applied sciences

MDPI

Article

Identification of Synonyms Using Definition Similarities in Japanese Medical Device Adverse Event Terminology

Ayako Yagahara [1,2,*], Masahito Uesugi [3] and Hideto Yokoi [4]

[1] Faculty of Health Sciences, Hokkaido University of Science, Hokkaido 006-8585, Japan
[2] Faculty of Health Sciences, Hokkaido University, Hokkaido 060-0812, Japan
[3] Faculty of Medical Informatics, Hokkaido Information University, Hokkaido 069-8585, Japan; uesugi@do-johodai.ac.jp
[4] Department of Medical Informatics, Kagawa University Hospital, Kagawa 761-0793, Japan; yokoi@med.kagawa-u.ac.jp
* Correspondence: yagahara-a@hus.ac.jp; Tel.: +81-11-676-8504

Abstract: Japanese medical device adverse events terminology, published by the Japan Federation of Medical Devices Associations (JFMDA terminology), contains entries for 89 terminology items, with each of the terminology entries created independently. It is necessary to establish and verify the consistency of these terminology entries and map them efficiently and accurately. Therefore, developing an automatic synonym detection tool is an important concern. Such tools for edit distances and distributed representations have achieved good performance in previous studies. The purpose of this study was to identify synonyms in JFMDA terminology and evaluate the accuracy using these algorithms. A total of 125 definition sentence pairs were created from the terminology as baselines. Edit distances (Levenshtein and Jaro–Winkler distance) and distributed representations (Word2vec, fastText, and Doc2vec) were employed for calculating similarities. Receiver operating characteristic analysis was carried out to evaluate the accuracy of synonym detection. A comparison of the accuracies of the algorithms showed that the Jaro–Winkler distance had the highest sensitivity, Doc2vec with DM had the highest specificity, and the Levenshtein distance had the highest value in area under the curve. Edit distances and Doc2vec makes it possible to obtain high accuracy in predicting synonyms in JFMDA terminology.

Keywords: terminology; synonym detection; medical device; edit distance; distributed representation; machine learning

check for updates

Citation: Yagahara, A.; Uesugi, M.; Yokoi, H. Identification of Synonyms Using Definition Similarities in Japanese Medical Device Adverse Event Terminology. *Appl. Sci.* **2021**, *11*, 3659. https://doi.org/10.3390/app11083659

Academic Editors: Andrea Prati and Akemi Galvez Tomida

Received: 27 February 2021
Accepted: 13 April 2021
Published: 19 April 2021

1. Introduction

In Japan, medical facilities and medical device manufacturers are required to submit medical device adverse event reports (MDAERs) to the Ministry of Health, Labor, and Welfare when medical devices are involved in adverse events such as catheter breakage occurring during a medical procedure. In MDAERs, a detailed description of the adverse events and the problems they give rise to for patients due to the medical devices are obtained through free writing (there are no guidelines for the written format); and it is difficult to categorize the cases involved for a statistical analysis of the adverse events.

Therefore, to standardize the terms in MDEARs, medical device adverse event terminology 1st edition (terminology of the Japan Federation of Medical Devices Associations (JFMDA terminology)) was published in March 2015 [1]. This terminology consists of 89 medical device terminology items developed by 13 industry groups in the industry that are members of the JFMDA. Each terminology item most commonly has three parts: medical device problems, patient problems, and components involved. In addition, each term of the "medical device problem" and "patient problem" categories have definition sentences, synonyms, and the term code of the Center for Devices and Radiological Health (CDRH) terminology (FDA code) [2], as shown in Figure 1.

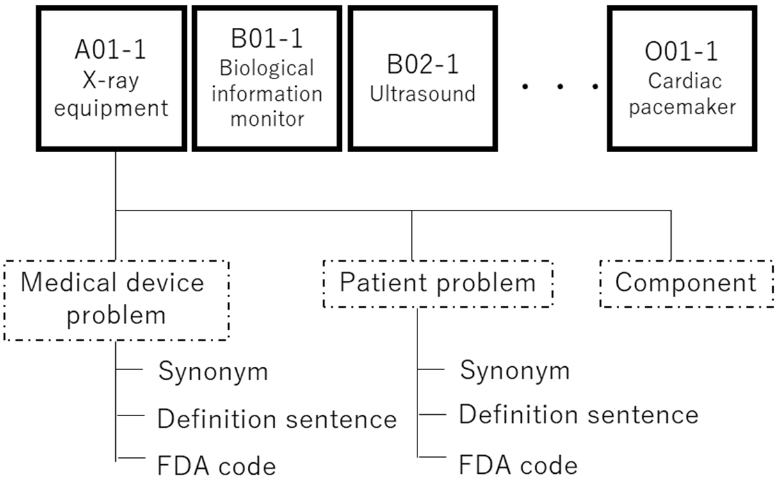

Figure 1. Overview of JFMDA terminology, 1st edition.

