数のトピック (主題) を持つと仮定する確率生成モデルである. トピックモデルは多くの分野で幅広く活用されており, 例えば文書集合を解析してカテゴリやトピックごとに分類したり, 顧客からのフィードバックやレビューを解析して商品やサービスに関する主要な問題点や改善点を特定したり, 生物医学研究では疾患や生物学的プロセスに関連するパターンを特定することに用いられている. 以下にトピックモデルの概要を示す. 一つの文書が一つのトピックを持つと仮定している「混合ユニグラムモデル」では文書集合全体で一つのトピック分布があるのに対して, トピックモデルでは文書ごとにトピック分布 $\boldsymbol{\theta}_d = (\theta_{d1}, \cdots, \theta_{dK})$ が存在する. ここで, $\theta_{dk} = p(k|\boldsymbol{\theta}_d)$ は文書 d の単語にトピック k が割り当てられる確率であり, $\theta_{dk} \geq 0, \sum_{k=1}^{K} \theta_{dk} = 1$ を満たす. このトピック分布 $\boldsymbol{\theta}_d$ に従って文書 d のそれぞれの単語にトピック z_{dn} が割り当てられる. そして割り当てられた各トピックの単語分布 $\boldsymbol{\phi}_{z_{dn}}$ に従って単語が生成される. ここで, トピックごとの単語分布は $\boldsymbol{\Phi} = (\boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_K)$ と表せ, $\boldsymbol{\phi}_k = (\phi_{k1}, \cdots, \phi_{kV})$ はトピック k の単語分布を表す. $\phi_{kV} = p(v|\boldsymbol{\phi}_k)$ はトピック k で単語 v が生成される確率 ($\phi_{kv} \geq 0, \sum_{v=1}^{V} \phi_{kv} = 1$) を表している. 単語の生成過程を図 6 に, 表 7 に 4 節で用いている記号を示す. 文書ごとのトピック分布 $\boldsymbol{\theta_d}$ 及びトピックごとの単語分布 $\boldsymbol{\phi_k}$ はカテゴリ分布のパラメータであるため, その共役事前分布であるディリクレ分布から生成されると仮定している.

表 7: 4 節で用いる記号

記号	説明
D	文書数
N_d	文書 d に含まれる単語数
V	文書集合に現れる単語の種類数
$\boldsymbol{W}$	文書集合
$\boldsymbol{w}_d$	文書 d
w_{dn}	文書 d の n 番目の単語
K	トピック数
N_k	文書集合全体でトピック k が割り当てられた単語数
N_{dk}	文書 d でトピック k が割り当てられた単語数
N_{kv}	文書集合全体で語彙 v にトピック k が割り当てられた単語数
θ_{dk}	文書 d でトピック k が割り当てられる確率
ϕ_{kv}	トピック k のとき語彙 v が生成される確率

1. for トピック $k = 1, \cdots, K$
   (a) 単語分布を作成 $\boldsymbol{\phi}_k \sim \mathrm{Dirichlet}(\beta)$
2. for 文書 $d = 1, \cdots, D$
   (a) トピック分布を生成 $\boldsymbol{\theta}_d \sim \mathrm{Dirichlet}(\alpha)$
   (b) for 単語 $n = 1, \cdots, N_d$
      i. トピックを生成 $z_{dn} \sim \mathrm{Categorical}(\boldsymbol{\theta}_d)$
      ii. 単語を生成 $w_{dn} \sim \mathrm{Categorical}(\boldsymbol{\phi}_{z_{dn}})$

図 6: トピックモデルの生成過程

同一の文書に含まれる単語でも, 異なるトピックが割り当てられることがあるため, 一つの文書が複数のトピックを持つことが可能である. また, 語彙ごとにトピックが割り当てられるわけではなく, 単語ごとにトピックが割り当てられるため, 同じ語彙でも異なるトピックが割り当てられる可能性も存在する. また, トピックモデルは単語にトピックを割り当てる, または単語をクラスタリングするモデルと考えることもできる. トピック分布 $\boldsymbol{\theta_d}$ と単語分布集合 $\boldsymbol{\Phi}$ が与えられたときの文書 $\boldsymbol{w}_d$ の確率は式 (3) で表せられる.

$$p(\boldsymbol{w}_d|\boldsymbol{\theta}_d, \boldsymbol{\Phi}) = \prod_{n=1}^{N_d} \sum_{k=1}^{K} p(z_{dn} = k|\boldsymbol{\theta}_d) p(w_{dn}|\boldsymbol{\phi}_k) = \prod_{n=1}^{N_d} \sum_{k=1}^{K} \theta_{dk} \phi_{k w_{dn}} \tag{3}$$

図 7 にトピックモデルのグラフィカルモデルを示す. グラフィカルモデルとは, 生成モデル内の変数の依存関係を直感的に理解できるように描いた表現方法である. ここで, 色付きの円は観測変数, 白の円は未知変数を表している. 四角は繰り返しを表し, 右下の数字は繰り返し回数を表している. また, 右側の四角は単語分布 $\boldsymbol{\phi}$ がトピック数 K あることを表している. 左外側の四角は文書数 D を, 左内側の四角は各文書に N 単語含まれることを表している. α はトピック分布のハイパーパラメータ, β は単語分布のハイパーパラメータを表している. このグラフィカルモデルより, トピック分布 $\boldsymbol{\theta}$ が文書ごとに存在し, トピック z が単語ごとに存在することが直感的に理解することができる.

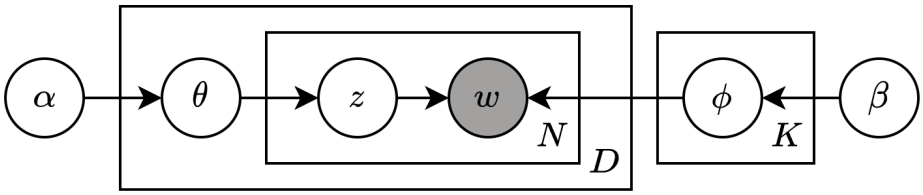

図 7: トピックモデルのグラフィカルモデル表現

4.1 Latent Dirichlet Allocation

トピックモデルの代表的な手法に潜在ディリクレ配分法 (Latent Dirichlet Allocation: 以下 LDA) がある．LDA では各文書のトピック分布 $\boldsymbol{\theta}$ がディリクレ分布に従うと仮定している．この分布はハイパーパラメータ α によって定義され, 式 (4) で表される．ここで, $Dir(\alpha, \cdots, \alpha) = Dir(\boldsymbol{\alpha})$, $B(\boldsymbol{\alpha})$ はディリクレ分布の正規化定数であり, 式 (5) で表される．α は文章中のトピック分布を平滑化する役割があり, 値が小さいほど各文書は少数のトピックに集中する傾向がある．逆に α の値が大きい場合は各文書がより多くのトピックを含む傾向にあり, 一つの文書内で多様なトピックが出現する確率が高くなる．

$$Dir(\boldsymbol{\theta}_d|\boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{i=1}^{K} \theta_{di}^{\alpha_i - 1} \tag{4}$$

$$B(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{K} \alpha_i)} \tag{5}$$

同様に各トピックの単語分布 $\boldsymbol{\phi}$ もディリクレ分布に従うと仮定している．単語分布はハイパーパラメータ β によって定義され, 式 (6) で表される．ここで, $Dir(\beta, \cdots, \beta) = Dir(\boldsymbol{\beta})$, $B(\boldsymbol{\beta})$ はディリクレ分布の正規化定数であり, 式 (7) で表される．β が小さいとき, 各トピックは比較的少数の単語に集中する傾向があり, トピック間の区別が明確になりやすい．$beta$ が大きいとき, 各トピックは多様な単語を含むようになる傾向があり, トピック間の明確な区別が困難になる場合がある．

$$Dir(\boldsymbol{\phi}_k|\boldsymbol{\beta}) = \frac{1}{B(\boldsymbol{\beta})} \prod_{j=1}^{V} \phi_{kj}^{\beta_j - 1} \tag{6}$$

$$B(\boldsymbol{\beta}) = \frac{\prod_{j=1}^{V} \Gamma(\beta_j)}{\Gamma(\sum_{j=1}^{V} \beta_j)} \tag{7}$$

LDA の生成過程を図 7 のグラフィカルモデルを用いて説明する．グラフィカルモデルの左側の α がトピック分布を生成するディリクレ分布のハイパーパラメータである．ディリクレ分布に従い, 各文書のトピック分布 $\boldsymbol{\theta}_d$ が決定する．例えば, オリンピックに関連するニュースの文章に対しては「スポーツ」や「経済」といったトピックの確率が高い分布が生成される．このとき,「スポーツ」などのトピックのラベルはトピックモデルが推定するわけではなく, 人手で判断して付与するものである．次に, 先ほど求めたトピック分布 $\boldsymbol{\theta}_d$ から文書中の各単語の潜在トピックを多項分布により求める．例えば文書中の単語数 $N = 10$ の場合,「スポーツ」のトピックの確率が高い文書では 6 つの単語に「スポーツ」トピックが割り当てられ, 残りの単語に次に確率が高いトピックを順に割り当てていくことで文書中の単語全てにトピックが割り当てられる．グラフィカルモデルの右側の β が各トピックの単語分布を生成するディリクレ分布のハイパーパラメータである．ディリクレ分布に従い, 各トピックの単語分布 $\boldsymbol{\phi}$ が決定される．例えば,「スポーツ」トピックであれば「野球」「ボール」などが,「経済」トピックであれば「GDP」や「円高」などの単語の確率が高い分布が生成される．

このように, LDA は文書集合の学習データからトピック分布 $\boldsymbol{\theta}$ と単語分布 $\boldsymbol{\phi}$ を推定し, 与えられた文書が推定した分布によって生成されると仮定した確率生成モデルである. LDA の出力であるトピック分布 $\boldsymbol{\theta}$ と単語分布 $\boldsymbol{\phi}$ の同時分布は式 (8) で表せられる.

$$P(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{\phi} : \boldsymbol{\alpha}, \boldsymbol{\beta}) = \prod_{d=1} P(\boldsymbol{\theta}_d|\boldsymbol{\alpha}) \prod_{k=1} P(\boldsymbol{\phi}_k|\boldsymbol{\beta}) \prod_{n=1} P(z_{d,n}|\boldsymbol{\theta}_d) P(w_{d,n}|\boldsymbol{\phi}_k z_{d,n}) \tag{8}$$

ここで説明した LDA は代表的なトピックモデルであり様々なタスクで用いられているが, 本研究の提案手法に用いるにはあまり適していないと考えている. 本研究は SNS の投稿に対するコメント, その中でも YouTube の動画に対するコメントを研究の対象としている. SNS や YouTube の投稿に対するコメントは, EC サイトや口コミサイトの商品レビューに比べて一文の長さが比較的短いという特徴がある. 一文が短い文章の場合, LDA では単語のスパース性が問題となりトピック分布などの推定が適切に行えない可能性がある. LDA で推定した単語分布は文書集合に出現する全ての単語を候補としているが, 例えば 1000 件の文章を対象としている場合, 一つの文章に一種類ユニークな単語が出現するだけで単語分布の候補数は 1000 個を超えることとなる. そして一文の長さが短い場合, 当たりの単語 (その文章中に含まれている単語) の数が少ないというスパース性が問題となり, 多くの文章に対して共通するトピック分布, 及び単語分布の精度が悪くなってしまう可能性が高い. そのため, SNS や YouTube の投稿に対するコメントの分析には LDA は適していないと考えている. そこで, 一文の長さが比較的短い場合でも適切にトピックの推定が行えると予想される Biterm Topic Model(BTM) を本研究の提案手法に用いることを考えた.

4.2 Biterm Topic Model

Biterm Topic Model(BTM) はトピックモデルの一種であり, 特に短い文書に適したモデルである. LDA とは異なり, 文書レベルではなく文書集合全体の単語の対 (Biterm) を直接モデリングすることにより短い文書における単語のスパース性の問題を緩和している. 図 8 に BTM の生成過程を, BTM のグラフィカルモデルを図 9 に示す.

