

NAIST-IS-DD1561201

## **Doctoral Dissertation**

# **Semantically Readable Distributed Representation Learning and Its Expandability Using a Word Semantic Vector Dictionary**

Ikuo Keshi

September 15, 2017

Graduate School of Information Science  
Nara Institute of Science and Technology

A Doctoral Dissertation  
submitted to Graduate School of Information Science,  
Nara Institute of Science and Technology  
in partial fulfillment of the requirements for the degree of  
Doctor of ENGINEERING

Ikuo Keshi

Thesis Committee:

Professor Satoshi Nakamura	(Supervisor)
Professor Yuji Matsumoto	(Co-supervisor)
Associate Professor Masashi Toyoda	(The University of Tokyo)
Associate Professor Yu Suzuki	(Co-supervisor)
Assistant Professor Koichiro Yoshino	(Co-supervisor)

# **Semantically Readable Distributed Representation Learning and Its Expandability Using a Word Semantic Vector Dictionary\***

Ikuo Keshi

## **Abstract**

Every human activity will be accumulated as unstructured text data with the progress in the social media and IoT (Internet of Things). Extracting value from the data is expected for distributed representation learning using neural networks. However, the problem to be solved is the sparsity of words in short text and the uninterpretable feature quantities automatically extracted by neural networks. The sparsity of words means that many of the words in short text to be analyzed are not included in the vocabulary learning distributed representations from unstructured text data.

In contrast to the distributed representations obtained by neural networks, we introduce a word semantic vector dictionary, constructed using a human expert who is a lexicographer. In the dictionary, a word represents its concepts, by relevance with or no relevance with the feature words consisting of 266 conceptual classifications. First, we propose a method that encodes knowledge in an encyclopedia text using the word semantic vector dictionary. Also, we achieve associative image retrieval to solve the problem of keywords attached to images.

Second, we propose a method to integrate feature word expansion using our dictionary with simple neural networks to solve the problem of word sparsity in Twitter. Also, we achieve reputation information extraction from Twitter. The integration showed that the accuracy of sentiment analysis was improved by learning context information even if words are sparse.

---

\*Doctoral Dissertation, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD1561201, September 15, 2017.

Finally, we propose a method that gives a specific meaning to each hidden node by introducing our dictionary into the initial weights and by using shallow neural networks. We determined the readability in a user test. A total of 52.4% of the top five weighted hidden nodes were related to tweets. For the expandability of the method, we constructed a diverse sentiment analysis benchmark and improved the word semantic vector dictionary for the purpose of distributed representations. We also conducted a word similarity task using a Wikipedia corpus to test the domain-independence of the method. We found the objective and subjective evaluation support each hidden node maintaining a specific meaning. Thus, our method succeeds in improving readability.

**Keywords:**

distributed representation, word semantic vector dictionary, paragraph vector, word2vec, sentiment analysis, Twitter

# 単語意味ベクトル辞書を用いた可読性のある意味表現 学習と拡張性\*

芥子 育雄

## 内容梗概

ソーシャルメディアとIoTの急速な進展により、あらゆる人間活動が非構造化テキストデータとして蓄積されようとしている。これらビッグデータから価値を取り出すことが、ニューラルネットワークを用いた分散表現学習に期待されている。課題は、Twitterなどの短文テキストは単語のスパース性があること、およびニューラルネットワークにより抽出された特徴量の理解が困難なことである。単語のスパース性とは、解析対象テキスト中の単語の多くが、非構造化テキストから分散表現を学習した語彙に含まれないことを指す。

ニューラルネットワークにより獲得される分散表現に対して、本論文では、辞書編纂の専門家が構築した単語意味ベクトル辞書を導入する。本辞書では、単語は266種類の概念分類から成る特徴単語との関係（関係あり、なし）により意味を表現する。専門家が特徴単語との関係を付与した単語を基本単語と呼ぶ。

