

## 自然言語処理を利用した類似障害情報の抽出と活用方法の提案

### Extraction of similar failure information using NLP and proposal of utilization method.

研究員：上田 良太 (日本ユニシス株式会社)  
栗原 崇至 (エプソンアヴァシス株式会社)  
杉本 智 (NTT コミュニケーションズ株式会社)  
主査：石川 冬樹 (国立情報学研究所)  
副主査：栗田 太郎 (ソニー株式会社)  
徳本 晋 (株式会社富士通研究所)

#### 研究概要

本稿では、AI 技術の発展に伴い日常業務での AI 活用を新規に検討するというユースケース下において、AI 知識、開発スキルが乏しい人間でも容易に AI を活用できないか検証を実施した。今回実施した検証では、ソフトウェアテストで収集された過去の障害事例（障害管理票）に着目して、既存 AI ツールで類似度判定を行った上位 10 事例と人が類似判定をした事例との一致率を算出し、AI ツールを活用した品質向上への可能性を評価した。

#### 1. はじめに

日常業務において自然言語で書かれた文書については数多く存在し、仕様書、設計書、テストケース、バグ管理票、課題票、障害管理票など日々の業務の中で作成される文書は大量にあるが、非定型情報のものが多く、解析または利用の着手に二の足を踏んでいるのが現状ではないかと考える。また、近年自然言語を AI 技術で分析/評価/活用したいというニーズの高まりもあるが、一見ハードルが高そうに感じるという背景もあり、まだまだ一般では簡単には実現できない状況にあるのではないかと考える。こうした背景に対し AI 技術、中でも自然言語処理を手軽に利用し、有効に活用する方法を提案する。

本稿では、2 章で課題設定、3 章で課題解決に向けたアプローチ、4 章で実施した実験と評価、5 章で考察、6 章で結論、7 章で今後の課題を記載している。

#### 2. 課題設定

##### 2.1 現状分析

筆者らの業務経験において、先に述べた日常業務で作成された文書は、プロジェクトまたはプロダクトに依存した多種多様な状況になっている。さらにこれらを統括した（プロジェクトまたはプロダクトを跨いだ）情報共有と有効活用は難しいのが現状である。例えば、PMBOK のような世界標準の知識体系に沿った記述方法（特に QCD のうち Q に関する記述など）もあるが、プロジェクト間による文章表現の統一を図ることや教育による統制までは難しく、また時間を経るごとに多彩になることはやむを得ない状況である。

##### 2.2 課題提起

今回はその中でもプロセスの最終段階で記載されている、品質にかかわる要素としての障害管理票について着目する。前出の現状分析を前提に以下を課題とした。

(1) 蓄積する過去の障害事例（障害管理票）の有効活用

多種多様な書式，特に自然言語を使用した文章記述から，類似障害を検出することで原因や対応策など参考となる情報を導き出せること。

### (2) 身近な AI 技術による解決の補助

一般に公開・普及している自然言語処理技術およびサービス（特に日本語処理）を利用することで，扱いの難しい非定型文章の活用を図る。

## 3. 課題解決に向けたアプローチ

### 3.1 自然言語処理による類似障害抽出の概略

今回は，障害管理票の中から自然言語としての記述である「タイトル」「概要」「障害詳細」などを扱い，自然言語処理技術を使用して類似障害を抽出・可視化する方法を提案する。概略のフローを図1に示す。