1. for トピック $z = 1, \cdots, Z$
   (a) 単語分布を作成 $\boldsymbol{\phi}_z \sim \mathrm{Dirichlet}(\beta)$
2. 文書全体に対してトピック分布 $\boldsymbol{\theta} \sim \mathrm{Dirichlet}(\alpha)$
3. biterm 集合 $\boldsymbol{B}$ における各 biterm b に対して
   (a) トピックを生成 $z \sim \mathrm{Multi}(\boldsymbol{\theta})$
   (b) 2 つの単語を生成 $w_i, w_j \sim \mathrm{Multi}(\boldsymbol{\phi}_z)$

図 8: BTM の生成過程

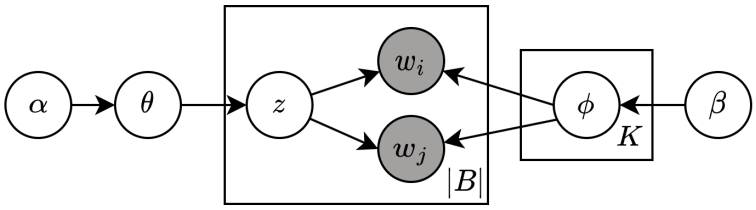

図 9: BTM のグラフィカルモデル

まず, 各トピック z に対して単語分布 $\boldsymbol{\phi}_z$ がディリクレ分布に従って生成される. 次に, 文書全体に対してトピック分布 $\boldsymbol{\theta}$ がディリクレ分布に従って生成される. LDA では各文書に対してトピック分布が存在していたため, 短い文書や単語が少ない文書ではトピックを適切に推定することが困難であるが, BTM では文書全体で集約された単語の共起性を利用するため LDA に比べて優れたトピックの推定が行える. そして, 各 biterm b に対してトピック分布 $\boldsymbol{\theta}$ に従いトピック z が選択される. 選択されたトピック z の単語分布 $\boldsymbol{\phi}_z$ から二つの単語を生成する. 図 8 の手順に従って biterm $b = (w_i, w_j)$ の同時確率は式 (9) のように, 文書全体の尤度は式 (10) のように表せられる.

$$\begin{aligned} P(b) &= \sum_z P(z)P(w_i|z)P(w_j|z) \\ &= \sum_z \theta_z \phi_{i|z} \phi_{j|z} \end{aligned} \tag{9}$$

$$P(B) = \prod_{(i,j)} \sum_z \theta_z \phi_{i|z} \phi_{j|z} \tag{10}$$

このように BTM では biterm という二単語の対をモデル化の基本単位としているため, 単語の数が少ないような短い文書でも文書集合全体の単語の共起性をとらえることでトピック推定の精度を向上させている. また, LDA とは異なり単一の文章を生成するという仮定ではなく, 文書集合から抽出した biterm を対象にそれらを生成するという仮定を持つ. つまり, 短い文書の場合に LDA では少ない単語の情報から文書ごとのトピック分布を推定するのに対し, BTM では全文書に出現する単語の共起から文書集合全体のトピック分布を推定するため精度が高くなる. 本研究で実験対象とする YouTube の動画に対するコメントは比較的一文が短いものが多く, BTM の優位性が予測される. BTM を用いた具体的な提案手法は 5.3 節で述べる.

5　提案手法

ここでは, 前項で説明した Bitem Topic Model を用いて, YouTube 上で自社製品やサービスを宣伝している動画に対するユーザーのコメントから, 宣伝している商品やサービスに対して関連性が高いコメントを抽出するシステムを提案する. また, 提案したシステムの精度を検証する方法についても説明する.

5.1　データ収集

実験に用いる YouTube のコメントは, YouTube Data API v3 を用いて取得した. YouTube Data API v3 は Google Cloud Console で API キーを作成し, API を有効化することで様々な YouTube データにアクセスすることが可能になる. そのうち, YouTube のコメントに関連するものとして表 8 のようなデータが挙げられる.

表 8: YouTube Data API v3 で取得できるコメント情報

項目	内容
videoID	コメントした動画の ID
textDisplay	現在表示されているコメント
textOriginal	最初に投稿されているコメント
authorDisplayName	コメント投稿者の名前
authorProfileImageUrl	コメント投稿者のアイコン
authorChannelUrl	コメント投稿者のチャンネル
authorChannelId	コメント投稿者のチャンネル ID
likeCount	コメントに付いたいいねの数
publishedAt	コメントの投稿日
updatedAt	コメントの最終更新日

本研究では, 表 8 のうち textDisplay(現在表示されているコメント) のみを抽出し, 実験を行なった. また, YouTube のコメントには元のコメントの他に別のユーザーが返信しているケースも多いが, 本研究では返信しているコメントは扱わず, 元のコメントのみを抽出し実験の対象としている. 抽出したコメントの一例を表 9 に示す.

表 9: 抽出したコメントの一例

絶対食べたい😋お腹すいた(›´ω`‹)
本当に…幸せそうな人を見るのって、こっちまで幸せな気持ちになるからいいよなぁ……!!!!!!!!☺
好きなことをして生きていくのがYouTuber
本当に努力の塊でしかない💁❤HIKAKINさんが頂点で本当によかったぁぁぁぁぁぁ😖😳💁
新幹線代往復４万チョイ払ってでも『みそる』🍷食べに行きたい

5.2 前処理手法

トピックモデルを含む自然言語処理の様々な手法において, テキストデータに対する前処理は非常に重要である. Web テキストを扱う場合には HTML タグや JavaScript のコードが含まれることもあり, 前処理としてそのようなノイズを除去する必要がある. また, 本研究では YouTube の動画に対するコメントをテキストデータとして扱うが, YouTube のコメントや SNS の投稿には絵文字や顔文字, URL, 話し言葉などを含んでいることが多い. そのため, 本研究でもトピックモデルによる分類を行う前の前処理は非常に重要である. 本研究で行った主な前処理とその簡単な説明を以下に示す.

5.2.1 クリーニング処理

空白、改行文字を削除

半角空白や全角空白, 及び “\n” などの改行文字を空文字に変換する.
例:「おはよう　今日はいい天気ですね」→「おはよう今日はいい天気ですね」

記号除去

“!” や “#”, 及び全角記号を除去する. また, YouTube のコメントの特性上顔文字が使われることも多いため, それらを除去する目的でもある.
例:「今晩、友達と映画を見に行く予定です！ 楽しみですね (ˇ ω ˇ)」→「今晩友達と映画を見に行く予定です楽しみですね」

絵文字除去

顔文字と同様に YouTube のコメントで使われることが多い. 感情分析などでは絵文字から情報を取得することもあるが, 本研究では感情に関する情報の抽出, 分析は行わないため絵文字は除去する.

数字を 0 に置換

自然言語処理の様々な手法において, 数字は意味を為さないことが多いため, 1 つ以上連続している数字を全て 0 に置換することが多い. 本研究でも数字に関する情報を必要としないため, 数字は全て 0 に置換する.
例:「目標は年間で 10 回のイベントを開催することです」→「目標は年間で 0 回のイベントを開催することです」

単語の正規化

単語の正規化とは, 単語の文字種の統一, つづりや表記ゆれなどを無くすことである. この処理を行うことで同じ意味で異なる表記や形態の単語が同じ形になり, テキストの処理や解析が容易になる. 単語の正規化にはいくつか種類があり, 例えば

- テキスト内のアルファベットを全て小文字に変換する
- 半角カナを全角に統一する
- 辞書を用いた単語の統一

などがある.
例:「Google でﾈｺの写真を検索してください」→「google でネコの写真を検索してください」

連続長音記号除去, 繰り返し文字のまとめ

話し言葉や若者言葉でよくある「きたーーーーーー」や「うおおおおおお」など, 連続して長音記号が含まれている場合や同じ単語が繰り返されているものを削除, または一つにまとめる処理を行った.

例：「食べたーーーーい！！」→「食べたーい」

その他の前処理

スパムの可能性があるため, URL を含むコメントを削除した. また, YouTube の特性上外国人のコメントも多く存在したため, 日本語と英語以外の言語を含んでいるコメントを削除した.

以上の前処理を表 9 のコメントに適用した結果を表 10 に示す.

表 10: 前処理後のコメント

絶対食べたいお腹すいた
本当に幸せそうな人を見るのってこっちまで幸せな気持ちになるからいよなぁ
好きなことをして生きていくのがyoutuber
本当に努力の塊でしかないhikakinさんが頂点で本当によかったぁ
新幹線代往復0万チョイ払ってでもみそる食べに行きたい

5.2.2 MeCab による形態素解析及び分かち書き

次に, トピックモデルで学習する際に必要な文章の分かち書きを行う. 分かち書きとは, 自然言語処理の様々な手法において文章を単語や形態素などの最小単位に分割する処理のことである. この処理を行うことで, 言語解析や機械学習の際にテキストをより扱いやすい形態で実験を行うことができる. また, 英文の場合は単語間にスペースが明示的に存在するため分かち書きは必要ない場合が多いが, 日本語に関しては単語間のスペースがなく, 文章を単語単位に分割する処理を行わないと機械が単語を認識し解析することが難しくなるため, 分かち書きが必要である.

本研究では MeCab を利用して形態素解析, 及び分かち書きを行った. MeCab は京都大学情報学研究科と日本電信電話株式会社コミュニケーション科学基礎研究所の共同研究で開発されたオープンソースの日本語形態素解析エンジンであり, 日本語の文法や単語の品詞情報をもとに文章を形態素に分解したり, 品詞の付与などが可能である. 本研究では MeCab を利用して対象のコメントに対して形態素解析を行い, 形態素に分割後スペース区切りで繋ぐことで分かち書きを行う. MeCab には最初から分かち書きを行う機能も含まれているが, 形態素解析によって抽出した品詞をストップワード除去に用いるためこの手法で分かち書きを行う.

また, 形態素解析の精度はエンジンのアルゴリズムの精度に加え, 形態素解析辞書の精度にも左右される. そのため, 形態素解析の目的にあった辞書を指定し, 解析することが重要となる. 表 11 は MeCab で形態素解析を行うときに主に用いられている辞書を比較したものである. 通常, MeCab での形態素解析には標準搭載されている mecab-ipadic を用いるが, mecab-ipadic は基本的な文法や専門用語に強い反面, 辞書の更新がないため新しい単語や固有表現に弱いという特徴がある. 本研究の分析対象である YouTube の動画に対するコメントは比較的新しい言葉や固有名詞などを含むことが多いため, mecab-ipadic に多数の web 上の言語資源から得た新語を追加し, カスタマイズした mecab-ipadic-NEologd を本研究では形態素解析の辞書に用いている.

表 11: 形態素解析辞書の比較

形態素解析辞書	特徴
mecab-ipadic	IPA コーパスをもとにした MeCab に標準搭載されている IPA 辞書. 基本的な日本語の文法や専門用語などの固有名詞に強いが, 辞書の更新がないため新しい言葉や固有名詞に弱い.
UniDic-mecab	言語学・国語学や音声情報処理など, より多様な目的に適した辞書. 「短単位」という揺れが少ない斉一な単位を見出し語に採用している.
mecab-ipadic-NEologd	多数の web 上の言語資源から得た新語を追加しカスタマイズした MeCab 用のシステム辞書. 辞書の更新が行われるので, 新しい固有表現に強い.

MeCab による形態素解析, 及び分かち書きの処理を行った後, 2.2 節で述べた通り品詞によるストップワード除去を行う. さらに一般的なストップワードリストを用いたストップワード除去を組み合わせることで, 最終的に実験に用いるテキストの形式として整形する. 品詞によるストップワード除去では, 助詞・助動詞などのトピック分類に必要ない品詞を除く手法や, 名詞・形容詞などのトピックにかかわる品詞を抽出する手法で行う. 図 10 では mecab-ipadic と mecab-ipadic-NEologd で形態素解析した結果を比較している. 例えば, 近年登場した少年漫画である「鬼滅の刃」という単語を含む文章を形態素解析した場合, mecab-ipadic では “鬼” “滅” “の” “刃” と分割されてしまっているが, 辞書の更新が行われる mecab-ipadic-NEologd では “鬼滅の刃” と一単語で認識されていて, 適切な形態素解析が行われていると言える.