最初に単語意味ベクトル辞書を利用して、百科事典テキストから抽出された10万語以上の単語意味ベクトルを半自動的に学習する方法を提案する。まず百科事典テキストの頻度の高い3700語を基本単語として、専門家が266種類の特徴単語との関係を付与した。次に百科事典テキストを用いたブートストラップアルゴリズムにより、10万語以上の単語意味ベクトルを学習した。この単語意味ベクトルを用いて、画像に付与されたキーワードに対して、百科事典の知識を利用した画像の連想検索を実現し、有効性を検証した。

次にTwitterにおける単語のスパース性を解消することを目的に単語意味ベクトル辞書を用いてツイート中の基本単語を特徴単語に拡張し、ニューラルネットワークで学習する統合化手法を提案する。本辞書は、専門家によって構築された

---

\*奈良先端科学技術大学院大学 情報科学研究科 博士論文, NAIST-IS-DD1561201, 2017年9月15日.

約2万語の基本単語を用いる。本統合化手法を用いて、Twitterからの評判情報抽出を実現した。提案手法は単語のスパース性を解消し、極性分析の精度が向上することを示した。

我々は更にニューラルネットワークの隠れ層ノードに特定の意味（特徴単語）を割り当て、単語意味ベクトル辞書を元に約2万語の基本単語に初期の重みを設定することにより、学習後も重みの大きな隠れ層ノードの意味が維持されることを示す。ユーザテストの結果、教師なし学習によるツイートの分散表現に対して、重みの大きな上位5ノードの52.4%がツイートと関連があることが示された。また、分散表現学習を前提に単語意味ベクトル辞書を改良し、特徴単語数を264種類とした。さらに多様性のある大規模な極性分析ベンチマークを構築し、提案手法の拡張性評価を行った。最後にWikipediaコーパスを用いて、提案手法のドメイン独立性の評価を行った。これら客観的、主観的評価により、隠れ層のノードは特定の意味を維持することを確認し、提案手法は分散表現学習の可読性を改善することを示した。

## キーワード

分散表現, 単語意味ベクトル辞書, パラグラフベクトル, word2vec, 極性分析, Twitter

# Contents

<b>Acknowledgements</b>	<b>xi</b>
<b>1. Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Research Target . . . . .	2
1.3 Thesis Scope . . . . .	4
1.3.1 Proposal 1: Associative Image Retrieval . . . . .	5
1.3.2 Proposal 2: Reputation Information Extraction from Twitter . . . . .	6
1.3.3 Proposal 3: Social Media Mining . . . . .	6
1.3.4 Resource 1: Word Semantic Vector Dictionary . . . . .	7
1.3.5 Resource 2: Evaluation Benchmark for Sentiment Analysis . . . . .	7
1.4 Rest of this thesis . . . . .	8
<b>2. Text Mining</b>	<b>11</b>
2.1 Introduction . . . . .	11
2.2 Commonsense Knowledge Bases . . . . .	11
2.2.1 Cyc . . . . .	11
2.2.2 ConceptNet . . . . .	12
2.3 Semantic Lexicon . . . . .	12
2.3.1 WordNet . . . . .	12
2.3.2 Sentiment Lexicon . . . . .	13
2.4 Distributed Representation . . . . .	14
2.4.1 Context Vector Representation . . . . .	14
2.4.2 Word2vec and Paragraph Vector Models . . . . .	14
2.5 SemEval: Shared Tasks on Sentiment Analysis in Twitter . . . . .	16
2.5.1 Datasets . . . . .	17
2.5.2 Evaluation Measure . . . . .	18
2.6 Conclusion . . . . .	19
<b>3. Associative Retrieval Using Knowledge in Encyclopedia Text</b>	<b>21</b>
3.1 Introduction . . . . .	21
3.2 Creating a Word Semantic Vector Dictionary Using an Encyclopedia Text . . . . .	22