The 13 industry groups independently created each of the terminology entries using a bottom-up approach by gathering the terms used regularly in medical facilities to facilitate communication between medical staff and medical device manufacturers. We are working on the mapping of these terminologies to ensure that they are consistent. In a previous study, we integrated the 89 terminology items using the resource description framework (RDF), based on the spelling of the terms [3]. One problem associated with mapping terminologies is that the cases where terms are considered to represent the same concept are included with different spellings (synonyms). There are about 3500 terms related to medical device problems, and manually conducting the verification/qualification requires great effort.

We focused on the definition sentences to detect synonym pairs with different spelling. There are two main approaches to detect similar sentences: edit distance and distributed representation. The edit distance is an algorithm for quantifying how two dissimilar strings are related by counting the minimum number of operations required to transform one string into the other. This algorithm was used to map terms between the International Classification of Diseases (ICD) code and clinical text [4] and to detect misspellings in the text [5]. The advantages of this algorithm are misspelling detection and normalization of clinical terms. Hence, there is a possibility that morphological and typographical variations in the definition sentences in JFMDA terminology can be identified effectively.

In distributed representation, named Word2vec [6] and Doc2vec [7], compute vectors of words and documents using simple neural networks with context information. The similarities between words and documents are calculated via the cosine similarity. The merit of distributed representation is embedding the concept of words as vectors, and this algorithm can detect synonyms with different spellings. There are some previous medical studies for synonym identification [8–10]. Some studies used Doc2vec, which is a method for conversion from documents to vectors in order for them to be applied in the detection of similar text [11–13] and allows mapping among the standard codes [14]. We believe that distributed representations can be applied to identify synonyms efficiently, and it is necessary to evaluate the applicability to our task. The purpose of our study is to detect synonyms automatically for JFMDA terminology items and compare the accuracy of the detection among distributed representations and edit distances.

2. Materials and Methods

2.1. The Flow of this Study

The flow of our study was as follows: definition pair creation, automatic synonym detection using distributed representation and edit distance, accuracy evaluation using receiver operating characteristic (ROC) analysis (Figure 2).

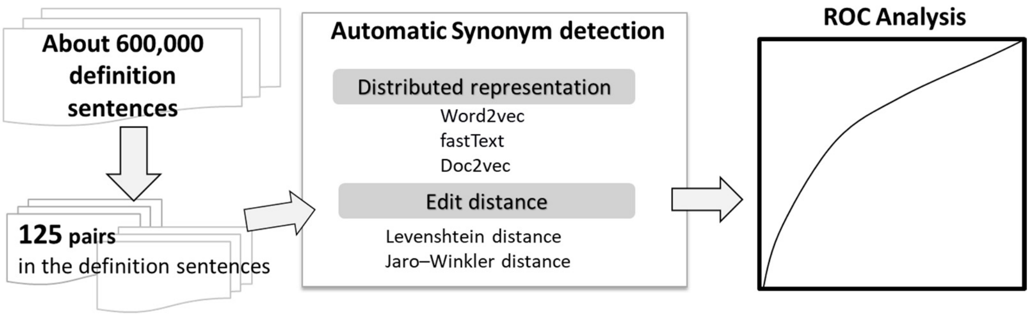
About 600,000 definition sentences
125 pairs in the definition sentences
Automatic Synonym detection
Distributed representation
Word2vec
fastText
Doc2vec
Edit distance
Levenshtein distance
Jaro–Winkler distance
ROC Analysis

Figure 2. Study flow.

2.2. Dataset Creation

Definition sentences of the terms in the medical device problem descriptions were the focus of this study. First, we extracted the terms and the sentences with their definition. We identified approximately 600,000 definition sentence pairs, and 125 pairs were extracted from these definition sentence pairs. To provide a baseline, for the 125 pairs, 50 similar definition pairs and 75 other definitions were identified by three experts in medical device safety.

2.3. Similarity Detection

In this study, we employed the Levenshtein distance [15] and the Jaro–Winkler distance [16] as the edit distance, and Word2vec [6], fastText [17,18], and Doc2vec [7] to generate distributed representations.