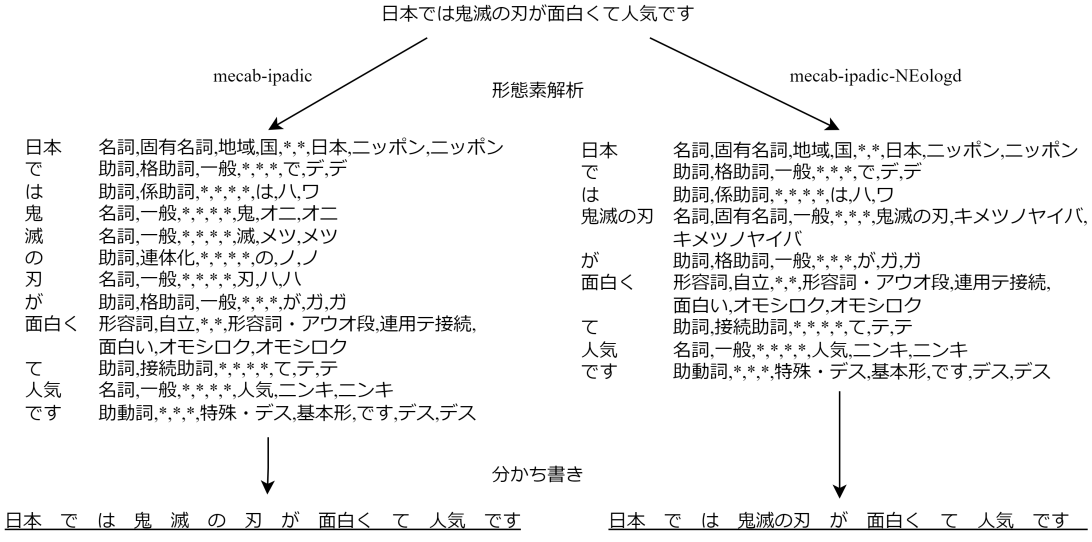

図 10: 辞書による形態素解析結果の比較

5.3 BTM によるトピック抽出

前処理及び分かち書きを行ったテキストデータに対して, 4.2 節で説明した Biterm Topic Model(BTM) を用いてトピックの推定, 及び単語分布の推定を行う. BTM の実装には Python のライブラリである bitermplus を用いた. bitermplus の get_words_freqs() メソッドを用いて, 整形したテキストデータから全ての単語リスト, 及び各単語の頻度などを抽出する. 抽出した単語リストを基に, get_vectorized_docs() メソッドを用いて与えられたテキストデータ内の各文書をベクトル形式に変換する. ベクトル化した各文書から隣接する単語の対 (biterm) を get_biterms() メソッドを用いて生成する. ここで生成した biterm により文書内の単語の共起性の情報を捉え, トピックをより正確に推定することができる. 生成した biterm を用いて BTM を訓練し, トピック分布, 及び単語分布を生成する. モデルを訓練する際にはいくつかのパラメータが存在し, 生成するトピック数：T, 各文書で考慮する単語数：M, ディリクレ分布のハイパーパラメータ α, β などが重要なパラメータである. 本研究では一つの動画に対するコメント集合に対して BTM を用いるため, トピック数 T はあまり大きくなくかつ最低限複数のトピックを抽出できるように $T = 5$ とした. 各文書で考慮する単語数とは, モデルの学習に使用する各文書の単語数を示しており, 各文書中の頻度上位 M 個の単語をトピックモデルの学習に用いる. 本研究の実験対象は一文が比較的短い文章であり含まれる単語数も少ないため $M = 10$ とした. トピック分布, 及び単語分布を生成するディリクレ分布のハイパーパラメータ α, β は本研究ではスカラー値を採用しており, その値は対象とする動画によって最適な値を求めて実験を行う. 以上の手順通り BTM を学習し, 推定した各トピックごとの単語分布より, 出現確率上位 n 単語を抽出する. n は後述する文章生成の際に冗長な文とならないような値を選択する必要があり, 本研究では $n = 10$ を基本的な値とした. 図 11 に BTM によって抽出したトピック ($T = 5$) とその出現確率上位 10 単語の例を示す. これは新作のカップラーメンを紹介している動画に対するコメントであり, 図 11 の結果にあるように「ラーメン」や「味噌ラーメン」, 商品開発者である「ヒカキン」などの単語が抽出できていることが分かる. また, 図下部にある perplexity はトピックモデルの性能評価によく使われている指標である. perplexity はモデルに従って単語を選ぶ困難さを表していて, 各単語の出現確率の逆数の幾何平均で定義されている (式 (11)). 直感的には perplexity はモデルによって生成する単語の候補数に対応していて, 候補数が少ない方が正解となる単語を当てやすいため, perplexity は低い方がモデルとしての性能が高いことを示している. 本研究ではこの perplexity をモデルの性能評価に用いてハイパーパラメータ α, β を調整する.

$$\text{perplexity}(\boldsymbol{W}|\text{M}) = exp\left(-\frac{\sum_{d=1}^{D} \log p(\boldsymbol{w}_d|M)}{\sum_{d=1}^{D} N_d}\right) \tag{11}$$

```
Topic 0: hikakin ヒカキン ラーメン 好き すごく 嬉しい 元気 絶対 努力 味噌ラーメン
Topic 1: ヒカキン ラーメン 美味し 楽しみ 応援 報告 みそ hikakin お願い ラーメン屋
Topic 2: ラーメン ヒカキン hikakin 尊敬 商品 youtuber 好き 絶対 報告 すごい
Topic 3: ラーメン ヒカキン 絶対 楽しみ hikakin 大好き 発売 好き すごい 味噌ラーメン
Topic 4: まずい 康平 奥村 味噌 外食 スガキヤ ちゃんねる 休日 土日 お昼
perplexity: 188.97331680352286
```

図 11: BTM の出力例

5.4 文章生成

各トピックごとの出現確率上位の単語を抽出した結果を用いて, 各トピックごとに文章を自動で生成する. 抽出した単語は各トピックの出現確率上位の単語であるため, ここで生成された文章は各トピックごとに代表的であるという仮説を立て, 以降の手法に適用する. 文章を自動で生成する方法として, 大規模言語モデルである GPT-4 を搭載した ChatGPT を用いる方法で行った. プロンプトには以下のルールを入力して各トピックごとに文章を生成した.

抽出した単語を空白区切りで入力する

BTM によって抽出した n 単語 (n は抽出する単語数) を空白区切りで入力する. ChatGPT に正しく単語を認識させるため, 空白区切りにする必要がある.

全ての単語を使用する

抽出した単語を全て使用することで, 後述する類似度計算の精度を向上させる.

人名がある場合は指定する

人名を ChatGPT が認識できない場合, 文章が支離滅裂になる可能性があるため個別に指定する.

一つの文章に多くの単語を用いて, 文章数は少なくする

生成される文章はトピックごとに代表的であるという仮説の元, YouTube のコメントの特徴に近い文章を生成したいため, 長文を避けるようなルールを追加している.

YouTube のコメントのように生成する

人が投稿するコメントに近しい文章を生成する.

図 12 に ChatGPT に入力する単語, プロンプト, 出力結果の例を示す. 提案手法では各トピックごとに文章を生成するため, 図 12 のような結果がトピックの数だけ生成される.

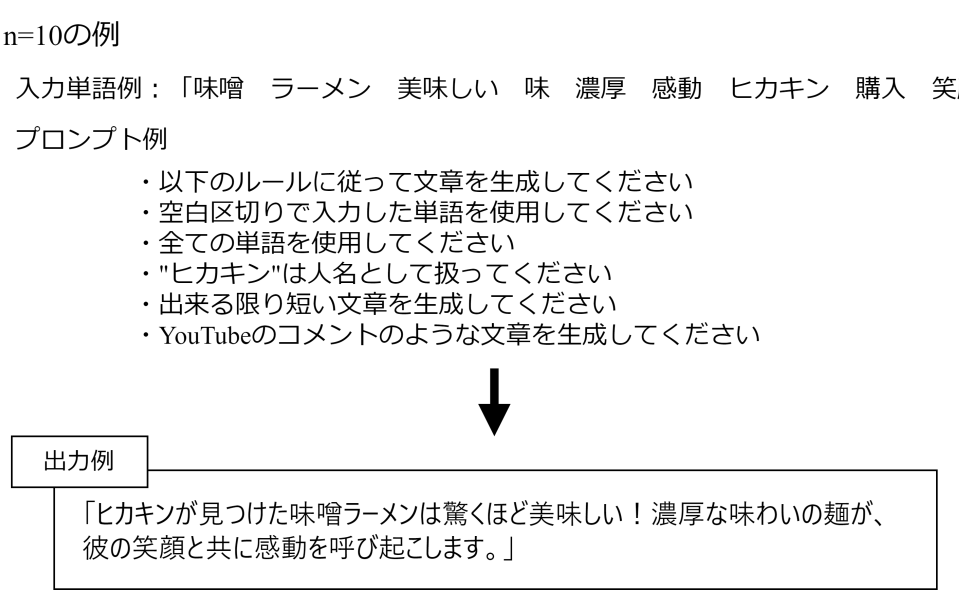

図 12: ChatGPT を用いた文章生成例

5.5 文章間の類似度計算

前節で ChatGPT を用いて自動で生成した文章と, 5.2.1 節で述べたクリーニング処理を施した YouTube の元コメントとの文章間の類似度を計算し, 数値が高い順に並び替えることで, 本研究の目的である, YouTube 上で自社製品やサービスを宣伝している動画に対するコメントのうち他のユーザーがその商品の購入を検討する際に参考にできるようなコメントの抽出を実現する. 文章間の類似度計算手法には Cos 類似度を用いる. Cos 類似度とは, 二つのベクトルが「どれくらい似ているか」を表す尺度であり, 二つのベクトルがなす角の Cos 値のことである. Cos 類似度は, 式 (12) のように二つのベクトルの内積を二つのベクトルのノルムで割ることで求められる.

$$\text{Cosine Similarity} = \frac{\boldsymbol{A} \cdot \boldsymbol{B}}{\|\boldsymbol{A}\| \, \|\boldsymbol{B}\|} \tag{12}$$

前節で生成した文章は, 元のコメント集合から抽出した潜在的なトピックごとの代表的な文章であるという仮説を立てていることから, 生成した文章との文章間の Cos 類似度が高い YouTube の元コメントはそのトピックとの関連性が高い文章といえる. また, 本研究では抽出したトピックにラベル付けを行わないため, どのようなトピックとの関連性が高いかは判断することができない. しかし本節で行う類似度計算では生成したトピック数の文章全て (k=5 であれば 5 文全て) と元のコメント一文との Cos 類似度をそれぞれ計算し, 一番高い数値をその文章の類似度として扱うため, 具体的に何に関するトピックかは判断できないが動画内容との関連性を数値で示すことができると考えた.

Cos 類似度は二つのベクトルの類似度を計算するため, 本手法に用いる上で文章をベクトル化する必要がある. 文章をベクトル化する手法は数多く存在するため, 目的によって適切な手法を選択することが重要である. 本研究では, 単語の重要度を考慮できる TF-IDF による文章のベクトル化と, 様々なタスクに適応できる汎用性を持つ BERT の事前学習済みモデルを用いた文章の埋め込み (ベクトル化) の二つの手法で実験を行う.

5.5.1 TF-IDF

TF-IDF とは, Term Frequency(単語頻度) と Inverse Document Frequency(逆文書頻度) の積で定義される, 文書中のある単語の重要度を表す指標である. TF は「一つの文書 d に単語 t がどれだけの割合で出現したか」を定量的に表した指標であり, 式 (13) で表せられる. ここで, $n_{d,t}$ は文書 d 中に単語 t が出現する回数, T は一つの文書における単語数の合計である.

$$TF_{d,t} = \frac{n_{d,t}}{\sum_{t=1}^{T} n_{d,t}} \tag{13}$$

IDF は「ある単語を含む文書が文書集合の中でどれくらいの割合を占めているか」を定量的に表した指標であり, 実際の式 (14) ではその割合の逆数の対数を取っている. つまり,「ある単語が一部の文書にしか現れない度合い」を計算していることとなる. ここで, N は全体の文書数, df_t は単語 t が出現する文書数を表す.

$$IDF_t = \log \frac{N}{df_t} \tag{14}$$

TF-IDF は式 (15) のように TF(式 (13)) と IDF(式 (14)) の積で表せられる.

$$TF\text{-}IDF_{d,t} = TF_{d,t} \times IDF_t \tag{15}$$

したがって, TF-IDF は以下の条件のときに高い数値を示す.

- その単語の単語頻度が高い
- 文書集合全体に対して, その単語の文書頻度が低い

この計算を全ての文書, 単語に対して行うことで文書に含まれる単語の重要度から文書の特徴を定量的に求めることができる. そして, 各単語に対する TF-IDF 値を要素としたベクトルを生成し, Cos 類似度の計算に用いる.

5.5.2 BERT

BERT(Bidirectional Encoder Representations from Transformers) は Google によって 2018 年に開発された Transformer をベースとした自然言語処理モデルである. 従来の自然言語処理モデルでは文章を前から読み文脈を理解していくのに対して, BERT では Masked Language Model(MLM) というモデルを使用することで文章を文頭と文末の双方向から学習している. MLM ではテキストの一部を [MASK] という別の単語で置き換えたテキストを入力し, その前後の文脈に基づいて [MASK] の単語を予測するようにモデルを訓練する. 文章を双方向から学習することにより, ある単語の前後の文脈を捉えることができ自然言語処理モデルとしての性能を大幅に向上させた. 加えて, 文単位での学習を行う Next Sentence Prediction(NSP) という手法を組み合わせることでさらにモデルの性能を向上させている. NSP は二つの文章の関係性について予測するタスクであり, 二つの文章を入力した後に「二つの文章は連続しているかどうか」を判定するタスクを繰り返すことで学習を行う. 図に示すように, 二つの文章が連続している場合は IsNext , そうでない場合は NotNext の判定を行う. 各入力の最初にある [CLS] トークンは主に分類タスクにおいて使用され, [CLS] トークンに関連付けられた隠れ層の状態 (ベクトル) が入力文の全体的な意味を捉えるようになる. また, [SEP] は文の区切りを示している. この NSP という手法により, 単語だけではなく文章のつながりに関して学習することができる.

Input = [CLS] 私は [MASK] を読んでいます。[SEP]
それはとても [MASK] です。[SEP]
Label = IsNext

Input = [CLS] 昨日、私は新しい [MASK] を買った。[SEP]
木星は [MASK] の中で最大の惑星です。[SEP]
Label = NotNext

図 13: NSP による文章のつながり判定例

本研究では, この BERT モデルを用いた文章の埋め込み (ベクトル化) を行い, Cos 類似度の計算に用いている. BERT モデルでは, テキストを前処理した後トークン化する. BERT におけるトークン化は通常の単語分割よりも複雑な場合があり, 例えば単語本体と接頭辞, 接尾辞に分割するサブワード分割を行うことがある (例：“playing” → [“play”, “##ing”]). また, BERT の出力は入力された各トークンに対する文脈依存の埋め込みである. これらの埋め込みは単語の意味がその文脈によってどのように変化するかを捉えることができる. 例えば,「この部屋は明るい」と「彼は明るい性格だ」では「明るい」の意味が異なり, 文章埋め込みの出力結果も異なる. このように BERT モデルを使用した文章の埋め込みでは, 文章を双方向から学習することで文脈の理解において高い精度を示したり, 単語の意味の差異を理解し適切な結果を出力したりできる. この BERT モデルを使用した文章のベクトル化と, 先述した TF-IDF を用いた手法とを比較して実験を行い, 本研究の目的に対してより効果的な手法がどちらなのかを検証する.