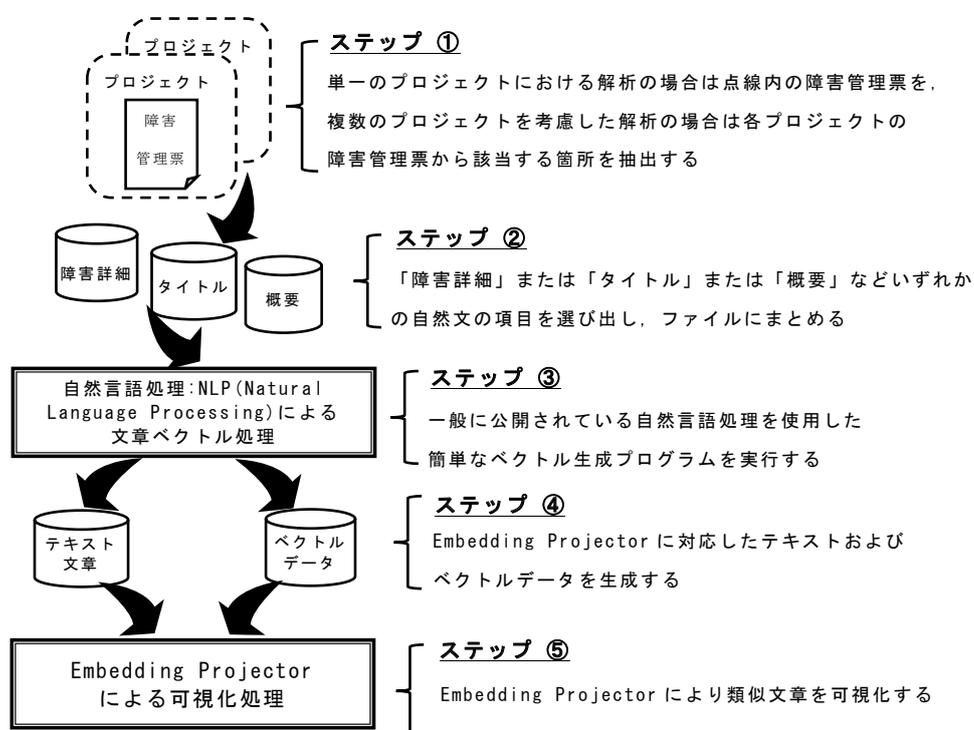


図1 自然言語処理を使用した類似障害抽出・可視化フロー

### 3.2 各ステップの説明

#### 3.2.1 ステップ①

「タイトル」「概要」「障害詳細」など自然言語で書かれた分析したい部分のみを各障害管理票より抽出する。

#### 3.2.2 ステップ②

ステップ①で，カテゴリ単位で抽出したものを，1つのテキストファイルとしてまとめる。

実際の方法としては Excel などを使用し，CSV 形式として作成と保存を行う。

#### 3.2.3 ステップ③

ここでは，TEXT (CSV) 化したデータを使い，AI ツールとして作成した簡易なプログラムにより，以降の解析に必要な文書ベクトルを生成する。作成したプログラムのソースコードは付録1を参照。

プログラムでは日本語自然言語処理のオープンソースライブラリ「GiNZA」を使用している。GiNZAについて以下に簡単に記載する。

「GiNZA」

株式会社リクルート・Megagon Labs

Version 4.x

ベクトル次元数 300

訓練コーパス UD Japanese BCCWJ v2.6 および GSK2014-A (2019) BCCWJ 版

参考：<https://megagonlabs.github.io/ginza/>

なお、今回使用した自然言語処理については、今回の目的の一つでもある、一般に広く容易に使えるように、特別な処理を追加することなく以下条件で実施している。

- (1) 学習済モデル (Pre-Training Model) を使用する。
- (2) 追加学習は行わない。
- (3) 辞書は使用しない。
- (4) ストップワードはデフォルトで使用する。
- (5) ファインチューニングは行わない。

3. 2. 4 ステップ④

プログラムは以下の2つのスタイルのファイルを生成する。

- (1) 各文章から生成された文章ベクトルファイル
- (2) 解析に使用する文章テキストファイル

2つのファイルは、ステップ⑤で使用可能なファイル形式 (TSV) として出力する。

3. 2. 5 ステップ⑤

生成されたベクトルファイルおよびテキストファイルを、Embedding Projector を使用して可視化する。Embedding Projector の使用手順については、付録2を参照。

Embedding Projector 上で可視化されたオブジェクト (文章) のいずれかを選択すると、その文章と cosine 距離が近い順に文章が抽出される。この cosine 距離が近いものを類似度が高いと判定し、類似障害を抽出する (図2)。



図2 Embedding Projector による類似する文章の表示例

Embedding Projector はオープンソースソフトウェアで、Google による機械学習研究

用として開発された「埋め込み層 (Embedding)」の視覚化と分析をインタラクティブに行うための Web アプリケーションで、多次元データをビジュアル化できる。また、2D/3D モードでデータを閲覧でき、それらを拡大、回転させることもできる。

主成分分析 (PCA), t-SNE (t-distributed stochastic neighbor embedding), カスタムリニアプロジェクションが用意されており、次元圧縮やクラスタを見るうえで汎用的で便利なツールである。

#### 4. 評価

AI ツールで抽出した上位 10 事例と被験者が目視で似ていると判定した事例 (1 以上, 最大 10 事例) の一致率を比較した。また被験者に対するアンケートにより, 目視で抽出に要した時間, および, AI ツールが算出した結果に対する納得度を回答してもらい有効性を検証した。なお, アンケート結果については付録 3 を参照。

##### 4. 1. 評価ポイント

AI ツールと被験者による抽出結果との一致率を比較するとともに, 以下のような観点にも着目して評価を行なった。

- 1 AI ツールで算出した類似度に閾値等で機械的な判断が可能か
- 2 記述フォーマット (記述内容, 文字数等) による一致率への影響
- 3 専門用語の有無による一致率への影響

##### 4. 2. 障害データ作成および比較元データ抽出

本実験では, 各々の業務プロジェクト (プロジェクト A~D) で使用した障害管理票から自然言語で記載されているタイトルや障害内容等の部分を結合し, 障害データを作成した。なお, データ内で障害に直接関係ない以下のような記述は削除している。