2.3.1. Edit Distance

The Levenshtein distance counts the frequency of editing operations (insert, substitute, and delete) that converts one string to the other string. The Jaro–Winkler distance accounts for the lengths of two strings and partially accounts for the type of typographical errors humans make when typing texts. The Jaro–Winkler distance (d_w) is calculated as:

$$d_w = sim(s_1, s_2) + lp\{1 - sim(s_1, s_2)\} \tag{1}$$

where *sim* is the Jaro Similarity for strings, s_i, l is the length of a maximum 4 characters long common prefix, and p is a constant scaling factor for how much the score is adjusted upwards for having common prefixes. The *sim* is calculated as:

$$sim(s_1, s_2) = \begin{cases} 0 & if\ m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) & otherwise \end{cases} \tag{2}$$

where $|s_i|$ is the length of s_i, m is the number of matching characters, and t is half of the number of transpositions.

2.3.2. Represented Distribution

Word2vec is used to group vectors of similar words together into a vector space to determine their similarities using a neural network. There are two architectures to produce a distributed representation of words: CBOW and skip-gram. Figure 3 shows the architectures of CBOW and skip-gram.

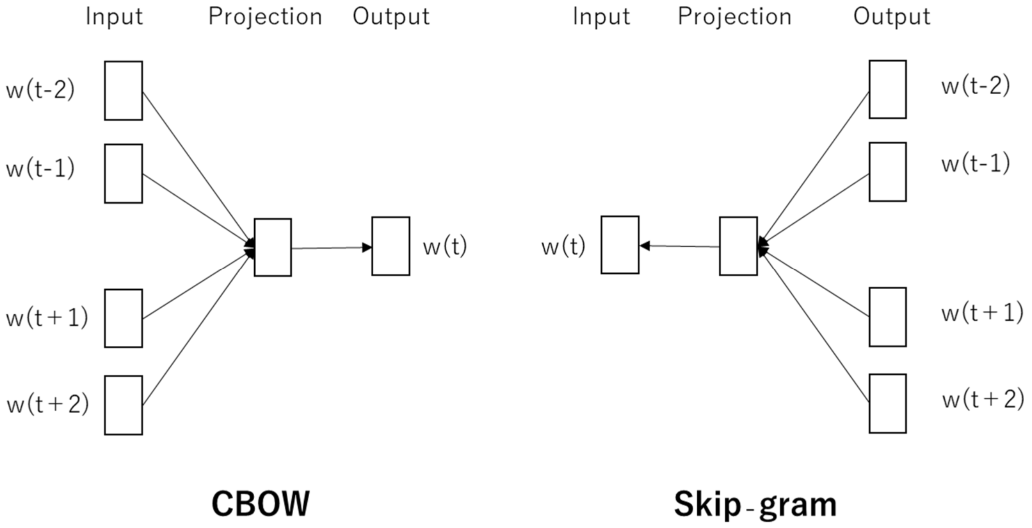

Figure 3. Architecture of CBOW and skip-gram [6].

The CBOW is a method for predicting the current word from surrounding context words. The objective function of the CBOW model is:

$$J = \frac{1}{V}\sum_{i=1}^{V} logp(w_i|w_{i-n}, \cdots, w_{i-1}, w_{i+1}, \cdots, w_{i+n}) \tag{3}$$

where V is the size of the vocabulary item, n is the window size, and $w(t)$ denotes words.

Skip-gram is a method for learning the sequence of neighboring words based on a word and its pattern of appearances. The objective function of the skip-gram model is:

$$J = \frac{1}{V}\sum_{i=1}^{V}\sum_{-n\ll j\ll n, j\neq 0} logp\left(w_{i+j}|w_j\right) \tag{4}$$

With the fastText algorithm, it is possible to take character level information into account in order to capture the meaning for suffixes/prefixes expanding Word2vec [18]. This algorithm assesses each word as a bag of character n-grams (Figure 4). There are several advantages of fastText: high training speed, applicability to large-scale corpora, and the efficiency for low-frequency words and words outside the vocabulary [19].

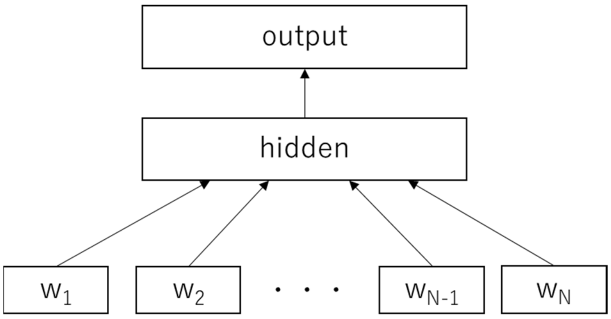

Figure 4. Architecture of fastText; w_t denote "N"-gram features.

The Doc2vec creates a numeric representation of a document of a certain length in order for it to be able to situate similar documents close to one another, much like Word2vec does. This method extends Word2vec by inputting additional parameters that are treated as an additional context vector. The learning algorithms are a distributed memory (DM) and a distributed bag of words (DBOW). Each paragraph or post message is mapped to a unique vector using the word as the target in a DM and context words as the target in the DBOW [18]. Figure 5 shows the architecture of DM and DBOW.