5.6 提案手法の精度検証

前節までの提案手法によるシステムの妥当性, 及び精度を検証するため, 人手で評価したデータとの比較を行う. 人手による評価として, 5.2.1 節で述べたクリーニング処理を施した元のコメント文に対して, 動画で宣伝している商品やサービスに関連しているかどうかを人手でアノテーションし, ラベル付けを行う. アノテーションの基準として以下のようなルールに当てはまるコメントに「関連性-高」のラベルを付け, その他のコメントに「関連性-低」のラベルを付与し, 正解ラベルが付いたデータを作成する.

- 動画で宣伝している商品やサービスに直接関係している
- 商品やサービスに対する視聴者の意見・感情などを含んでいる
- 商品やサービスを宣伝している動画内容に関係している

「関連性-高 / 低」の二値分類を行った結果から,「関連性-高」ラベルを付けたコメントの件数を a,「関連性-低」ラベルを付けたコメントの件数を b とする. そして, 前節で文章間の類似度を計算し降順にソートしたテキストデータの上位 a 件を「関連性-高」と予測したデータ, 下位 b 件を「関連性-低」と予測したデータとみなし, 正解ラベルを付与したデータと予測データに対して Confusion Matrix(混同行列) を求める. Confusion Matrix とは, 二値分類問題で出力された結果をまとめた行列 (≒表) のことで, 機械学習モデルの性能を測る指標として用いられていることが多い. 本研究では,「関連性-高 / 低」の二値分類に関して, 提案手法により予測したデータを機械学習モデルで予測したデータとみなし, Confusion Matrix を求める.

表 12 が一般的な機械学習モデルにおける Confusion Matrix である. 行が正解のクラス (ラベル) を, 列が機械学習モデルで予測したクラス (ラベル) を表している. TP(True Positive) は Positive ラベルが付いているものを正しく「Positive」だと予測していて, 注目対象を正しく分類できる, また対処すべき注目事象を特定できることを表す. TN(True Negative) は Negative ラベルが付いているものを正しく「Negative」だと予測していて, 注目対象以外を正しく分類できる, また注目対象を見逃さず損失を避けられることを表す. FN(False Negative) は Positive ラベルが付いているものを誤って「Negative」だと予測していて, 注目対象を誤ってそれ以外に分類してしまう, また注目対象を見逃し利益の獲得を逃してしまうことを表す. FP(False Positive) は Negative ラベル

が付いているものを誤って「Positive」だと予測していて, 注目すべきではないものを誤って分類してしまう, また分類する事象によっては無駄なコストがかかることを表す. これらを使い, 機械学習モデルの性能を測る様々な指標を計算することができる.

表 12: Confusion Matrix

		機械学習モデルの予測	
		Positive	Negative
実際の正解ラベル	Positive	**TP(True Positive)** Positiveと判定し, それが正解	**FN(False Negative)** Positiveと判定したが, 実際はNegative
	Negative	**FP(False Positive)** Negativeと判定したが, 実際はPositive	**TN(True Negative)** Negativeと判定し, それが正解

以下に表 12 の Confusion Matrix の例から求められる様々な指標を示す.

Accuracy

正解率・正確度・精度などと呼ばれる, 全予測結果の中で正しい予測をしたもの (TP・TN) の割合のことである. 総合的なモデルの性能を示すのに用いられることが多いが, クラスに偏りがある場合 (例：Positive が極端に多く, Negative が極端に少ない), はモデルの性能を正しく評価できないこともある.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{16}$$

Error Rate

不正解率と呼ばれる, Accuracy の逆で全予測結果の中で誤った予測をしたもの (FP・FN) の割合のことである. Accuracy と同様に, クラスの偏りがある場合にはモデルの性能を正しく評価できないこともある.

$$\text{Error Rate} = \frac{FP + FN}{TP + FP + FN + TN} \tag{17}$$

Sensitivity・Recall

感度・再現率・検出率などと呼ばれる, 正解クラスが Positive であるとき, 予測モデルも Positive だと判定した割合のことである. 実際に正解クラス (例：癌の検出など) を見逃さないことが重要な事象の際に重視される指標である.

$$\text{Sensitivity・Recall} = \frac{TP}{TP + FN} \tag{18}$$

Specificity

特異度と呼ばれる, 正解が Negative であるとき, 予測モデルも Negative だと判定した割合のことである. 疫病検査の例では, 罹患していない人の結果が陰性となる率であり, 負のケースを正確に識別することが重要な事象の際に重視される.

$$\text{Specificity} = \frac{TN}{FP + TN} \tag{19}$$

Precision

適合率と呼ばれる, モデルが Positive と予測したときに実際にそれが Positive である割合のことである. 偽陽性 (誤った正の予測) を最小限に抑えたい事象のときに重視される指標である.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{20}$$

F1-measure

F1 値・F 値などと呼ばれる, Precision(適合率) と Recall(再現率) の調和平均のことである. 一般的に Precision と Recall の間にはトレードオフの関係があるが, そのバランスを取る必要がある事象のときに重視される指標である.

$$\text{F1-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{21}$$

このように Confusion Matrix を使用した機械学習モデルの性能評価には様々な指標が存在するが, どの指標がそのモデル・目的に対して最適な性能評価を行えるかの判断を行う必要がある. したがって, ここからは本研究の目的・提案手法においてどの指標が適切なのかを考えていく. 表 13 は本研究における Confusion Matrix を示している.

Accuracy(正解率) は, 全予測結果の中で関連性があるコメント及び関連性がないコメントをどれだけ正確に分類できているかの指標である. 関連性あり・なしのそれぞれのコメント数は動画内容・商品 (サービス)・視聴者層などによって大幅に異なるため, 偏りが発生しやすいと考えられる.

表 13: 提案手法の Confusion Matrix

		提案手法の予測	
		関連性がある コメント	関連性がない コメント
人手によって アノテーション したクラス	関連性がある コメント	**TP(True Positive)** 関連性ありと判定し, それが正解	**FN(False Negative)** 関連性ありと判定したが, 実際は関連性なし
	関連性がない コメント	**FP(False Positive)** 関連性なしと判定したが, 実際は関連性あり	**TN(True Negative)** 関連性なしと判定し, それが正解

つまり, Accuracy が高い場合でも, 特定の動画においてモデルが適切に性能を評価しているとは限らないといえる．Error Rate は Accuracy と同様にクラスの偏り次第では適切な評価が行えない場合がある．Sensitivity・Recall(感度・再現率) は人手で関連性があると判断したコメントのうち, 提案手法によって関連性があると予測されたコメントの割合である．この値が高いということは, 実際に関連性があるコメントの取りこぼしが少ないことを意味しているので, 本研究の提案手法を評価する指標として有効であると考えられる．Specificity(特異度) は Sensitivity・Recall の逆で, 人手で関連性がないと判断したコメントのうち, 提案手法によって関連性がないと予測されたコメントの割合である．本研究では商品・サービスとの関連性があるコメントを抽出する手法を提案しているため, Specificity は提案手法の評価をする指標として適切ではないと考えられる．Precision(適合率) は提案手法によって関連性があると予測されたコメントのうち, 人手で関連性があると判断したコメントの割合である．この値が高いということは, 提案手法によって抽出したコメントを他のユーザーが確認したとき, それが実際に商品との関連性があり, 商品を購入する際に参考になりうるコメントである可能性が高いことを意味している．実際に関連性があるコメントを取りこぼしている可能性はあるが,「他のユーザーが商品の購入判断材料にできるような関連性のあるコメントを抽出する」という本研究の提案手法を評価する指標として最適であると考えられる．F1-measure(F 値) は Precision と Recall の調和平均であり, どちらも提案手法の評価を行う指標として有効であるため, F1-measure を用いた総合的な評価も有効であると考えられる．したがって, 本研究の提案手法を評価する指標としては, Sensitivity・Recall, Precision, F1-measure を主に用いることとする．

第 III 部
実験

第Ⅲ部では第Ⅱ部で述べた本研究の提案手法を用いて実験を行った結果を述べる. 改めて, 本研究の研究目的は YouTube 上で商品やサービスを宣伝している動画のコメントを分析し, その動画で宣伝している商品やサービスに関連しているコメントを抽出するシステムの作成である. 第Ⅱ部で述べた通り, 対象のコメント集合に対して Biterm Topic Model(BTM) を用いて出現確率が高い単語を抽出し, 文章を生成する. ここで生成された文章は BTM によって推定したトピックに出現しやすい単語を用いているため, 動画内容に対して代表的な文章であると仮定することができ, その文章との類似度が高い元のコメントを商品との関連性が高いと判断することができるという仮説が立つ. この仮説の妥当性, 及びシステムの精度を検証するために人手でアノテーションしたデータとの Confusion Matrix を計算し, 様々な指標で分析を行った.

6　実データを用いた実験結果と考察

本節では実際の YouTube 上の動画からコメントを抽出し実験を行った結果, 及び結果から読み取れた提案手法の有効性・妥当性について述べる. 実験対象とした動画の選定には, 商品・サービスや料理のレシピなどを紹介・宣伝していることを条件としている. また, その中でもコメントの総数や視聴者層の違い, 宣伝しているチャンネルが企業であるか個人であるかの違いなど, 様々な条件下での動画を対象として実験を行うことで提案手法が成立する条件を考察する.

6.1　みそきん (カップラーメン) の紹介動画

YouTuber のヒカキンが自身で商品開発を行った「みそきん (味噌味のカップラーメン)」の概要を紹介している動画 [7](以下「みそきん」) からコメントを抽出した. 以下に詳細な実験条件を示す.

コメント件数

抽出したコメントは 1517 件であり, そこから前処理・分かち書きの過程で削除された文章 (絵文字のみの文章や一文字のみの文章) を除いた 1380 件を用いて BTM で分析した. 類似度計算に用いるコメントは前処理のみを施したコメント 1475 件を用いた.

前処理

前処理の形態素解析の段階では「名詞」と「形容詞」のみを抽出し分かち書きを行った.

BTM の各パラメータ

- トピック数 $T = 5$
- 単語分布から抽出する出現確率上位の単語数 $n = 10$
- 各文章ごとに考慮する単語数 $M = 10$
- 反復回数 iterations $= 20$
- ディリクレ分布のハイパーパラメータ α, β
  各ハイパーパラメータはスカラー値であり, 一般的に 0.01~1 の値をとる. 本研究では BTM の評価指標の perplexity を用いてハイパーパラメータを調整した. α, β をそれぞれ 0.01~1 で変化させ, perplexity が最低値をとった値で実験を行う.「みそきん」では $\alpha = 0.92, \beta = 0.14$ で実験を行った.

6.1.1 BTM によって抽出した単語リスト

上記の条件で実験を行い, BTM によって抽出した出力確率上位 10 単語を表 14 に示す. 表から読み取れる通り, 商品開発者である「ヒカキン」や商品の「ラーメン」, 味に関する「みそ」,「美味し」,「まずい」などの単語が BTM によって推定したトピックに出現しやすい単語であることが分かる. コメント集合を把握した上で他の単語を見ていくと, ラーメン自体が「好き」, 商品が「楽しみ」などの単語や, ヒカキンの商品開発に対する「努力」を「尊敬」しているコメント, ヒカキンのことが好きな視聴者からは「嬉しい」「報告」に喜んでいたり,「絶対」買うなどのコメントも多くみられ, 適切に単語を抽出できていることが分かった. トピック間で単語を見ていくと, Topic 0~3 では「ラーメン」,「ヒカキン」,「好き」,「味噌」などが複数のトピックで共通しており, 同じようなトピックを推定していると考えられる. また, 共通している単語以外でもほとんどの単語が商品に関連しているか商品や開発者に対する視聴者の感想・感情の単語であり, 商品に関連するトピックであることは明らかである. それに対して Topic 4 では「味噌」や「まずい」など味に関連する単語も存在するが,「康平」,「奥村」,「休日」など商品との関連性はない単語も多く含まれていて, 商品に関連しているトピックではないことが分かる.

表 14: BTM で抽出した単語リスト (みそきん)

	Topic 0	Topic 1	Topic 2	Topic 3	Topic 4
1	hikakin	ヒカキン	ラーメン	ラーメン	まずい
2	ヒカキン	ラーメン	ヒカキン	ヒカキン	康平
3	ラーメン	美味し	hikakin	絶対	奥村
4	好き	楽しみ	尊敬	楽しみ	味噌
5	すごく	応援	商品	hikakin	外食
6	嬉しい	報告	youtuber	大好き	スガキヤ
7	元気	みそ	好き	発売	ちゃんねる
8	絶対	hikakin	絶対	好き	休日
9	努力	お願い	報告	すごい	土日
10	味噌ラーメン	ラーメン屋	すごい	味噌ラーメン	お昼

6.1.2 文章生成

表 14 に示している単語を使って, ChatGPT(GPT-4) を用いて文章を生成した. ChatGPT には各トピックの単語と文章生成のためのプロンプトを入力する. 入力したプロンプトは以下のとおりである. 基本的には 5.4 節で述べた通りのプロンプトを入力するが, 対象の動画にのみ必要なプロンプトがあれば加えて入力する. 今回の「みそきん」の場合では, 商品の開発者である「ヒカキン」,「hikakin」を人名として指定する必要がある.