3.2.1	Definition of Word Semantic Vector . . . . .	22
3.2.2	Selection of Feature Words and Criteria for Manual Entries . .	23
3.2.3	Bootstrapping Algorithm of Word Semantic Vectors . . . . .	24
3.3	Image Retrieval Using the Word Semantic Vector . . . . .	26
3.3.1	Outline of the Digital Photo Catalog System . . . . .	27
3.3.2	Evaluation of the Digital Photo Catalog System . . . . .	30
3.3.3	Learning Function by User . . . . .	32
3.4	Conclusion . . . . .	34
<b>4.</b>	<b>Evaluation Benchmark</b>	<b>37</b>
4.1	Introduction . . . . .	37
4.2	Test Collection for Evaluation of Japanese Information Retrieval Systems	37
4.2.1	Overview of BMIR-J2 . . . . .	38
4.2.2	Design issues and discussion . . . . .	40
4.3	Japanese Twitter Sentiment Analysis Benchmark . . . . .	41
4.3.1	Single Domain Benchmark . . . . .	41
4.3.2	Diverse and Large-scale Benchmark . . . . .	43
4.4	Conclusion . . . . .	45
<b>5.</b>	<b>Reputation Information Extraction from Twitter</b>	<b>47</b>
5.1	Introduction . . . . .	47
5.2	Related Work . . . . .	48
5.3	Proposed Method . . . . .	50
5.3.1	Reputation Information Extraction Method from Twitter (Proposed Method) . . . . .	50
5.3.2	Learning Paragraph Vectors Using Expanded Feature Words .	52
5.4	Experiment of Extracting Reputation Information from Twitter . . . .	53
5.4.1	Procedure . . . . .	53
5.4.2	Experimental Method . . . . .	55
5.4.3	Results . . . . .	57
5.4.4	Consideration . . . . .	60
5.4.5	On the effectiveness of the proposed method on a benchmark with diversity even if it is minimal . . . . .	63
5.5	Conclusion . . . . .	66



<b>6. Semantically Readable Distributed Representation Learning and Its Expandability</b>	<b>69</b>
6.1 Introduction . . . . .	69
6.2 Related Work . . . . .	70
6.3 Proposed Method . . . . .	72
6.3.1 Hypothesis . . . . .	72
6.3.2 Model Setting for Testing the hypothesis . . . . .	74
6.4 Experiments . . . . .	77
6.4.1 Learning Word Vectors by Our Method and Evaluation of Correlation Coefficients . . . . .	78
6.4.2 Evaluation of Sentiment Analysis . . . . .	80
6.4.3 User Test for Readability . . . . .	82
6.5 Expandability Evaluation . . . . .	85
6.5.1 The Improvement of the Word Semantic Vector Dictionary . .	85
6.5.2 Evaluation Using the Diverse and Large-scale Benchmark . .	87
6.5.3 Domain Independent Test . . . . .	91
6.6 Conclusion . . . . .	95
<b>7. Conclusion</b>	<b>97</b>
7.1 Summary of Contributions . . . . .	97
7.2 Future Direction . . . . .	98
<b>References</b>	<b>101</b>

## List of Figures

1	Research Target. . . . .	2
2	Problem Definition. . . . .	4
3	Time series analysis of product related tweets on Sharp Corp (3/24/2014 - 3/27). Topics surrounded by solid squares indicate excitement from user's tweets, and items surrounded by dotted squares are related to media. . . . .	8
4	Word2vec learning images. . . . .	15
5	Word2vec and paragraph vector models. . . . .	15
6	Flow of bootstrapping algorithm for word semantic vectors. . . . .	25
7	Image retrieval in semantic vector space. . . . .	27
8	An example of image retrieval (Query "Resort House"). (Photos are from JPRC Photo Disc.) . . . . .	29
9	An example of text retrieval (Query "Resort House"). . . . .	31
10	An example of image retrieval using relevance feedback. (Photos are from JPRC Photo Disc.) . . . . .	32
11	Overall flow of the proposed method. . . . .	50
12	An example of extracted core words from a tweet and expanded feature words from the core words. . . . .	52
13	Experimental procedure of the extraction of the reputation information from Twitter. . . . .	53
14	Why do we think that the meaning of the hidden layer is maintained? .	73
15	Example of retrofitting "Disease." . . . .	75
16	Skip-gram model setting for testing. . . . .	76
17	Procedures of sentiment analysis. . . . .	80
18	Example of a task by crowdsourcing. . . . .	82
19	Application image of social media filtering using the conceptual axis. .	84
20	Procedures of sentiment analysis for the expandability evaluation. . .	86