- ・ 証跡への参照情報
- ・ 障害管理システムへの参照 URL
- ・ ログやデバッグトレースなどのシステム固有の情報

上記で作成した障害データの中から比較元データを選定し, 同一プロジェクト内で類似度判定を行い, 類似障害か判定している。

また, 前項で記述した評価ポイントを検証するため, 各プロジェクトで障害データ, および, 比較元データの抽出条件にバリエーションを持たせた。

##### プロジェクト A (データ量 : 393 件)

- ・ 障害管理票の特性  
システム開発プロジェクトの結合試験の障害管理票
- ・ 障害データ  
「タイトル」 + 「障害内容」を結合 (平均 130 文字)
- ・ 比較元データの抽出条件  
被験者 (該当プロジェクト未参加) にて無作為に 5 件 (A-1~A-5) をピックアップ

##### プロジェクト B (データ量 : 464 件)

- ・ 障害管理票の特性  
アプリケーション製品検査の障害レポート
- ・ 障害データ  
「タイトル」のみ (6~94 文字)
- ・ 比較元データの抽出条件  
被験者 (該当プロジェクト未参加) にてタイトルに着目した際に実運用で課題がある

と意見があった以下の 5 件をピックアップ

- B-1) 特徴的な用語が含まれないもの  
例: 「マウスを画面に載せた際にカーソルが移動する」 など
- B-2) 長文 (94 文字)
- B-3) 短文 (6 文字)
- B-4) 障害概要が判断できないもの  
例: 「文言が表示されない」 など
- B-5) 専門用語で書かれたもの

#### プロジェクト C (データ量: 132 件)

- ・ 障害管理票の特性  
システム開発プロジェクトのユーザー試験の障害管理票  
(記述フォーマットがある程度整っている障害管理票を使用)
- ・ 障害データ  
「タイトル」 + 「障害内容」 + 「対処方法」 を結合 (平均 213 文字)
- ・ 比較元データの抽出条件  
被験者 (該当プロジェクト参加) にて, 実際にプロジェクト期間中に調査した障害 3 件をピックアップ
  - C-1) 対処として特定の機能修正を行っている障害を抽出
  - C-2) 対処として特定の仕様書修正を行っている障害を抽出
  - C-3) 対処として DB 更新を行っている障害を抽出

#### プロジェクト D (データ量: 304 件)

- ・ 障害管理票の特性  
システム開発プロジェクトの運用開始後にユーザーから申告された障害管理票
- ・ 障害データ  
「タイトル」 + 「障害内容」 + 「対処方法」 を結合 (平均 195 文字)
- ・ 比較元データの抽出条件  
被験者 (該当プロジェクト参加) にて, 実際に対応を行った障害 3 件をピックアップ
  - D-1) 対処として頻度が高かった障害を抽出
  - D-2) 対処として一時期多発した障害を抽出
  - D-3) 対処としてイレギュラーだった障害 (専門用語で説明されている) を抽出

### 4. 3. 実験結果

3 章で提案した手順に則り, 各プロジェクトの比較元データと障害データの類似度と被験者で判定したデータを比較検証した結果を以下に示す. なお, データ保護の観点から比較元データ, 障害データについては番号で示している.

#### プロジェクト A

5 つの比較元データに対し 4 つで 70% を超える一致を確認できた. cosine 距離が 0.3 以内と判定された障害に関しては 7/9 (78%) と高確率で類似障害として一致している.

なお, アンケートの結果としては, 目視での類似障害抽出に 1 ケースあたり 15 分の時間を有しており, AI ツールの選定結果に対しては「全て納得できる」または「ほぼ納得できる」と肯定的な回答を得られた.

表1 プロジェクトAの結果

A-1			A-2			A-3			A-4			A-5		
被験者選定	Ginza算出		被験者選定	Ginza算出		被験者選定	Ginza算出		被験者選定	Ginza算出		被験者選定	Ginza算出	
類似案件番号	類似案件番号	Cosine類似度 (cosine距離)	類似案件番号	類似案件番号	Cosine類似度 (cosine距離)	類似案件番号	類似案件番号	Cosine類似度 (cosine距離)	類似案件番号	類似案件番号	Cosine類似度 (cosine距離)	類似案件番号	類似案件番号	Cosine類似度 (cosine距離)
3781	3781	0.257	3730	3603	0.386	3883	3669	0.21	3617	3617	0.13	3850	3916	0.282
3860	3864	0.288	3603	3904	0.387	3669	3883	0.231	3941	3941	0.164	3933	3933	0.295
	3768	0.307	3779	3600	0.459	3840	3823	0.407	3715	3715	0.477	3948	3850	0.299
	3915	0.42	3907	3839	0.517		3840	0.409		3782	0.477	3893	3931	0.346
	3808	0.423		3853	0.518		3972	0.421		3701	0.495	3951	3951	0.371
	3586	0.472		3730	0.55		3862	0.471		3837	0.506	3681	3789	0.383
	3860	0.473		3896	0.605		3824	0.487		3764	0.523	3702	3681	0.454
	3691	0.49		3793	0.618		3721	0.497		3806	0.53		3893	0.485
	3654	0.493		3671	0.623		3582	0.508		3756	0.551		3601	0.495
	3758	0.497		3645	0.654		3586	0.531		3661	0.554		3807	0.527
一致率	100% (2/2)		一致率	50% (2/4)		一致率	100% (3/3)		一致率	100% (3/3)		一致率	71% (5/7)	