- 空白区切りで入力する単語を使って文章を生成してください.
- 全ての単語を使用してください.
- YouTube のコメントにあるような文章を生成してください.

- 可能な限り短い文章を生成してください.
- “ヒカキン”, “hikakin” は人名として扱ってください.

上記のプロンプトと BTM で得られた各トピックの出現確率上位の単語を ChatGPT に入力して文章を生成した結果を表 15 に示す. AI による文章生成であるため, 人が作成した文章に比べて違和感を感じる部分が多い.「ヒカキン」と「hikakin」を重複せずに人名として使用していて, 他の単語はすべて使われている. 違和感は感じるが文章の意味は理解でき, YouTube のコメントに存在してもおかしくない文章が生成されている.

表 15: ChatGPT による文章生成結果 (みそきん)

	生成結果
Topic 0	ヒカキンさんがラーメンが好きなのを知って、すごく嬉しいです！ 彼の元気と努力が絶対味噌ラーメンのように強い味を出していますね！
Topic 1	ヒカキンさんが美味しいラーメンを楽しみにしているのを聞いて、応援する 気持ちが強まります。みそラーメンを食べる報告、お願いしますね！ ラーメン屋さんでの体験、楽しみにしています！
Topic 2	ヒカキンさんがラーメン好きなのは、尊敬する YouTuber としての素晴らしい 一面です。彼の商品や活動に絶対注目して、報告を待っています。 すごいですね！
Topic 3	ヒカキンさんがラーメンを絶対楽しみにしているのを聞いて、私も大好きな 味噌ラーメンが好きになりました。彼の新しい発売はいつもすごいですね！
Topic 4	康平さんと奥村さんが休日の土日に外食でスガキヤの味噌ラーメンを食べた けど残念ながらまずいと感じたようですね。 お昼のちゃんねるでその話を聞きました。

6.1.3 二種の類似度計算法による結果

ChatGPT(GPT-4) で生成した文章と, 前処理を施した元の YouTube のコメントとの文章間の類似度を計算した. 5.5 節で述べた通り, TF-IDF による文章のベクトル化と BERT による文章のベクトル化をそれぞれ行い, Cos 類似度を計算する. 二つの手法で計算した結果を比較し, それぞれがどのような特徴を持っているか考察する. 図 14,15 は各類似度計算法によって算出した文章間の類似度をグラフ化したものである. BERT を用いた類似度計算ではほとんどのコメントが 0.7∼0.96 の間の値をとっているのに対し, TF-IDF を用いた類似度計算では 0∼0.7 の値をとるコメントも多く存在した. 本研究では類似度計算した結果の数値が重要というわけではなく, 商品との関連性があるコメントの類似度が高く算出されることが重要なため, 図 14,15 からではどちらの手法が本研究において適切であるかは判断できない. しかし, 図 15 から TF-IDF を用いた類似度計算では低い類似度を示すコメントが多いことが分かり, そのコメントは文章の長さが短い傾向にあることが分かった (表 17). 反対に BERT を用いた類似度計算では長さが短い文章でも類似度が高いケースが存在していたため (表 16), YouTube のコメントのように短い文章を多く含むテキストデータに対する分析の場合, BERT による類似度計算の方が優れていると考えられる.

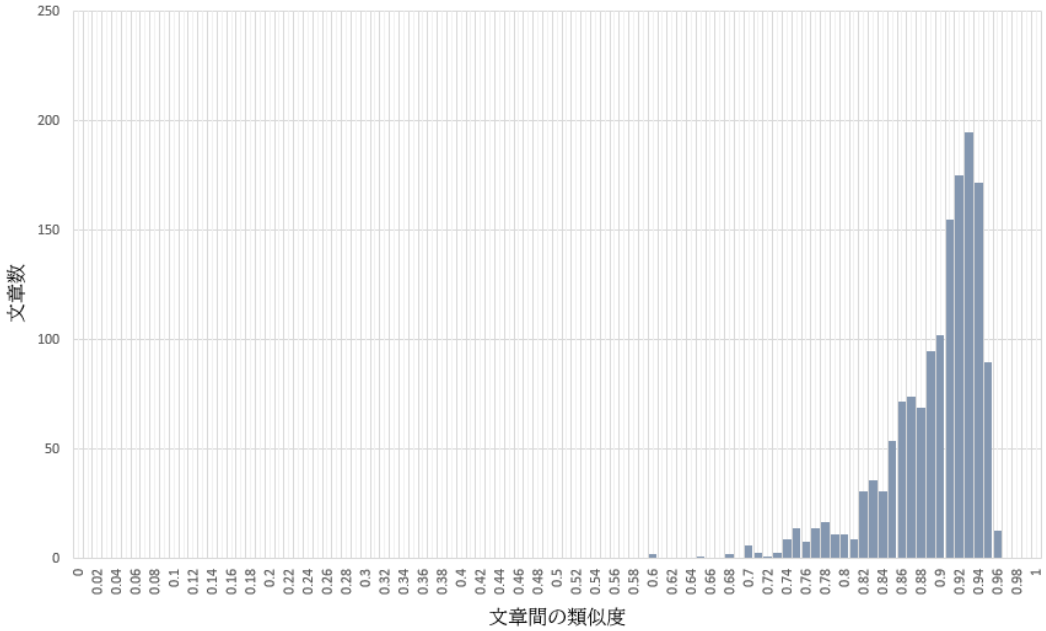

図 14: BERT による類似度計算結果

表 16: BERT による類似度計算結果の一例

類似度結果	前処理済みの元コメント
0.95739305	アレルギーの関係で小麦がたくさん食べれないのでメシ版が凄く嬉しい味噌大好きです絶対食べます
0.943691134	味噌ラーメンからもやしメンマネギ赤い調味油って感じがくるり意識してそうで好き
0.939244092	味噌ってのがいいですね食べてみます次あれば豚骨も期待してます
0.936367273	白味噌を入れることによってガツンとくる旨味が鈍くなる気がするけどどうなんだろうか早く買って食べたい
0.914627552	食べてみたいです楽しみ
0.905122876	今後色んな味出して欲しい
0.903024077	ファミマローソンでも販売して欲しい
0.901709378	それでも人工甘味料入ってる
0.890335798	味噌味以外も出してほしい

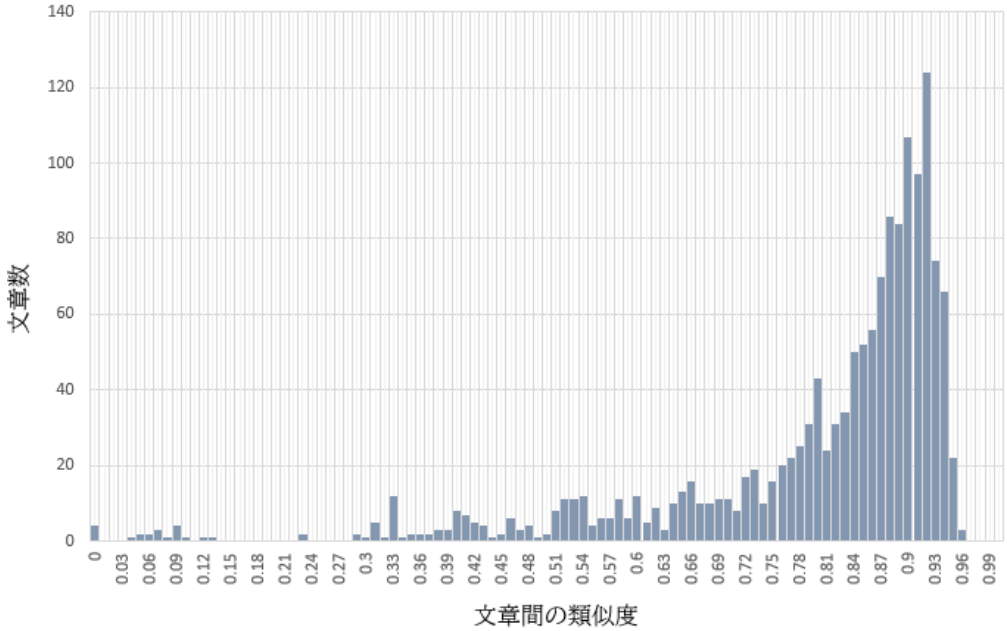

図 15: TF-IDF による類似度計算結果

表 17: TF-IDF による類似度計算結果の一例

類似度結果	前処理済みの元コメント
0.933934656	味噌ってのがいいですね食べてみます次あれば豚骨も期待してます
0.927453136	アレルギーの関係で小麦がたくさん食べれないのでメシ版が凄く嬉しい味噌大好きです絶対食べます
0.914770669	白味噌を入れることによってガツンとくる旨味が鈍くなる気がするけどどうなんだろうか早く買って食べたい
0.912356335	味噌ラーメンからもやしメンマネギ赤い調味油って感じがくるり意識してそうで好き
0.883447275	ファミマローソンでも販売して欲しい
0.831779144	今後色んな味出して欲しい
0.800387187	味噌味以外も出してほしい
0.710342807	食べてみたいです楽しみ
0.648517863	それでも人工甘味料入ってる

6.1.4 提案手法の精度検証結果

文章間の類似度を計算し, 値が高いコメントを「商品との関連性がある」と判断するという仮説を検証するため, 人手でアノテーションして正解ラベルを付与したデータと比較して分析を行う.「みそきん」のコメントに対するアノテーションは以下の基準で行った. アノテーションを行った結果,「関連性-高」のラベルが付与されたコメントが 934 件,「関連性-低」のラベルが付与されたコメントが 541 件であった.

関連性-高

- 商品である「みそきん」に直接関係するコメント (例：味, 値段, 具など)
- 商品の発売日, 販売場所についてのコメント
- 開発過程のエピソードに関するコメント
  (例：カップラーメンを開発するために全国のラーメンを食べていたことへの興味・関心など)
- 「みそきん」に対する視聴者の感想・感情・意見 (例：美味しそう, 絶対買う, 味噌以外も作ってほしいなど)
- ヒカキンのラーメン作りへのこだわり, 努力に関するコメント
  (他の視聴者の購買意欲に繋がる可能性があるため)

関連性-低

- ラーメン作りに関係ない, ただヒカキンを褒めているコメント
  (他の視聴者が読んでラーメン作り以外の情報しか得られないもの)
- 視聴者自身のことを述べているコメント
  (例：所謂自分語りや○○店のラーメンが好きです等のコメント)
- その他全く関係ないコメント

人手でアノテーションし正解ラベルを付与したデータと, 提案手法によって類似度を計算したコメントを比較し, 提案手法の精度, 及び仮説の検証を行う. 提案手法では類似度を計算しただけであり, 商品との関連性が高いと言える閾値は設定されていない. そのため, 類似度上位のコメントに対して, 人手で「関連性-高」のラベルを付与した数と同じ数のコメントだけ「関連性-高」のラベルを付与する. それ以下の類似度のコメントには「関連性-低」のラベルを付与する.「みそきん」の場合, 正解ラベルは「関連性-高」が 934 件,「関連性-低」が 541 件であり, 同じ数だけラベル付けしたデータを用いて, 二値分類の Confusion Matrix を計算する. しかしこの手法では Confusion Matrix の FP, FN の値が同じになり, Precision, Recall, 及び F1-measure が同じ値を示してしまう (式 (18), (20), (21)). そのため, 類似度上位 25%, 50%, 75%を閾値として「関連性-高」のラベルを付与したデータを作成し, 同様に Confusion Matrix を計算することで提案手法の精度をより深く考察する.

まず初めに, 人手の正解ラベルと同数のラベルを付与した結果について述べる. BERT を用いて類似度計算したデータから求めた Confusion Matrix を表 18 に, TF-IDF を用いて類似度計算したデータから求めた Confusion Matrix を表 19 に示す.