## List of Tables

1	Datasets of Subtask A from 2013 to 2017. . . . .	17
---	--	----

2	The confusion matrix. Cell XY stands for “the number of tweets that the classifier labeled X and the gold standard labels as Y.” P, N, U stand for Positive, Negative, Neutral, respectively [44] . . . . .	18
3	Classification of feature words. . . . .	23
4	Grant criteria by logical relationship. . . . .	23
5	Grant criteria by associative relationship. . . . .	24
6	Captions to describe images. . . . .	28
7	Semantic vectors for images. . . . .	28
8	Buttons on the main window of the digital photo catalog system. . . .	30
9	Examples of a caption for the representative image in each group and its queries. . . . .	33
10	Configuration of the benchmark. . . . .	42
11	Statistical information of each benchmark. . . . .	42
12	Number of labels given to tweets by crowdsourcing . . . . .	43
13	Japanese Twitter Sentiment Analysis Benchmark . . . . .	43
14	Evaluation results1: macro-averaged F score for predicting positive and negative tweets in the dev. set and the test set (the error rates). . .	57
15	Parameters of feature extraction and their values for Product A and Product B. . . . .	59
16	The number of succeeded tweets and failed tweets by the proposed method in Product A. . . . .	60
17	The number of succeeded tweets and failed tweets by the proposed method in Product B. . . . .	60
18	The typical feature words in the succeeded tweets and the failed tweets by the proposed method in Product B (their occurrence ratio to their pair group). . . . .	62
19	Configuration of the experimental benchmark. . . . .	64
20	Parameters of feature extraction and their values for the proposed method and the conventional method. . . . .	65
21	Evaluation results2: F scores(SD) in the three class and the 2-class classifications by PV-DBOW (the conventional method) and PV-DBOW (the proposed method). . . . .	65
22	Hyper-parameter settings for learning word vectors. . . . .	77

23	Example of retrofitted and learned word vectors for a core word that is a feature word itself. . . . .	77
24	Example of retrofitted and learned word vectors for a core word that is not a feature word. . . . .	78
25	Evaluation results 1: Correlation coefficients between initial and learned word vectors. . . . .	79
26	Evaluation results 2: Macro-average F-score for predicting positive and negative tweets in 3-class sentiment analysis. . . . .	81
27	Evaluation results 3: Readability of hidden nodes and macro-average F-score of the corresponding 2-class sentiment analysis. . . . .	83
28	Top feature words given to core words . . . . .	86
29	Hyper-parameter settings for learning word and paragraph vectors. . .	87
30	Statistical information on the learning of the Twitter corpus . . . . .	87
31	Evaluation results 4: Correlation coefficients between seed vectors and learned word vectors. . . . .	88
32	Evaluation results 5: Macro-average F-score for predicting positive and negative tweets in 2-class sentiment analysis. . . . .	89
33	Relationship between the correlation coefficients after feature extraction and the task accuracy of the dev. set for the PV-DM. . . . .	90
34	Relationship between the correlation coefficients after feature extraction and the task accuracy of the dev. set for the PV-DBOW. . . . .	90
35	No Bulk Learning for PV-DM. . . . .	91
36	No Bulk Learning for PV-DBOW. . . . .	91
37	Statistical information on the learning of the Wikipedia corpus . . . .	92
38	Evaluation results 6: Correlation coefficients (closed test) . . . . .	93
39	Evaluation results 7: Word similarity task using Wikipedia corpus. . .	93
40	Example of Top n weighted feature words and similar words for core words . . . . .	94
41	Word similarity test using Wikipedia corpus for the evaluation of 264 feature words and 266 feature words . . . . .	94
42	Word similarity test for the evaluation of the dictionary and distributed representation . . . . .	95