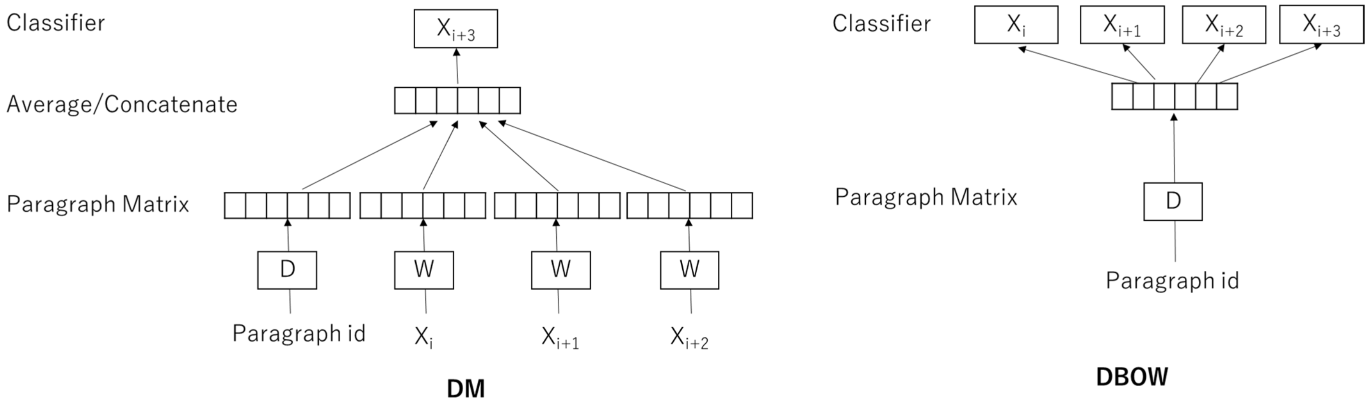

Figure 5. Architecture of Doc2vec; X_t denotes words in a sentence.

Wikipedia in Japanese (downloaded on 29 June 2018) was used as the learning data to create a distributed representation. In a Japanese document, there are no spaces between words, and spaces were inserted in the text from Wikipedia based on Japanese grammar and dictionary entries using MeCab [20] and mecab-ipadic-NEologd [21], with verb outputs as bare infinitives. Then, the Gensim package [22] in Word2vec and the library of fastText [23] were used to create trained vectors. In the parameters of each of these algorithms, the number of dimensions of the vectors was set to 300, the number of epochs to 5, and the size of the context window to 5; loss function was hierarchical softmax and the minimum number of word occurrences was set as 1. In determining these parameters, the optimal parameters in Word2vec and fastText were explored as follows: vectors from 200 to 1000, epochs 5 and 10, context windows from 5 to 20, and loss functions were softmax (only fastText), hierarchical softmax, and negative sampling. Other parameters were set to default. In Doc2vec with DM and DBOW, pre-trained word vectors were downloaded from [24]. All experiments for the training models were run on a computer with the Ubuntu 18.04 operating system, Intel Core i7-9700K, and 64 GB RAM, with the Programming language Python 3.8.3.

2.4. Similarity Calculations

In the edit distance, the similarity index is the distance between two definition sentences without symbols using the python-Levenshtein module (version 0.12.0) [25]. In Word2vec, fastText, and Doc2vec, cosine similarity was also introduced. The average vector values were calculated using vectors allocated to each word in definition sentences with symbols deleted and verbs changed to dictionary forms. In addition, sentence vectors were inferred using the genism package in Doc2vec. The cosine similarities of the pairs were calculated as follows:

$$cosine\ similarity = \frac{A \cdot B}{\|A\| \|B\|} \tag{5}$$

where A is an average vector of an input definition sentence and B is an average vector of the other definition sentence.

2.5. Evaluation

Receiver operating characteristic (ROC) analysis was carried out to evaluate the extraction accuracy of similar definition sentences, and the area under the curve (AUC) was calculated. In addition, the cutoff value was identified from the ROC curve using the Youden Index. Sensitivity and specificity were also calculated based on the cutoff value. The ROC analysis was conducted using JMP 13.2.1.

3. Results

The number of characters per definition sentence was 20.1 characters, and the difference in the number of characters between the definition sentence pairs was 7.1 characters. The number of words per definition sentence was 11.8 words, and the difference in the number of words was 4.3 words. The sensitivities, specificities, and AUC were obtained from the ROC analysis (Table 1). In the sensitivity, the values in all algorithms tend to be low. The value of the Jaro–Winkler Distance was 0.780, and this value was the highest among the sensitivities. In addition, only the sensitivity of the Jaro–Winkler Distance was higher than the specificity. In specificity, the highest was 0.880 in Doc2vec with DM. The second-best methods were the Levenshtein distance, fastText with skip-gram, and Doc2vec with DBOW. In AUC, the AUC values in the edit distance algorithms tended to be better compared to those in the distributed representation. In particular, the AUC in the Levenshtein distance was the highest. The distributed representation that has the highest AUC was Doc2vec with DBOW. Table 2 shows examples of our results.