表 18: みそきん (BERT) の Confusion Matrix (正解ラベルと同数のラベル付与)

みそきん-BERT		提案手法の予測	
		関連性がある コメント	関連性がない コメント
人手によって アノテーション したクラス	関連性がある コメント	**TP = 648**	**FN = 286**
	関連性がない コメント	**FP = 286**	**TN = 255**

表 18 の Confusion Matrix から各指標を計算すると以下のような結果が得られる．この結果より,「みそきん」に対する提案手法の全体的な正解率は Accuracy = 0.6122 であることが分かった. Recall, Precision, F1-measure = 0.6938 であり, 提案手法で予測した関連性があるコメントが約 7 割正解していること, また人手で関連性があると判断したコメントのうち約 7 割が提案手法でも関連性ありと予測されていることが分かる.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} = 0.612203... \tag{22}$$

$$\text{Sensitivity・Recall} = \frac{TP}{TP + FN} = 0.693790... \tag{23}$$

$$\text{Precision} = \frac{TP}{TP + FP} = 0.693790... \tag{24}$$

$$\text{F1-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \times TP}{2 \times TP + FP + FN} = 0.693790... \tag{25}$$

表 19: みそきん (TF-IDF) の Confusion Matrix (正解ラベルと同数のラベル付与)

みそきん-TF-IDF		提案手法の予測	
		関連性がある コメント	関連性がない コメント
人手によって アノテーション したクラス	関連性がある コメント	**TP = 608**	**FN = 326**
	関連性がない コメント	**FP = 326**	**TN = 215**

表 19 の Confusion Matrix から各指標を計算すると以下のような結果が得られる．TF-IDF を用いて類似度計算したデータを用いた場合全体の正解率は 0.5580 となり, BERT を用いて類似度計算したデータよりも低い値を示した．また, Recall, Precision, F1-measure も同様に BERT に比べて低く, 提案手法で予測したコメントのうち約 6.5 割が実際に関連性があるコメントであること, また人手で関連性ありと判断したコメントのうち約 6.5 割が提案手法でも関連性ありと予測された結果となった．

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} = 0.557966... \tag{26}$$

$$\text{Sensitivity・Recall} = \frac{TP}{TP + FN} = 0.650963... \tag{27}$$

$$\text{Precision} = \frac{TP}{TP + FP} = 0.650963... \tag{28}$$

$$\text{F1-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \times TP}{2 \times TP + FP + FN} = 0.650963... \tag{29}$$

次に, 各類似度計算法によって計算した文章間の類似度の上位 25%, 50%, 75%に対して「関連性-高」のラベルを付与して同様の実験を行った結果を表 20 に示す．Accuracy を見ると, 閾値をどの値に設定しても全体的な正解率は 50%を超えていることが分かる．Precision は閾値を低く設定するにつれて値が低くなるが, 一番低い 75%の場合でも 0.66 あり, どの閾値に設定した場合でも提案手法で予測したコメントのうち約 6.6 割以上が正解していることが分かる．閾値を低く設定するにつれて Precision が低くなるのは, 誤って「関連性-高」と判断するコメントが増えるからであると考えられる．Precison とは逆に Recall は閾値を低く設定するにつれて値が高くなった．これは「関連性-高」と判断するコメントが多くなると, その取りこぼし (「関連性-高」を「関連性-低」と判断する事象) が少なくなるからであると考えられる．Recall の値は閾値 25%のときに約 3 割であり, 提案手法の性能としては満足できない結果であるが, 閾値 75%のときは約 8 割で取りこぼしなく関連性があるコメントを抽出できていることが分かる．

表 20: 「関連性-高」の閾値を上位 25%, 50%, 75%に設定した結果 (BERT)

評価指標＼閾値	25% (369件)	50% (737件)	75% (1106件)
Accuracy	0.503050...	0.578983...	0.614915...
Precision	0.772357...	0.712347...	0.665461...
Recall	0.305139...	0.562098...	0.788008...
F1-measure	0.437452...	0.628366...	0.721568...

F1-measure は Precision と Recall の調和平均であり, Accuracy と同様に全体的な提案手法の性能を測ることができる. 閾値 25%では 5 割を切っているため閾値として適切ではないことが分かる. 閾値 75%のときに F1-measure は最大となるが, Precision は閾値設定を行わずに正解ラベルと同数のラベルを付与したときの方が値が高いため, Precision と F1-measure(Recall) のどちらを優先するかは考察の余地がある.

次に, TF-IDF を用いて類似度計算したデータに閾値を設定してラベルを付与した結果から計算した各指標を表 21 に示す. BERT での結果 (表 20) と比較すると, 対応する全ての値が低いことが分かる. また, 閾値を低くするにつれて Precision が低くなることや, Recall・F1-measure が高くなることなどは一致しており, 類似度計算法による差異は値のみであることが分かった. そのため, 本研究に対してより有効な類似度計算法は BERT で文章をベクトル化し, Cos 類似度を計算する方法であると考えられる.

表 21: 「関連性-高」の閾値を上位 25%, 50%, 75%に設定した結果 (TF-IDF)

評価指標＼閾値	25% (369件)	50% (737件)	75% (1106件)
Accuracy	0.490847...	0.522033...	0.567457...
Precision	0.747967...	0.655359...	0.633815...
Recall	0.295503...	0.517130...	0.750535...
F1-measure	0.423637...	0.578096...	0.687254...

6.1.5 LDA との比較

提案手法で用いている BTM が代表的なトピックモデルである LDA と比べて優れているかどうかを検証する. BTM を用いてトピック推定, 及び単語抽出を行った過程 (6.1.1 節) を LDA で行い, その後は同様の手法で実験を行った. 表 22 に LDA を用いてトピック推定を行い, トピックごとの出現確率上位 10 単語を抽出した結果を示す. BTM と同様に商品に関連している単語を多く抽出

表 22: LDA で抽出した単語リスト (みそきん)

	Topic 0	Topic 1	Topic 2	Topic 3	Topic 4
1	ラーメン	0	0	まずい	絶対
2	ヒカキン	ラーメン	ヒカキン	ヒカキン	hikakin
3	好き	ー	ヒカキン	夢	ラーメン
4	夢	嬉しい	夢	美味しかっ	美味し
5	hikakin	味	姿	すごい	夢
6	凄い	ラーメン	キン	ラーメン	楽しみ
7	努力	みそ	売り切れ	youtube	0
8	商品	報告	尊敬	努力	セブン
9	尊敬	美味しかっ	hikakin	尊敬	大好き
10	凄	コンビニ	嬉しい	hikakin	発売

できているが, BTM とは異なり「0」や「ー」など, 数字や記号がいくつか抽出されている結果となった. その後, 表 16 の単語を用いて文章を生成し, BERT による文章のベクトル化を用いた類似度計算を行った. 6.1.4 節で元のコメントに正解ラベルを付与したデータと, 同じ件数のラベルを付与した予測データから Confusion Matrix と各指標を計算した結果を表 23 に示す. BTM を用いた手法の結果 (表 18) と比較すると, LDA を用いた手法では正しく予測している TP と TN が減り, 誤った予測をしている FP と FN が増加している. それに伴い, 計算した各指標も BTM の結果に比べて少し低くなっている. このことから, 短いテキストに対するトピック推定及び単語抽出は LDA よりも BTM の方が優れていること, また提案手法で用いるトピックモデルとしても BTM の方が優れていると考えられる.

表 23: LDA を用いた手法の Confusion Matrix

<table>
<tr><td colspan="2" rowspan="2">みそきん-LDA-BERT</td><td colspan="2">提案手法の予測</td></tr>
<tr><td>関連性がある
コメント</td><td>関連性がない
コメント</td></tr>
<tr><td rowspan="2">人手によって
アノテーション
したクラス</td><td>関連性がある
コメント</td><td>TP = 643</td><td>FN = 291</td></tr>
<tr><td>関連性がない
コメント</td><td>FP = 291</td><td>TN = 250</td></tr>
<tr><td colspan="2">Accuracy</td><td colspan="2">0.605423...</td></tr>
<tr><td colspan="2">Precision</td><td colspan="2">0.688436...</td></tr>
<tr><td colspan="2">Recall</td><td colspan="2">0.688436...</td></tr>
<tr><td colspan="2">F1-measure</td><td colspan="2">0.688436...</td></tr>
</table>

6.2 豚汁のレシピ紹介動画

料理研究家のリュウジが自身の YouTube チャンネルで投稿した豚汁のレシピ, 及び調理過程の動画 [8](以下「豚汁」) からコメントを抽出した. 以下に詳細な実験条件を示す.

コメント件数

抽出したコメントは 1337 件であり, そこから前処理・分かち書きの過程で削除された文章 (絵文字のみの文章や一文字のみの文章) を除いた 1306 件を用いて BTM で分析した. 人手によるアノテーション, 及び類似度計算に用いるコメントは前処理のみを施したコメント 1329 件を用いた.

前処理

前処理の形態素解析の段階では「名詞」と「形容詞」のみを抽出し分かち書きを行った.

BTM の各パラメータ

- トピック数 $T = 5$

- 単語分布から抽出する出現確率上位の単語数 $n = 10$
- 各文章ごとに考慮する単語数 $M = 10$
- 反復回数 iterations $= 20$
- ディリクレ分布のハイパーパラメータ α, β
  他の実験と同じ手法でハイパーパラメータを調整した.「豚汁」では $\alpha = 0.93, \beta = 0.1$ で実験を行った.

6.2.1 BTM によって抽出した単語リスト

上位の条件に従って実験を行い, BTM によって抽出した出力確率上位 10 単語を表 24 に示す. 表 24 から読み取れる通り, 動画投稿者の料理研究家である「リュウジ」や調理している「豚汁」, 豚汁の具材である「ゴボウ」,「大根」, 豚汁の味にかかわる「味噌」,「ニンニク」などの単語が BTM によって推定したトピックに出現しやすい単語であることが分かる. コメント集合を把握した上で他の出力されている単語を見ると, このレシピが「美味しい」や「最高」などの視聴者の感情を表している単語や, この動画を見て「家族」に「今日」「作りました」などの単語が多く出力されているため, 適切に単語を抽出できていることが分かった. トピック間で単語を見ていくと, Topic 0~3 では「豚汁」,「リュウジ」,「美味しい」,「レシピ」が共通して出現していて, リュウジが紹介しているレシピで豚汁を作った感想に関するトピックが推定されていると考えられる. Topic 4 ではほぼすべての単語が豚汁の具や味など, レシピの内容に関わる単語を抽出しているため, 豚汁のレシピに関するトピックであると判断することができる.

表 24: BTM で抽出した単語リスト (豚汁)

	Topic 0	Topic 1	Topic 2	Topic 3	Topic 4
1	豚汁	豚汁	豚汁	料理	豚汁
2	レシピ	美味しい	至高	リュウジ	野菜
3	リュウジ	美味しく	レシピ	動画	ゴボウ
4	美味しい	リュウジ	リュウジ	最高	味噌
5	料理	味噌	大好き	豚汁	白だし
6	動画	生姜	今日	美味しい	生姜
7	作りました	味噌汁	美味しかっ	好き	ニンニク
8	最高	作りました	美味しい	兄さん	料理
9	美味しく	レシピ	料理	レシピ	大根
10	今日	ニンニク	シリーズ	家族	ネギ

6.2.2 文章生成

表 24 に示している単語を使って, ChatGPT(GPT-4) を用いて文章を生成した結果を示す. ChatGPT に入力したプロンプトは以下の通りである.「豚汁」では料理研究家の「リュウジ」を人名とする以外は 5.4 節で述べた通りのプロンプトを入力した.

- 空白区切りで入力する単語を使って文章を生成してください.

- 全ての単語を使用してください.
- YouTube のコメントにあるような文章を生成してください.
- 可能な限り短い文章を生成してください.
- “リュウジ” は人名として扱ってください.

上記のプロンプトと BTM で得られた各トピックの出現確率上位の単語を ChatGPT に入力して文章を生成した結果を表 25 に示す.「みそきん」の文章生成の結果 (表 15) と比較すると, 文章がより自然で伝わりやすく, さらに短い文章が生成されていると思われる. これは単語抽出結果 (表 24) も「みそきん」より対象の動画と関連している単語を抽出できていることに起因していると考えられる.

表 25: ChatGPT による文章生成結果 (豚汁)

	生成結果
Topic 0	今日、リュウジの料理動画を見て豚汁のレシピを作りました！ 最高に美味しくて、美味しいです！
Topic 1	リュウジのレシピで作った豚汁、美味しい！ 生姜とニンニクが効いた味噌汁、本当に美味しくて最高！
Topic 2	今日、リュウジのレシピで豚汁を作りました。 至高の味で、大好きな料理シリーズがまた一つ増えました！ 美味しかった！
Topic 3	リュウジ兄さんの動画を見て料理した豚汁、家族にも大好評！ 本当に美味しいレシピでした。最高です！
Topic 4	豚汁にゴボウ、大根、ネギをたっぷり使い、味噌と白だしで味付け。 生姜とニンニクで風味豊かな料理になりました！

6.2.3 二種の類似度計算法による結果

ChatGPT(GPT-4) で生成した文章と, 前処理を施した元の YouTube のコメントとの文章間の類似度を計算した結果を述べる.「みそきん」と同様に BERT と TF-IDF による文章のベクトル化を用いて Cos 類似度を計算し, 二つの手法がどのような特徴を持っているかを考察する. 図 16, 17 は各類似度計算法によって算出した文章間の類似度をグラフ化したものである.「みそきん」の結果と同様に, BERT を用いた手法ではほとんどのコメントが 0.7∼0.96 の値を取っている. しかし, TF-IDF を用いた手法では「みそきん」と比べて 0∼0.7 の値を持つコメントの数が少ないことが図 17 から分かった. これは「豚汁」の文章生成の精度が「みそきん」に比べて高いことと,「豚汁」の動画に対する元のコメントが「みそきん」に比べて商品 (レシピ) に関連するものが多いことが起因していると考えられる. 表 27, 26 はそれぞれの類似度計算法の結果の一例であり, 後に人手でアノテーションする際には「関連性-高」と判断されるコメントである. 表 26 では長さが短い文章のときに類似度が低くなってしまっているが, 同じコメントに対して表 27 では類似度が高く算出されている. この結果より, BERT を用いた手法の方が短い文章でも適切に文章間の類似度を測れているといえる.