### プロジェクトB

「タイトル」のみでは cosine 距離が 0.4 以上と判定された障害が大半を占める結果となり、他プロジェクトと比べると一致率は低い結果となった。記述フォーマットの観点から、短文 (B-3) は単語要素が少なく特徴的な文章ベクトルを抽出できなかったと考える。また、専門用語で書かれたもの (B-5) は、今回は辞書なしで検証を行なっているので抽出されていないと考える。

なお、アンケートの結果としては、目視での類似障害抽出に 1 ケースあたり 10 分の時間を有しており、AI ツールの選定結果に対しては「どちらともいえない」～「全部違っている」と否定的な回答が多く見受けられた。

表2 プロジェクトBの結果

B-1			B-2			B-3			B-4			B-5		
被験者選定	Ginza算出		被験者選定	Ginza算出		被験者選定	Ginza算出		被験者選定	Ginza算出		被験者選定	Ginza算出	
類似案件番号	類似案件番号	Cosine類似度 (cosine距離)	類似案件番号	類似案件番号	Cosine類似度 (cosine距離)	類似案件番号	類似案件番号	Cosine類似度 (cosine距離)	類似案件番号	類似案件番号	Cosine類似度 (cosine距離)	類似案件番号	類似案件番号	Cosine類似度 (cosine距離)
115	115	0.502	344	267	0.400	116	353	0.253	175	146	0.472	145	45	0.408
249	96	0.525	267	344	0.565	70	395	0.440	146	220	0.543	147	36	0.478
284	66	0.623	61	449	0.574	408	390	0.494	11	88	0.562	44	176	0.519
455	189	0.624	87	274	0.588	212	354	0.552	88	144	0.646	38	174	0.552
251	120	0.636	107	427	0.616	391	356	0.553	44	171	0.718	311	53	0.567
93	59	0.638	228	313	0.649	393	453	0.610	220	11	0.724	219	296	0.571
131	120	0.639	254	292	0.668	265	260	0.610	171	251	0.743	289	288	0.579
96	113	0.644	111	63	0.693	239	394	0.612	394	44	0.747	36	139	0.590
3	203	0.646	274	463	0.705	160	361	0.623	239	243	0.751	37	272	0.635
113	79	0.651	292	418	0.706	27	404	0.629	393	449	0.752	385	399	0.639
一致率	30% (3/10)		一致率	40% (4/10)		一致率	0% (0/10)		一致率	60% (6/10)		一致率	10% (1/10)	

### プロジェクトC

cosine 距離が 0.3 以内で類似障害としてカウントされた課題が 5/5 (100%) と高確率で一致している。これは、障害管理票の記載者が同一人物 (4/5 同一人物が記載) しており、文章構成が似ていたためと考える。

また、C-2 で被験者が抽出した件数と AI ツールで抽出した結果で一致していない理由として、類似障害を参照する記述があり、人は類似と判定できたが AI ツールでは判定できなかったと考える。

なお、アンケートの結果としては、目視での類似障害抽出に 1 ケースあたり 5 分の時間を有しており、AI ツールの選定結果に対しては「ほぼ納得できる」「どちらともいえない」と回答していた。

表3 プロジェクトCの結果

C-1			C-2			C-3		
被検者選定	Ginza算出		被検者選定	Ginza算出		被検者選定	Ginza算出	
類似案件番号	類似案件番号	Cosine類似度 (cosine距離)	類似案件番号	類似案件番号	Cosine類似度 (cosine距離)	類似案件番号	類似案件番号	Cosine類似度 (cosine距離)
102	120	0.169	111	111	0.154	98	98	0.157
120	102	0.224	23	50	0.29	35	6	0.334
12	45	0.327	38	44	0.328	74	30	0.341
15	69	0.34	50	77	0.355		61	0.372
59	22	0.422	68	93	0.371		11	0.404
	12	0.489	91	23	0.433		90	0.429
	15	0.511	101	66	0.491		28	0.475
	88	0.567	105	2	0.522		85	0.502
	37	0.602	124	91	0.565		52	0.548
	71	0.632	132	101	0.589		35	0.612
一致率	80% (4/5)		一致率	50% (5/10)		一致率	66% (2/3)	