Table 1. Accuracy of the algorithms. Underlined numbers are the highest value of each item.

Algorithm	Sensitivity	Specificity	AUC
Levenshtein Distance	0.680	0.840	**0.814**
Jaro–Winkler Distance	**0.780**	0.680	0.737
Word2vec with CBOW	0.600	0.733	0.707
Word2vec with skip-gram	0.560	0.787	0.723
fastText with CBOW	0.600	0.787	0.744
fastText with skip-gram	0.560	0.840	0.745
Doc2vec with DBOW	0.640	0.853	0.768
Doc2vec with DM	0.480	**0.880**	0.681

Table 2. Examples of similar and non-similar definition pairs (English terms in parentheses). The bold, italic underlined numbers were determined to be similar.

Term 1	Term 2	Definition 1	Definition 2	Baseline	LD	JWD	W2V with CBOW	W2V with skip-gram	FT with CBOW	FT with skip-gram	D2V with DBOW	D2V with DMPV
火事 (Fire)	機器不具合 (Device failure)	機器から煙がでること (Smoke is emitted by a device)	装置から煙がでること。 (Smoke is emitted by a machine)	***Similar***	**2**	***0.867***	***0.977***	***0.984***	***0.950***	***0.950***	***0.908***	***0.833***
機器材料の不良 (Poor of device material)	反応容器の破損 (A fracture of the reaction container)	全体又は一部の欠け (A fragment in whole or part of the components)	反応容器または一部の欠け (A fragment in a part of the components or the reaction container)	***Similar***	***6***	***0.722***	0.775	0.884	0.531	0.675	0.786	0.591
誤穿刺 (Incorrect puncture)	使用 (Use)	間違った部位を穿刺すること (Puncturing the wrong part)	本来の穿刺部位でないところを誤って刺してしまうこと (Accidentally puncturing at a point that was not the target (point))	***Similar***	19	0.463	0.896	0.942	0.751	0.871	***0.867***	0.417
圧力不良 (Insufficient pressure)	動作不良 (Malfunction)	意図した加圧動作をしないこと (The device does not perform the intended pressurization)	意図した作動をしないこと (The device does not work as intended)	Dissimilar	***3***	***0.955***	***0.965***	***0.965***	0.739	0.899	0.847	***0.857***

Table 2. *Cont.*

Term 1	Term 2	Definition 1	Definition 2	Baseline	LD	JWD	W2V with CBOW	W2V with skip-gram	FT with CBOW	FT with skip-gram	D2V with DBOW	D2V with DMPV
機器材料の不良 (Failure of the material)	損傷 (Damage)	機器の破れ (A device split)	穴が開いた状態 (A state with an open hole)	Dissimilar	7	0.000	0.496	0.750	0.006	0.417	0.718	0.506
故障 (Defect)	機械的不良 (Mechanical failure)	構成部品の全体又は 一部が外れて機械的に分離すること (All or part of the components come off the machine mechanically and separately)	構成部品の全体又は一部を損ない傷ついたりすること (All or part of the components come off a machine and are damaged)	Dissimilar	10	***1.000***	0.914	0.957	0.668	0.892	0.849	0.763

The thresholds were as follows: Levenshtein Distance (LD), 7; Jaro–Winkler distance (JWD), 0.683; Word2vec (W2V) with CBOW, 0.916; W2V with skip-gram, 0.959; fastText (FT) with CBOW, 0.776; FT with skip-gram, 0.905; Doc2vec (D2V) with DBOW, 0.860; and D2V with DM, 0.804. Underlined parts were evaluated to be similar based on the thresholds. In the LD, the pair was determined to be similar if the distance of the pair was equal or less than the threshold. The others were opposite.

4. Discussion

This study clearly shows that the sensitivity and AUC in the editing distances were better than those in the distributed representation. In particular, the Jaro–Winkler distance had the best sensitivity value and the Levenshtein distance had the best AUC value. In term of specificity, the value in Doc2vec with DM was the best.

As one of the features of the baseline with the models, the pairs with similar definition sentences tended to have the smaller differences in the number of characters; the differences in the number of characters with the pairs by different definition sentences had a wider variable range (Figure 6). In the Levenshtein Distance, the threshold value by ROC analysis was 7, and it is considered that the specificity was higher because the number of similar pairs with the value of 7 or more was small.