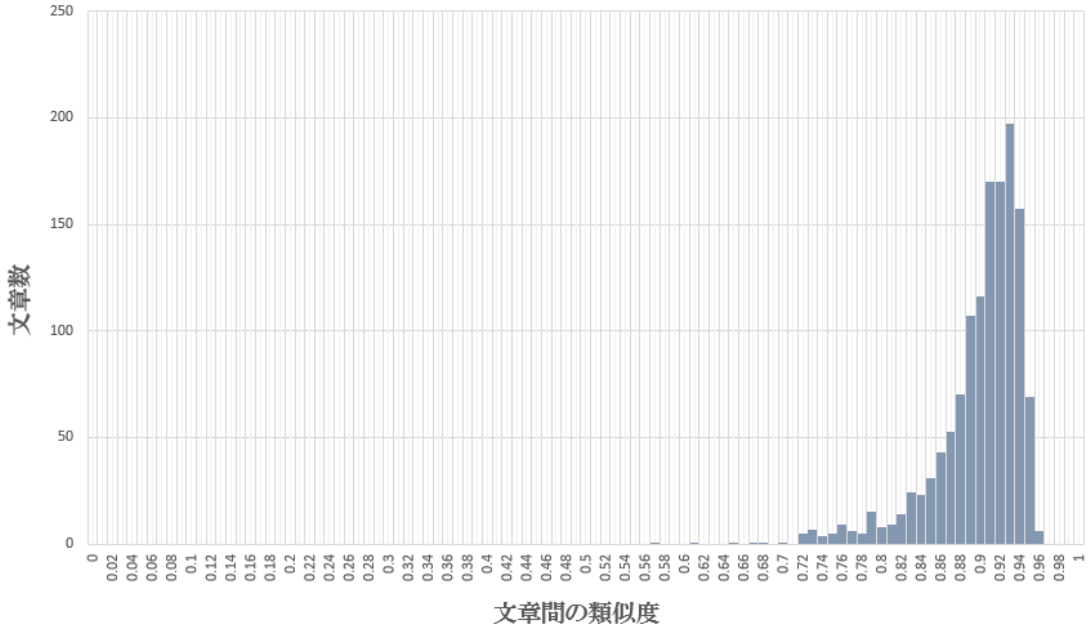

図 16: BERT による類似度計算結果

表 26: BERT による類似度計算結果の一例

類似度結果	前処理済みの元コメント
0.955579638	このレシピのお陰で豚肉入り味噌汁から卒業出来ました流石至高ですね 美味しいし娘に伝授します
0.945638359	今日このレシピで豚汁作ったら家族に大好評でしたすごく美味しかったです
0.938940465	ほんとに美味しい最後のニンニク最高今からの季節これで決まりいつも 美味しいレシピありがとうございます
0.93707478	ゴボウがこんなに美味しいなんて豚汁のごぼうは鬼門だったのですが香り 高くて味が染みて豚の脂も味噌も吸って美味しい目からウロコでした ゴボウ探して食べました至高シリーズまた待ってます
0.930768073	これは美味いニンニクと生姜が良かったです
0.92407757	これ本当に美味しくてリピートしまくってます
0.886364281	さつまいも入れても美味しかった
0.842903078	ニンニクが強いかも

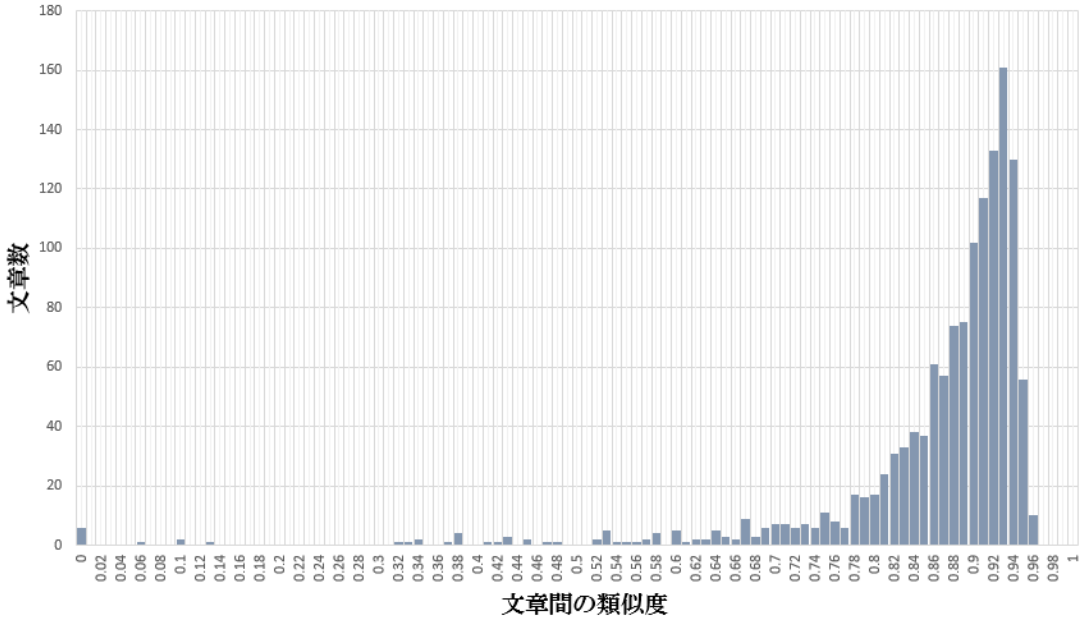

図 17: TF-IDF による類似度計算結果

表 27: BERT による類似度計算結果の一例

類似度結果	前処理済みの元コメント
0.940920903	このレシピのお陰で豚肉入り味噌汁から卒業出来ました流石至高ですね 美味しいし娘に伝授します
0.926962319	ゴボウがこんなに美味しいなんて豚汁のごぼうは鬼門だったのですが香り 高くて味が染みて豚の脂も味噌も吸って美味しい目からウロコでした ゴボウ探して食べました至高シリーズまた待ってます
0.910778972	今日このレシピで豚汁作ったら家族に大好評でしたすごく美味しかったです
0.896905186	ほんとに美味しい最後のニンニク最高今からの季節これで決まりいつも 美味しいレシピありがとうございます
0.889494581	これは美味いニンニクと生姜が良かったです
0.813619133	これ本当に美味しくてリピートしまくってます
0.676619315	さつまいも入れても美味しかった
0.594438816	ニンニクが強いかも

6.2.4 提案手法の精度検証結果

文章間の類似度を計算し, その値が高いコメントを「商品 (豚汁のレシピ) との関連性がある」と判断するという仮説を検証するため, 人手でアノテーションして正解ラベルを付与したデータと比較して分析を行う.「豚汁」のコメントに対するアノテーションは以下の基準で行った. アノテーションを行った結果,「関連性-高」のラベルが付与されたコメントが 882 件,「関連性-低」のラベルが付与されたコメントが 447 件であった.

関連性-高

- 紹介している豚汁のレシピに直接関係するコメント (例：具, 味付け, 調味料)
- 実際に豚汁を作った視聴者のレシピに対する感想 (例：美味しかった, 家族に喜んでもらえた)
- アレンジをした視聴者のコメント (例：里芋を入れた方が美味しかった)
- 調理方法に関連するコメント (例：ねぎの切り方, 茹で時間)

関連性-低

- レシピには関係ないが動画内容に関係しているコメント (例：飲酒しながら撮影していることに対するコメント)
- 動画投稿者の他の料理動画に対するコメント (例：○○のレシピを教えてほしい)
- 視聴者自身のことを述べているコメント (例：ゴボウ買い忘れた, ハイボール飲みたくなる)
- その他全く関係ないコメント

人手でアノテーションし正解ラベルを付与したデータと, 提案手法によって類似度を計算したコメントを比較し, 提案手法の精度, 及び仮説の検証を行う. 提案手法では類似度を計算しただけであり, 商品との関連性が高いと言える閾値は設定されていない. そのため, 類似度上位のコメントに対して, 人手で「関連性-高」のラベルを付与した数と同じ数のコメントだけ「関連性-高」のラベルを付与する. それ以下の類似度のコメントには「関連性-低」のラベルを付与する.「豚汁」の場合, 正解ラベルは「関連性-高」が 882 件,「関連性-低」が 447 件であり, 同じ数だけラベル付けしたデータを用いて, 二値分類の Confusion Matrix を計算する. また,「みそきん」と同様に類似度上位 25%, 50%, 75%を閾値として「関連性-高」のラベルを付与したデータを作成し, 同様に Confusion Matrix を計算することで提案手法の精度をより深く考察する.

まず初めに, 人手の正解ラベルと同数のラベルを付与した結果について述べる. BERT を用いて類似度計算したデータから求めた Confusion Matrix を表 28 に, TF-IDF を用いて類似度計算したデータから求めた Confusion Matrix を表 29 に示す. 表 28 より,「豚汁」に対する提案手法の全体的な正解率は Accuracy=0.703536 であり,「みそきん」よりも正解率が高いことが分かった. また, Precision, Recall, F1-measure=0.776643 であり, 同様に「みそきん」よりも高い精度でコメントを分類できていることが分かった. Precision が 0.776643 であることから, 提案手法で予測したコメントのうち約 7.7 割が人手で関連性があると判断したコメントであることが分かり, 提案手法に有用性があると判断することができると思われる. Recall も同様に 0.776643 であり, 人手で関連性があると判断したコメントのうち約 7.7 割が提案手法で関連性があると予測されているため, 他のユーザーにとって有益なコメントの取りこぼしが少ないといえる.

表 28: 豚汁 (BERT) の Confusion Matrix (正解ラベルと同数のラベル付与)

豚汁-BERT		提案手法の予測	
		関連性がある コメント	関連性がない コメント
人手によって アノテーション したクラス	関連性がある コメント	**TP = 685**	**FN = 197**
	関連性がない コメント	**FP = 197**	**TN = 250**
Accuracy		0.703536...	
Precision		0.776643...	
Recall		0.776643...	
F1-measure		0.776643...	

表 29 より, TF-IDF を用いて類似度計算をしたデータを用いた場合の全体の正解率は Accuracy=0.665914 であり,「みそきん」と同様に BERT を用いた手法よりも正解率が下がることが分かった. 他の指標も BERT を用いた手法より少し低い結果となったが,「みそきん」の同じ条件の結果と比較すると 10%程度高くなっているため, 文章生成がより自然で短いことが重要であると考えられる.

表 29: 豚汁 (TF-IDF) の Confusion Matrix (正解ラベルと同数のラベル付与)

豚汁-TF-IDF		提案手法の予測	
		関連性がある コメント	関連性がない コメント
人手によって アノテーション したクラス	関連性がある コメント	**TP = 660**	**FN = 222**
	関連性がない コメント	**FP = 222**	**TN = 225**
Accuracy		0.665914...	
Precision		0.748299...	
Recall		0.748299...	
F1-measure		0.748299...	

次に, 各類似度計算法によって計算した文章間の類似度の上位 25%, 50%, 75%に対して「関連性-高」のラベルを付与して同様の実験を行った結果を表 30 に示す. Accuracy を見ると, 閾値をどの値に設定しても全体的な正解率は 50%を超えていて, 閾値を下げたときの正解率の上昇率が「みそきん」の結果 (表 20) よりも高くなっていることが分かる. Precision は閾値を低く設定するにつれて値が低くなるが, 一番低い 75%の場合でも 0.76 あり, どの閾値に設定した場合でも提案手法で予測したコメントのうち約 7.6 割以上が正解していることが分かる. 閾値を低く設定するにつれて Precision が低くなるのは, 誤って「関連性-高」と判断するコメントが増えるからであると考えられる. Precison とは逆に Recall は閾値を低く設定するにつれて値が高くなった. これは「関連性-高」と判断するコメントが多くなると, その取りこぼし (「関連性-高」を「関連性-低」と判断する事象) が少なくなるからであると考えられる. Recall の値は閾値 25%のときに約 3 割であり, 提案手法の性能としては満足できない結果であるが, 閾値 75%のときは約 8.5 割で取りこぼしなく関連性があるコメントを抽出できていることが分かる. F1-measure を含め, 全ての指標で「みそきん」の結果よりも上回っていることが分かり, 改めて単語抽出と文章生成の結果が重要であると考えられる.

表 30: 「関連性-高」の閾値を上位 25%, 50%, 75%に設定した結果 (BERT)

評価指標 \ 閾値	25% (332件)	50% (664件)	75% (996件)
Accuracy	0.534988...	0.656884...	0.726109...
Precision	0.897590...	0.820783...	0.760040...
Recall	0.337868...	0.617913...	0.858276...
F1-measure	0.490939...	0.705045...	0.806176...