### プロジェクトD

結果としては、全体的に AI ツールではあまり類似障害を抽出できていない。この理由として、プロジェクト A～C は試験に関する障害管理票を扱ったのに対して、プロジェクト D はシステムの運用開始後にユーザーから問合せが来た情報をまとめたものを利用しているため、不特定多数の人物による三者三様の書き方となり文章構成が統率されていなかったことが原因と考える。また、cosine 距離が 0.3 以内で類似障害としてカウントされた課題が 7/9 (78%) と高確率で一致しているが、D-1 で cosine 距離が 0.14 といった類似の誤判定は 2 件検出されている。これは、文章整形の際に特徴的な単語が除外され、残った文章が酷似していたことが原因であった。

なお、アンケートの結果としては、目視での類似障害抽出に 1 ケースあたり 15 分の時間を有しており、AI ツールの選定結果に対しては D-2 では「ほぼ納得できる」の回答だったものの、D-1、D-3 では「どちらともいえない」「全部違っている」となっており肯定的な回答が得られなかった。

表4 プロジェクトDの結果

D-1			D-2			D-3		
被検者選定	Ginza算出		被検者選定	Ginza算出		被検者選定	Ginza算出	
類似案件番号	類似案件番号	Cosine類似度 (cosine距離)	類似案件番号	類似案件番号	Cosine類似度 (cosine距離)	類似案件番号	類似案件番号	Cosine類似度 (cosine距離)
169	160	0.141	279	281	0.128	382	180	0.428
177	161	0.143	281	285	0.128		406	0.469
191	205	0.261	285	297	0.142		289	0.521
205	401	0.285	291	291	0.144		171	0.533
260	169	0.346	297	279	0.156		441	0.567
298	409	0.366	344	299	0.391		399	0.614
351	311	0.369	356	154	0.401		423	0.632
392	266	0.388	405	381	0.489		201	0.688
401	450	0.423	431	166	0.522		259	0.698
433	390	0.424	447	405	0.538		300	0.716
一致率	30% (3/10)		一致率	60% (6/10)		一致率	0% (0/10)	

## 5. 考察

実験結果から、類似度 (cosine 距離) が 0.3 以下は類似障害として一致する確率が高い (約 73% (19/24)) ことが分かるため、類似度の閾値として活用できるのではないかと考える。なお、一致しなかった 5 件中 2 件については、文章整形の際に特徴が除かれてしまっており、作業者の感性が誤検知に繋がっている。文章整形が必要な場合は明確なルール化が必要と考える。

障害内容の文章構成が揃っていれば cosine 距離が 0.3 以下で検出され類似判定されていたことから、記述フォーマットを定め文章構成を統率することが AI ツールによる判定

に大きく影響していると思われる。一方、C-2 や D-1 のように被験者で類似障害と抽出した件数と AI ツールの類似度判定に開きがあるが、こちらは記述フォーマットが統一されず文章構成が異なっているため、乖離が出ていると考える。以上より、記述フォーマット、特に文章構成を合わせる取り組みは AI 導入の前準備として非常に有効な方法と考える。文字数の観点で見た場合、文字数の少ないプロジェクト B(94 文字以下)では一致率が低く、それ以外のプロジェクト(100 文字を超える)では比較的一致率が高い傾向が見られた。これにより類似文章を抽出するためにはある程度の文章の長さが必要であると読み取れる。

一般的に言われている自然言語処理の前処理の必要性を実験で再確認することができた。B-5、D-3 では専門用語の記述があったため、AI ツールで類似判定できていないことが見受けられる。これは前処理として辞書作成等実施すれば改善されると考える。

アンケート結果から、被験者による類似障害抽出に 1 障害あたり平均 12 分を要しており、データ総量が増えればこの時間も増えると予想できる。記述フォーマット統一と合わせれば前準備で実施した文章中の不要部分削除にかかる時間も削減でき、上記に示した文章整形時のルール化にも効果が期待できる。また、抽出の際に「業務経験に基づく解釈」や「ソフトウェアの振る舞いからの推測」といった被験者のノウハウが必要ということが読み取れる。業務経験の浅い担当者の場合、抽出精度の低下、および抽出に有する時間の増加が予想されるため、AI ツールによる類似障害抽出は業務経験不足の補完や、効率面で有効であると考えられる。

## 6. 結論

今回、GiNZA と Embedding Projector といった 2 つのオープンソースの AI ツールを活用し、類似障害を検出することができたため、AI を活用した品質向上の第一歩となった。類似度 (cosine 距離) が 0.3 以下となるものについては約 73% と高確率で人が判断した類似障害と一致していたため、膨大な情報の中からこのような AI ツールを使い閾値によるフィルターをかけて情報の絞り込みを行うような使い方は有効と考える。

GiNZA を使用した今回の検証では、記述フォーマットを統一し、文章構成を寄せていくことにより AI ツールによる類似判定の精度が向上した。そのため、文章構成が統率されているようなプロジェクトにターゲットを絞ると効果的である。

また、今回は簡易なプログラムを作成して検証を行ったが、有用な結果を得ることができた。したがって高度な AI 知識、開発スキルを保有していない場合でも、AI を活用できるアプローチを提案することができたと考えられる。