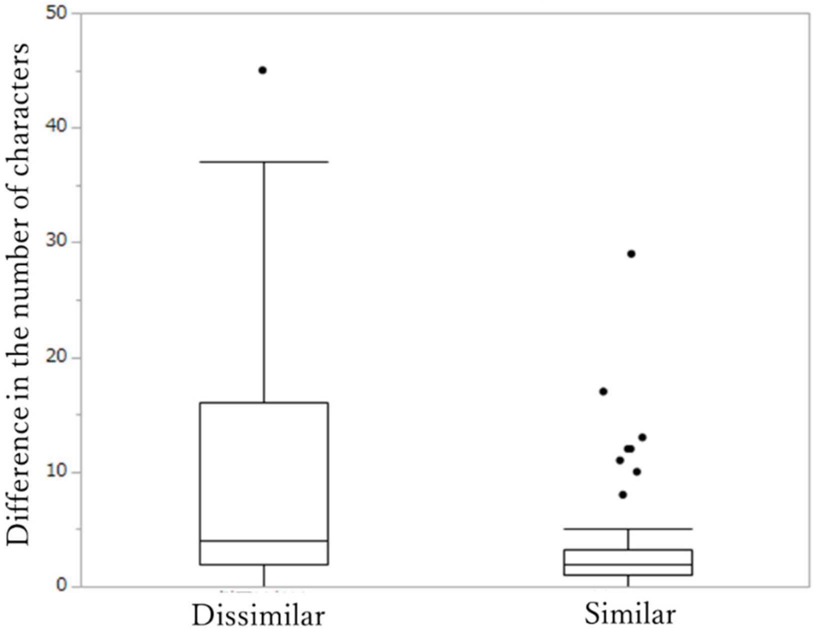

Figure 6. Differences in the number of characters between definition pairs.

The Jaro–Winkler distance had the best sensitivity. As a feature of the description of a definition sentence of the JFMDA terminology, there is a common phrase that occurs: “A fragment in a part of the components (一部の欠け)”, “A fragment in whole or part of the components (全体又は一部の欠け)", or "A fragment in a part of the components or the reaction container (反応容器または一部の欠け)". Further, common phrases are sometimes found at the beginning and end of sentences, such as "Dirt, foreign matter, etc., adhere to the entire or to parts of the surface of a component (構成部品の全体又は一部の表面に汚物、異物等が付着すること)" and "Dirt, foreign matter, stain, etc., adhere to the surface or inside of a component (構成部品の表面や内部に汚物、異物、汚れ等が付着すること". This definition pair was evaluated as dissimilar in Levenshtein distance but similar in Jaro–Winkler distance for all differences in the number of characters between the definition sentences. In addition, in the definition sentences in the terminology, there were many sentences that included common phrases such as "A device does not perform the intended XX". The value of *sim* in the Jaro–Winkler distance formula tends to increase as it contains a common phrase. As a result, it is considered that the Jaro–Winkler distance is the algorithm that detects similar pairs, including common phrases, better than the Levenshtein distance, and the sensitivity in Jaro–Winkler distance become the highest among all models. However, although the words or phrases that fill out "XX" are different in their meanings, it is a drawback that the pairs with common phrases may be determined as synonymous from the calculation characteristics.

The represented distribution tended to have low sensitivity and high specificity. Cosine similarity values of the similar pairs tended to be narrower and better than the values of the dissimilar pairs in all models. Because the range of the cosine similarity in similar pairs overlapped in a part of the higher range of cosine similarities of the dissimilar pairs shown in Figure 7, we considered that the sensitivity became poorer than the specificity. In addition to the above, the fact that the number of dissimilar pairs is larger than that of similar pairs may contribute to the improvement in specificity.