次に, TF-IDF を用いた類似度計算したデータに閾値を設定してラベルを付与した結果から計算した Confusion Matrix の各指標を表 31 に示す. BERT での結果 (表 30) と比較すると,「みそきん」と同様に対応している全ての指標が低いことが分かる. ここまでの実験で BERT を用いた類似度計算法による結果が TF-IDF を用いた類似度計算による結果を全て上回っているため, 本提案手法では BERT による文章のベクトル化の方が優れているといえる. ただし, 元コメントの文章数や文章の長さによって結果が変わる可能性があるため, 今後様々な条件を持つコメント集合に対して実験を行う必要がある.

表 31: 「関連性-高」の閾値を上位 25%, 50%, 75%に設定した結果 (TF-IDF)

評価指標 \ 閾値	25% (332件)	50% (664件)	75% (996件)
Accuracy	0.510910...	0.626787...	0.679458...
Precision	0.849397...	0.790662...	0.728915...
Recall	0.319727...	0.595238...	0.823129...
F1-measure	0.464579...	0.679172...	0.773162...

7　考察

6 節で述べてきた実験結果より, 提案手法の有用性, 及び仮説の検証結果についての考察をまとめる.

7.1　BTM の優位性についての考察

BTM によるトピックの推定とそのトピックの単語分布から出現確率上位の単語を抽出した結果より, BTM を用いることで YouTube のような一文の長さが比較的短い文書集合からでもトピックを適切に推定できることが分かった. また, 動画の主題 (宣伝している商品やサービス, レシピ) に関連する単語を主に抽出していたため, YouTube のコメント集合から動画の主題を推定する手法としてトピックモデルが有用であると考えられる. 従来のトピックモデルである LDA による実験では, BTM に比べて抽出する単語に記号や数字が含まれていたり, 抽出した単語の被りが多いことから文章生成の質が落ちていると思われる. その結果全ての指標で BTM より少し劣っていることが分かったが, BTM の優位性を明らかにするには一文の長さが極端に短いコメントを多く含む動画や, 文章数が多い, または少ない場合で実験を行う必要があると考えられる.

7.2　文章生成に関する仮説の検証

BTM によってトピックの出現確率上位の単語を抽出し, それらの単語を用いて ChatGPT(GPT-4) で文章生成を行った. ここで生成した文章は, BTM によってコメント集合, つまり対象の YouTube の動画に関するトピックに対して代表的な文章であるという仮説を立てている. この仮説に妥当性がある場合, 生成した文章と元の YouTube のコメントとの文章間の類似度を計算したとき商品・サービスとの関連性が高いコメントの類似度が高く算出されると考えられる. この仮説を検証するため, 人手でアノテーションしたデータと類似度計算したデータを比較した. その結果, 対象の動画によってぶれは生じるが Accuracy が 0.6∼0.7 程度, Precision が 0.7∼0.8 程度となり, 類似度を計算した結果に妥当性があるといえる結果となった. このことから, BTM によって抽出した単語を用いて ChatGPT により生成した文章は対象の動画のトピックに対して代表的な文章であるという仮説の妥当性が示されていると考えられる.

7.3　類似度計算方法の考察

BERT を用いて文章をベクトル化し Cos 類似度を計算する手法と, TF-IDF を用いて文章をベクトル化し Cos 類似度を計算する手法の二種類の類似度計算法を比較し, どちらが本研究の提案手法として優れているかを考察する. 各実験の類似度計算の結果より, 商品との関連性が高いコメントの類似度が BERT を用いた手法では適切に高い値を示していたのに対し, TF-IDF では低い値を示しているコメントが多く存在した. 特に文章の長さが短い場合に適切な値をとらないコメントが多いと思われる. また, Confusion Matrix より算出した各指標から, BERT を用いた手法の方が TF-IDF を用いた手法よりも Accuracy や Precision が高い結果となった. このことから, 本研究の提案手法で用いる類似度計算法としては BERT を用いた手法の方が優れているといえる. しかし, 一文の長さが極端に短いコメントが多い場合や対象の動画に対するコメントの傾向, 例えば気軽に投稿できる雰囲気であるかよく考えられた文章の割合が多いコメント欄であるかどうかにより結果が異なる可能性があるため, 様々な条件下での実験を行い BERT の優位性を明らかにする必要があると考えられる.

7.4 提案手法の有用性についての考察

YouTube 上で自社製品やサービスを宣伝している動画に対するコメントのうち, 対象の商品・サービスに関連しているコメントを抽出する提案手法の有用性, 実用性について考察する. 各実験の Confusion Matrix より算出された評価指標から, 提案手法の全体的な正解率 (Accuracy) は約 6 割~7 割程度であることが分かり, 関連性の有無の分類をある程度行えているといえる. しかし, Accuracy はラベルの偏りが著しい場合に適切な評価を行えないため, ラベルの偏りが少ないコメント集合を持つ動画に対してのみ有効な指標であると考えられる. Precision は予測モデルが Positive だと判断した事象のうち正解が Positive である事象の割合であり, 本研究においては提案手法によって「関連性-高」と判断されたコメントのうち人手で「関連性-高」と判断できるコメントの割合である. 各実験結果より Precision は 0.7~0.8 程度の値を示した. このことから, 提案手法によって関連性が高いと判断したコメントのうち約 7 割~8 割が正解しているため, 提案手法の有用性が示されていると考えられる. また, Recall は人手で関連性ありと判断したコメントが提案手法によってどれくらい関連性ありと判断されるかの指標であり, 他のユーザーにとって有益な情報をどれだけ取りこぼさないかという意味でもある. 関連性があると判断する閾値を人手のラベルと同数に設定した場合, Recall は Precison と同じ値になるため, 提案手法によって YouTube の元コメントから有益なコメントを抽出できることも示されていると考えられる. 最後に, 閾値を上位 25%, 50%, 75%に設定した場合の結果について考察する. 人手のラベルと同数に設定した結果と比較すると, 75%に設定したときに Accuracy, 及び F1-measure が高くなった. しかし, 提案手法の精度を検証する際に一番適切な Precision に関しては人手のラベルと同数に設定したときの方が高いため, 本研究の目的達成には人手のラベルと同数に設定する手法が適切であると考えられる.

第IV部
おわりに

8　まとめ

本研究では YouTube 上で商品やサービスを宣伝している動画のコメントから対象の商品やサービスに関連している視聴者のコメントを抽出するシステムの作成を目指した. 提案手法として, まず Biterm Topic Model(BTM) を用いて元のコメント集合からトピックの推定, 及び単語分布の推定を行い, 各トピックの出現確率上位の単語を抽出した. 次に抽出した単語を基に文章を生成し, 元のコメントとの文章間の類似度を計算した. 類似度の計算には BERT を用いた文章のベクトル化と, TF-IDF を用いた文章のベクトル化を用いて Cos 類似度を計算する二種類の手法で行った. 最後に類似度計算した結果を降順にソートし, 類似度上位のコメントを商品やサービスとの関連性があるコメントと判断するという手法を提案した. この提案手法の精度を検証するため, 人手で正解ラベルを付与したデータと類似度計算したデータとを比較し,「関連性あり」と「関連性なし」の二値分類とみなして Confusion Matrix を計算した.

実コメントを用いた実験結果より, BTM を用いることで YouTube のコメントのような一文が比較的短い文書集合からでもトピックの推定が行えることを確認した. また, 提案手法に用いる類似度計算法は BERT による文章のベクトル化から Cos 類似度を計算する手法が TF-IDF による手法と比べて優れていることが分かった. Confusion Matrix から算出した評価指標より, 提案手法の全体の正解率が約 6 割~7 割程度であることが分かった. Precision が約 0.7~0.8 程度の値を示したため, 提案手法によって関連性があると予測されたコメントのうち約 7 割~8 割が実際に商品との関連性があるコメントであること分かり, 提案手法の有用性が示されていると考えられる. また, 提案手法によって類似度計算を行い降順にソートしたデータに対して, 関連性があると判断する閾値を上位 25%, 50%, 75% , 人手のラベルと同数, に設定し各評価指標を比較した結果, 全体の正解率, 及び F1-measure が最も高くなるのが上位 75%に閾値を設定したときであることが分かった. 提案手法の実用性を高める上では Precision が重要な指標となるため, Accuracy や F1-measure とのバランスを考慮したうえで Precision が高い人手のラベルと同数に閾値を設定する手法が望ましいと考えられる.

9　今後の課題

Biterm Topic Model によって抽出した単語から文章を自動生成する際, 提案手法では ChatGPT(GPT-4) による手法で行っているが, より良い手法を提案することが重要であると考えている. 例えば, 抽出した単語を基に文章を生成するプロセスにコーパス (YouTube のコメント集合) を学習したデータを取り入れることで, より対象の動画と関連している文章を生成することができると思われる.

BERT を用いて文章をベクトル化して Cos 類似度を計算する手法と TF-IDF による手法を比較し BERT の優位性を示したが, 一文の長さが短い傾向にあるコメント集合の場合などに異なる結果が生じる可能性があるため, 様々な条件下での実験を行い, BERT を用いた手法の優位性を明らかにするとともに提案手法が成立する条件を明確にする必要があると考えられる.

提案手法の精度を検証する過程で人手によるアノテーションを行うが, 本研究では個人によるアノテーションで実験を行ったため, 基準が曖昧になることが多々あった. そのため, 複数人で基準をすり合わせながらアノテーションを行うことでより質のいい正解データを作成することができるようになると考えられる.

本論文で述べている実験結果では提案手法の有用性を示せているが, 提案手法が有効な条件を明確にする必要がある. 一文の長さが短い傾向にあるコメント集合の場合以外にも, 例えばコメントの総数が極端に多い, または少ない場合や, 企業チャンネルと YouTuber の動画に対するコメントの質の違いなど, 様々な条件のコメント集合に対して実験を行い, 提案手法が有効な条件を明確にすることが重要であると考えている. 加えて, 提案手法が適用できる範囲の拡大と実用化に向けたシステムの改善を行っていくことが今後の課題として挙げられる.

謝辞

本研究の遂行にあたり, 指導教官として終始多大なご指導を賜った, 東京都立大学大学院システムデザイン研究科電子情報システム工学域 相馬 隆郎 准教授に深謝致します。計算機応用工学研究室の皆様には, 本研究の遂行にあたり多大なご助言, ご協力頂きました. ここに感謝の意を表します。

参考文献

[1] 市川 知春, 武田 和大, 原 崇：「機械学習を用いた自然言語処理による商品レビューの評価」, 日本シミュレーション学会論文誌, Vol.13, No.2, pp.83-91, 2021

[2] 東 和幸, 高橋 仁, 中川 博之, 土屋 達弘：「単語の出現頻度と類似性に基づいたトピックモデル洗練化手法」, コンピュータソフトウェア, Vol.36, No.4, pp.25-31, 2019.

[3] 谷口 佑子, 津田 和彦：「テキストマイニングを用いた口コミ分析による点数評価の信頼性確認手法」, 人工知能学会, Vol.31, 3A1-4, 2017.

[4] 吉見 憲二：「グルメサイトにおけるクチコミの信頼性確保に関する一考察」, IPSJ SIG Technical Report, Vol.2014-DPS-161 No.2, Vol.2014-EIP-65 No.2.

[5] Blei, M. David., Ng, Y. Andrew., and Jordan, I. Michael : “Latent Dirichlet Allocation”, Journal of Machine Learning Research 3, pp.993-1022, 2003.

[6] Xiaohui Yan, Jiafeng Guo, Yanyan Lan, Xueqi Cheng : “A Biterm Topic Model for Short Texts”, WWW ’13 Proceedings of the 22nd international conference on World Wide Web, pp.1445-1456, 2013.

[7] 王 博, 服部 隆志, 萩野 達也：「Twitter における口コミ情報の抽出と分析」, 情報処理学会第 81 回全国大会, No.2, pp.139-140, 2019.

[8] 堺 雄之介, 伊藤 栄典：「動画サイトにおける視聴者コメントの特徴抽出」, 人工知能学会, 知識ベースシステム研究会, Vol.124, pp.17-22, 2021.

[9] 岡本 一志, 柴田 淳司：「過去の商品レビューに関する類似性分析」, 日本知能情報ファジィ学会, Vol.36, pp.355-356, 2020.

[10] 西田 有輝, 楊 添翔, 山下 遥, 後藤 正幸：「強調データの拡張学習による Biterm Topic Model の解釈性向上法に関する一考察」, 人工知能学会全国大会, Vol.36, 2022.

[11] 國府 久嗣, 山崎 治子, 野坂 政司：「内容推測に適したキーワード抽出のための日本語ストップワード」, 日本感性工学学会, Vol.12, No.4, pp.511-518, 2013.

[12] HikakinTV.「ヒカキンのカップラーメン『みそきん』が 5/9 発売！ 初のブランド『HIKAKIN PREMIUM』立ち上げました。」2023-04-27. https://www.youtube.com/watch?v=SyTY7-ZTens, (参照：2023-04-28)

[13] 料理研究家リュウジのバズレシピ. 「ただの『肉入り味噌汁』じゃない、本当に旨い『豚汁』の作り方【至高の豚汁】『Pork miso soup』」 2020-06-14. https://www.youtube.com/watch?v=OL8o03u8l2Y, (参照：2024-01-19)