## 7. 今後の課題

専門用語が出てくるような文章については辞書などによるチューニングが必要であるが、チューニング作業については AI に関する専門的な知識が必要、かつ、手間がかかると想定される。専門的な知識がなくともチューニング作業が容易にできる仕組みがあれば、さらに活用の幅が広がるが、本稿では検討できていないため今後検討が必要であると考えられる。

「6. 結論」で AI ツールを使った閾値による判定を提案しているが、最適な閾値について本稿では検証できていない。この最適な閾値については今後検証が必要と考える。

また、文章構成が統一されている障害管理票では精度が高い結果となることが確認できたと考えられる。記述フォーマットを定義する取り組みは各プロジェクトですで行われている場合もあるが、フォーマット内の文章構成を統一するような取り組みを行なっているプロジェクトは少ないと考え、例えば Bot などによりヒアリング形式で記述していくことで自然と文章構成が統一されていくような仕組みの検討は有効ではないかと考える。

## 付録1 作成したプログラムのソースコード

```
[Python]
#=====
#
# 【目的】
# GiNZA使って、複数のCSVテキスト文章から、
# 文章の長短、単語に依存しない文章ベクトルを生成し
# Embedding Projectorを使用して類似を表示させる。
# 【使用】
# トークナイザー(形態素解析)はGiNZA Model (V4.0.0)を使用。
# 【表示】
# Embedding Projector(http://projector.tensorflow.org/)
# 出力ファイル「basedoc_text.tsv」と
# GiNZAでの文章ベクトル「basedoc_vector_ginza.tsv」を使用し、
# PCA(T-SNEやUMAPなども選択する事も可)による次元圧縮を行い、
# COS類似度などを選択し3次元可視化する。
#=====
import pandas as pd
import re
import torch
import ginza
import spacy
nlp = spacy.load('ja_ginza')

#=====
# GiNZAによる形態素解析および文章ベクトル算出サブ処理
#
# 【解説】
# GiNZAによるトークナイズ(形態素解析)と併せて
# 同時に文章の長さに依存しないベクトルを生成する。
# GiNZAの次元数はデフォルトで300次元(V4.0.0)。
# <引数>
# text: 文章テキスト
# <戻り値>
# GiNZAによる300次元(V4.0.0)の文章ベクトル
#=====
def compute_vector(text):
    doc1 = nlp(text)
    tokens = []
    print("変換中====> ", doc1)
    for sent in doc1.sents:
        for token in sent:
            tokens.append(token.orth_)
    return doc1.vector # 文章の長さ(単語数(Token数)分)に依存しない 1 x 300次元のベクトルを生成

#=====
# GiNZA文章ベクトル算出メイン処理
#
# 【解説】
# 複数のテキスト文章からGiNZAで文章の長さに依存しない文章ベクトルを出力します。
# COS類似度など、以降の類似計算に単語数に依存しない方法での計算に使用可能。
# <入力ファイル>
# basedoc.csv:
#     変換したい複数の原文をCSV形式で登録したもの
# <出力ファイル>
# basedoc_text.tsv:
#     以下のベクトルファイルに対応した原文ファイル
# basedoc_vector_ginza.tsv:
#     GiNZAによる文章ベクトルデータ。1文章=300ベクトルデータで出力
#=====

# 入力ファイル・出力ファイルの作業フォルダ(適切なフォルダパスを指定のこと)
work_folder = r'C:\Users\%UserName\Desktop\PythonPG' + '\%'

basedocs_df = pd.read_csv(work_folder + 'basedoc.csv')
basedocs_df[["text"]] = basedocs_df[["text"]].astype(str) # 文字列にしておく
vectors_ginza = []
basedocs = []
```

```
for basedoc in basedocs_df["text"]:
    basedoc = re.sub('\n', " ", basedoc) # 改行文字の削除
    strip_basedoc = re.sub(r'[:-@]', "", basedoc) # 全角記号の削除
    try:
        if len(strip_basedoc) > 3: # 文字数が少なすぎると適切なベクトルが得られない可能性があるため
            vector = compute_vector(strip_basedoc)
            vectors_ginza.append(vector)
            basedocs.append(basedoc)
    except Exception as e:
        continue

# TSVファイル出力
pd.DataFrame(basedocs).to_csv(work_folder + 'basedoc_text.tsv', sep='¥t', index=False, header=None)
pd.DataFrame(vectors_ginza).to_csv(work_folder + 'basedoc_vector_ginza.tsv', sep='¥t', index=False, header=None)
```