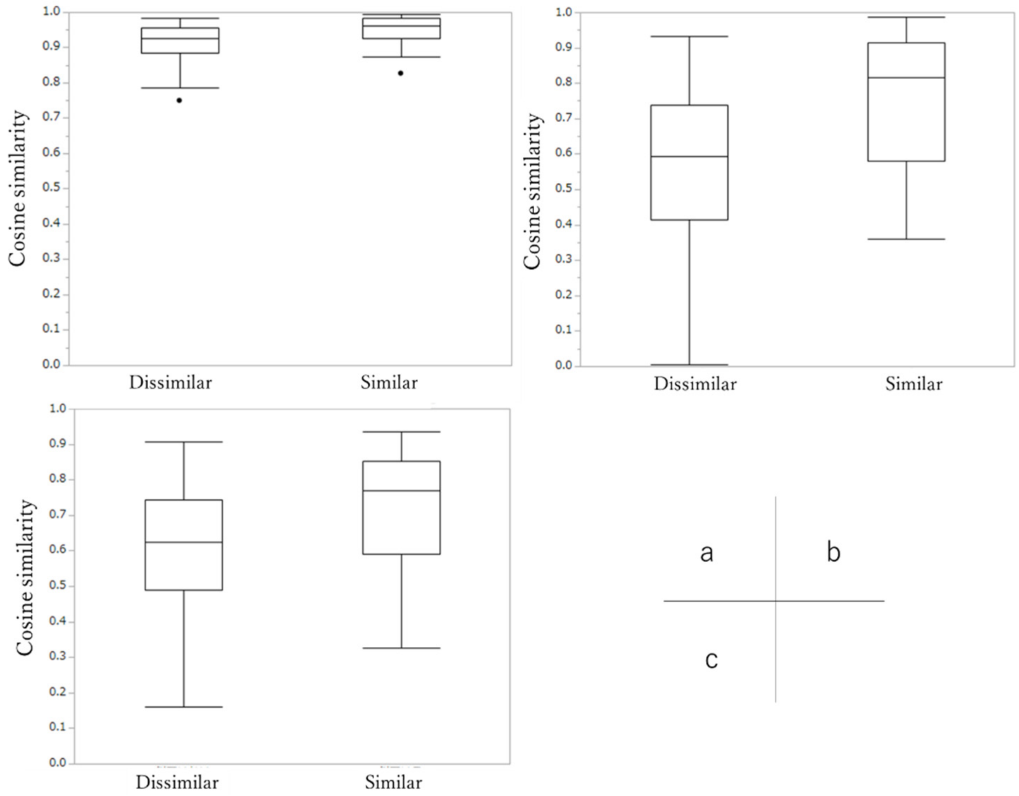

Figure 7. Examples of the range of cosine similarities in similar and dissimilar pairs in each algorithm: a, Word2vec with skip-gram; b, fastText with CBOW; c, Doc2vec with DM.

Among the definition sentence pairs that were similar in the baseline, there were 10 definition sentence pairs that were not determined to be similar in editing distances. Among 10 pairs, 2 pairs were judged to be similar by only Doc2vec with DBOW. For example, it could determine the pair, “Puncturing the wrong part (間違った部位を穿刺すること)” and “Accidentally puncturing at a point that was not the target (point) (本来の穿刺部位でないところを誤って刺してしまうこと)” as similar and agreed with the baseline shown in Table 2, and another pair was “Puncturing the wrong part (間違った部位を穿刺すること)” and “ Punctuation to a site that is not the intended area. (目的以外の部位への穿刺).” However, edit distances were not determined to be similar because the order of characters and words was different and there was no common phrase. Even using Word2vec and fastText, this definition sentence pair could not be determined to be synonyms. Although discussing two similar cases detected by Doc2vec with DM may not be sufficient because it was not statistically significant, we believe it is meaningful to conduct more investigations while increasing the number of pairs in the future.

5. Conclusions

This article evaluates the accuracy of synonym identification in JFMDA terminology using three distributed representation methods and two edit distance methods. We may conclude as follows: The Levenshtein distance was the most useful method for evaluating the similarity of definition sentences among the different algorithms because it acquired the highest AUC. The Jaro–Winkler distance has the potential to identify common phrases.

Author Contributions: Conceptualization, A.Y. and H.Y.; methodology, A.Y. and H.Y.; software, A.Y. and M.U.; validation, A.Y.; formal Analysis, A.Y.; investigation, A.Y.; resources, A.Y.; data curation, A.Y.; writing—original draft preparation, A.Y.; writing—review and editing, H.Y. and M.U.; visualization, A.Y.; supervision, H.Y.; project administration, H.Y.; funding acquisition, H.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the Research on Regulatory Science of Pharmaceuticals and Medical Devices from the Japan Agency for Medical Research and development (AMED), grant number 20mk0102150h0002.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data were obtained from the homepage of The Japan Federation of Medical Devices Associations at https://www.jfmda.gr.jp/activity/committee/fuguai/, accessed on 6 January 2014.

Conflicts of Interest: The authors have no conflict of interest to declare.

References

1. Pharmaceuticals and Medical Devices Agency. Publication and Utilization of Medical Device Adverse Event Terminology. Available online: https://www.pmda.go.jp/files/000204139.pdf (accessed on 21 February 2021). (In Japanese)
2. National Cancer Institute. Centers for Devices and Radio-logical Health (CDRH) Terminology Files. Available online: https://evs.nci.nih.gov/ftp1/FDA/CDRH/About.html (accessed on 21 February 2021).
3. Yagahara, A.; Tanikawa, T.; Ogasawara, K.; Yokoi, H. Integration of Japanese Medical Device Adverse Event Terminologies. *Stud. Health Technol. Inform.* **2017**, *245*, 1345. [PubMed]
4. Chen, Y.; Lu, H.; Li, L. Automatic ICD-10 coding algorithm using an improved longest common subsequence based on semantic similarity. *PLoS ONE* **2017**, *12*, e0173410. [CrossRef] [PubMed]
5. Tissot, H.; Dobson, R. Combining string and phonetic similarity matching to identify misspelt names of drugs in medical records written in Portuguese. *J. Biomed. Semant.* **2019**, *10*, 1–7. [CrossRef] [PubMed]
6. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. In Proceedings of the ICLR Workshops Track, Scottsdale, AZ, USA, 2–4 May 2013.
7. Le, Q.; Mikolov, T. Distributed representations of sentences and documents. In Proceedings of the 31st International Conference on Machine Learning, PMLR, Beijing, China, 21–26 June 2014; pp. 1188–1196.
8. Yeganova, L.; Kim, S.; Chen, Q.; Balasanov, G.; Wilbur, W.J.; Lu, Z. Better synonyms for enriching biomedical search. *J. Am. Med. Inform. Assoc.* **2020**, *27*, 1894–1902. [CrossRef] [PubMed]
9. Wang, Y.; Liu, S.; Afzal, N.; Rastegar-Mojarad, M.; Wang, L.; Shen, F.; Kingsbury, P.; Liu, H. A comparison of word embeddings for the biomedical natural language processing. *J. Biomed. Inform.* **2018**, *87*, 12–20. [CrossRef] [PubMed]
10. Jagannatha, A.N.; Chen, J.; Yu, H. Mining and ranking biomedical synonym candidates from Wikipedia. In Proceedings of the Sixth International Workshop on Health Text Mining and Information Analysis (Louhi), Lisbon, Portugal, 17 September 2015; pp. 142–151.

11. Zheng, T.; Gao, Y.; Wang, F.; Fan, C.; Fu, X.; Li, M.; Zhang, Y.; Zhang, S.; Ma, H. Detection of medical text semantic similarity based on convolutional neural network. *BMC Med. Inform. Decis. Mak.* **2019**, *19*, 156. [CrossRef] [PubMed]
12. Pokharel, S.; Zuccon, G.; Li, X.; Utomo, C.P.; Li, Y. Temporal tree representation for similarity computation between medical patients. *Artif. Intell. Med.* **2020**, *108*, 101900. [CrossRef] [PubMed]
13. Feng, S. The proximity of ideas: An analysis of patent text using machine learning. *PLoS ONE* **2020**, *15*, e0234880. [CrossRef] [PubMed]
14. Barretto, E.H.S.; da Costa Patrao, D.F.; Ito, M. Analysis of Usage of Term Weighting Algorithm for Mapping Health Procedures into the Unified Terminology of Supplemental Health (TUSS). *Stud. Health Technol. Inform.* **2019**, *264*, 1496–1497. [CrossRef]
15. Levenshtein, V.I. Binary codes capable of correcting deletions, insertions, and reversals. *Sov. Phys. Dokl.* **1966**, *10*, 707–710.
16. Winkler, W.E. String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage. *Proc. Sect. Surv. Res. Methods* **1990**, 354–359.
17. Joulin, A.; Grave, E.; Bojanowski, P.; Douze, M.; Jégou, H.; Mikolov, T. FastText.zip: Compressing text classification models. *arXiv* **2016**, arXiv:1612.03651.
18. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
19. Wang, Y.; Wang, J.; Lin, H.; Tang, X.; Zhang, S.; Li, L. Bidirectional long short-term memory with CRF for detecting biomedical event trigger in FastText semantic space. *BMC Bioinform.* **2018**, *9*, 507. [CrossRef]
20. MeCab: Yet Another Part-of-Speech and Morphological Analyzer. Available online: https://taku910.github.io/mecab/ (accessed on 21 February 2021).
21. MeCab-ipadic-NEologd: Neologism Dictionary for MeCab. Available online: https://github.com/neologd/mecab-ipadic-neologd (accessed on 21 February 2021).
22. Řehůřek, R.; Sojka, P. *Gensim–Python Framework for Vector Space Modelling*; NLP Centre, Faculty of Informatics, Masaryk University: Brno, Czech Republic, 2011; Volume 3.
23. FastText. Available online: https://github.com/facebookresearch/fastText (accessed on 21 February 2021).
24. Pretrained doc2vec Models on Japanese Wikipedia. Available online: https://github.com/yagays/pretrained_doc2vec_ja (accessed on 21 February 2021).
25. Python-Levenshtein 0.12.2. Available online: https://pypi.org/project/python-Levenshtein/ (accessed on 21 February 2021).