## 付録2 Embedding Projector 使用手順

### (1) Embedding Projector の起動

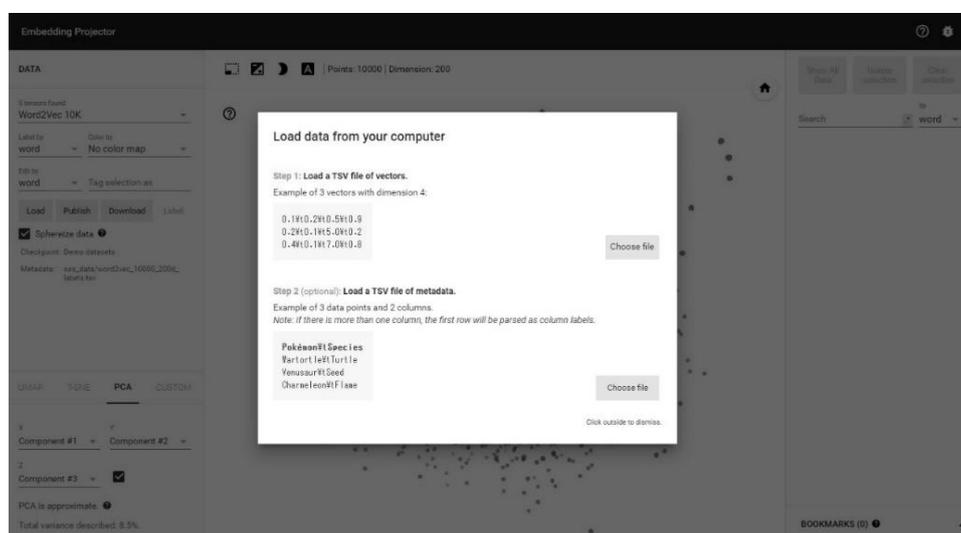
下記サイトより起動する。

<https://projector.tensorflow.org/>

### (2) ベクトルファイルおよびテキストファイルの読み込み

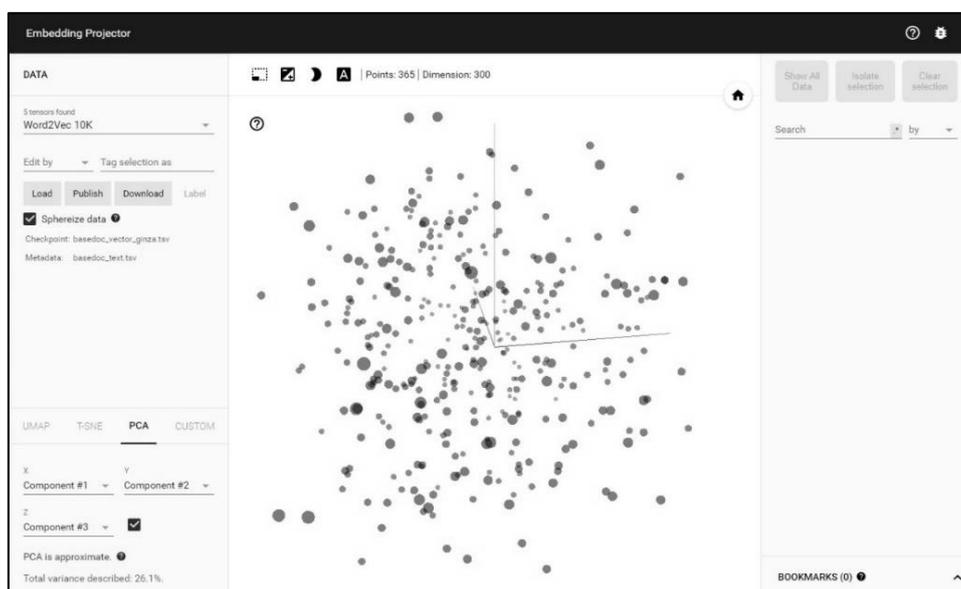
ツール左にある [Load] ボタンより「Load data from your computer」を開き、「Step1: Load a TSV file of vectors.」より「Choose File」にて出力されたベクトルファイル (TSV) を選択する。

「Step2 (optional): Load a TSV file of metadata.」 「Choose File」にて出力された文章ファイル (TSV) を選択する。



### (3) データを表示する

デフォルトで PCA とし、GiNZA では 300 次元を 3 次元に次元圧縮し、分布状態が 3D 表示される。



(4) 文章類似性を確認する

類似する文章の表示方法として、まずは起点になる文章の選択方法として大きく以下の方法がある。

- ・表示されている点を選択する。
- ・画面右の「Search」へ、キーワードなどを入力し、文章を選択する。

文章を選択すると、デフォルトで cosine (コサイン距離) により距離に近い順に右画面に上位より 100 位 (デフォルトの場合) まで表示されると同時に、3D 画面上でも 100 位までの点と文章が表示される。



### 付録3 アンケート設問・結果

自身での選定実施について回答してください(1~3)

1. 選択にどの程度の時間(分)がかかりましたか？

回答	プロジェクト/比較元データ	回答
	A-1	15分
	A-2	15分
	A-3	15分
	A-4	15分
	A-5	15分
	B-1	10分
	B-2	10分
	B-3	8分
	B-4	12分
	B-5	9分
	C-1	5分
	C-2	5分
	C-3	5分
	D-1	15分
	D-2	15分
	D-3	15分

2. 選択するときの手掛かりになったものはどんなことですか？(複数選択)

- a. 登場する単語
- b. 業務経験値からの想像
- c. その他

回答	プロジェクト/比較元データ	回答
	A-1	a, b
	A-2	a, b
	A-3	a, b
	A-4	a, b
	A-5	a, b
	B-1	a
	B-2	a, c(文の最後を見て、何についての不具合であるのかをメインに考えた)
	B-3	a
	B-4	a, b, c(登場する単語と同義語(類義語))
	B-5	a, b
	C-1	a, b
	C-2	a, b
	C-3	a, b
	D-1	a, b
	D-2	a, b
	D-3	a, b

3. 選択する際に困ったことは何ですか？（複数選択）

- a. 文に書かれている意味がわからない
- b. 業務経験による解釈ノイズ（違った解釈、捉え方など）
- c. その他

回答	プロジェクト/比較元データ	回答
	A-1	c(単語の一致だけでなく、SW/システムの振る舞いや業務の推測も必要)
	A-2	c(単語の一致だけでなく、SW/システムの振る舞いや業務の推測も必要)
	A-3	c(単語の一致だけでなく、SW/システムの振る舞いや業務の推測も必要)
	A-4	c(単語の一致だけでなく、SW/システムの振る舞いや業務の推測も必要)
	A-5	c(単語の一致だけでなく、SW/システムの振る舞いや業務の推測も必要)
	B-1	c(一通り最後の不具合までざっと見ないと、含めるべきかどうか分からない)
	B-2	c(一見似ている不具合が少ないない場合、どれを選ぶかが難しい)
	B-3	c(似た不具合が見つからない)
	B-4	a
	B-5	c(似た不具合が見つからない)
	C-1	b
	C-2	b
	C-3	b
	D-1	b
	D-2	b
	D-3	b

自身の選択結果とAIツールの結果を見て回答してください(4～6)

4. AIツールの選定は納得できる内容ですか？

- 全て納得できる
- ほぼ納得できる
- どちらともいえない
- ほぼ違っている
- 全部違っている

回答	プロジェクト/比較元データ	回答
	A-1	全て納得できる
	A-2	ほぼ納得できる
	A-3	全て納得できる
	A-4	全て納得できる
	A-5	ほぼ納得できる
	B-1	ほぼ違っている
	B-2	どちらともいえない
	B-3	全部違っている
	B-4	ほぼ納得できる
	B-5	ほぼ違っている
	C-1	ほぼ納得できる
	C-2	どちらともいえない
	C-3	ほぼ納得できる
	D-1	どちらともいえない
	D-2	ほぼ納得できる
	D-3	全部違っている

5. 設問4で「全て納得できる」「ほぼ納得できる」の場合、どのあたりがそう思いましたか？

回答	プロジェクト/比較元データ	回答
A-1		選択したものがAIでもすべて上位に選定できている。
A-2		選択したものがAIでもおおむね上位に選定できている。
A-3		選択したものがAIでもすべて上位に選定できている。
A-4		選択したものがAIでもすべて上位に選定できている。
A-5		選択したものがAIでもおおむね上位に選定できている。
B-1		
B-2		
B-3		
B-4		タイトル文字列がそもそも短く意味が不明な文字列の為、どの項目が上位であっても正しさを判定できない。
B-5		
C-1		ほぼAIで選定されており、選定されていなかった障害は人の経験から判断したものであったため、AIによる選定は難しいと思うため。
C-2		
C-3		ほぼAIで選定されており、選定されていなかった障害は人の経験から判断したものであったため、AIによる選定は難しいと思うため。
D-1		
D-2		一致率は6割程度であるものの、頻発した当初の案件をAIでもれなく抽出できていたため。
D-3		

6. 設問4で「全部違っている」「ほぼ違っている」の場合、どのあたりがそう思いましたか？

またその相違点はどのようなことが背景にあると思いますか？

回答	プロジェクト/比較元データ	回答
A-1		
A-2		
A-3		
A-4		
A-5		
B-1		人間側は、「コマンドが「何度も」送られることを重視して選択したが、AIのほうはそうではなく、コマンドの「内容が」違うという内容でも選択対象としている。同じ「コマンド」という単語により選択対象となったものと推測される。
B-2		
B-3		1個も同じ選択項目が無い為、「文字列」というより「異なる」という観点で選択されていると推測される。
B-4		
B-5		人間は、○○○○には「△△△△なんか」が複数存在し、それらが似たような場所に表示されるため、それらも含めて項目を選択したが、AIはそれらは選択対象としていないように見える。AIは「△△△△なんか」が複数存在することは知らないため、他の単語（「表示」など）の類似度から判断したと推測される。
C-1		
C-2		
C-3		
D-1		
D-2		
D-3		イレギュラーな障害であり専門用語なども含まれていたため抽出できなかったと考える。