## Acknowledgements

主指導教員の中村哲教授は、私がシャープ（株）入社時の先輩でもあり、多元ビッグデータプロジェクトを立ち上げられたことを知り、2014年秋からシャープとの共同研究をお願いしました。翌年、中村教授に勧めて頂いたことに背中を押され、シャープを早期退職し、第2の人生のために博士後期課程入学を決意しました。入学後は博士号を2年で取得できるようにご指導頂いたことに心から感謝致します。

副指導教員の松本裕治教授は、共同研究の立ち上げ時に深夜でもメールに返信頂くなど、自然言語処理研究に対する情熱に刺激を受けました。自然言語処理研究会、中間発表、公聴会において本論文を精練するための数多くのコメントを頂きました。心から感謝致します。

東京大学の豊田正史准教授には、学外専門家として博士論文審査員を引き受けて頂き、有難うございました。公聴会で頂いたコメントは、追加実験により考察を行い、本論文（提案3）を改善できたと思います。心から厚く感謝致します。

副指導教員の鈴木優准教授には、データ工学の立場からアドバイスを頂き、特に入学後最初の論文（提案2）投稿時に手厚くご指導頂きました。また、本論文（提案3）のクラウドソーシングを活用した大規模なベンチマーク作成やユーザ評価において多大なるご支援を頂きました。心から感謝致します。

副指導教員の吉野幸一郎助教には、意味解析の立場からアドバイスを頂き、特に研究立ち上げ時に研究の方法論や機械学習に関してご指導頂きました。心から感謝致します。

Carnegie Mellon University の Graham Neubig 助教（前本学助教）には、本研究の課題を解決するためにベイズ最適化やレトロフィッティングなどの論文・ツールを御紹介頂き、研究の助言を頂きました。心から感謝致します。

本学情報科学研究科 Sakriani Sakti 助教、須藤克仁准教授、田中宏季助教には、研究における貴重なご助言を頂きました。心より感謝致します。

恩師の北陸先端科学技術大学院大学の溝口理一郎教授には、大阪大学大学院時代に研究の方法論や人工知能研究の面白さを教えて頂きました。今も現役の研究者としてご活躍しておられ、刺激を受けています。心から感謝しております。

シャープ株式会社 IoT クラウド事業部第1サービス開発部の上田徹氏、野田浩明氏、向井理朗氏、大原一人博士、阿部一博博士の皆様に対して、本研究のご支援および議論頂き、心から感謝しております。

シャープ株式会社の元連想検索開発グループのメンバー、特に黒武者健一氏（現在ヤフー株式会社）には画像データベース（提案1）の実装や連想検索のユー

ザインタフェース応用において多大なる貢献を頂いたことに感謝しております。

元シャープ株式会社パソコン事業部の名井哲夫氏（現在イ・ソフト株式会社副社長）、中川潤子氏には、「パソコンナビ2001-リッキーくん」を商品企画頂いたことに感謝しております。リッキーくんにより、大阪大学大学院の頃から追いかけてきた私の夢が一つ実現出来ました。リッキーくんはパソコン事業として成功したとは言えないことが心残りです。

シャープ株式会社の元上司の皆様には感謝致します。特に藤本好司博士（元龍谷大学教授）には博士号を取得することを強く意識付け頂きました。大崎幹雄先生（元奈良学園大学教授）には連想検索研究をご指導頂きました。千葉徹博士（現在レジリオ株式会社代表取締役）には、私のミッションを研究者からプロデューサーに変え、e-ライフ（ホームネットワーク、ホームヘルスケア）のプロジェクト立ち上げをご指導頂きました。千葉滋博士には、単身赴任先の幕張において楽しく仕事をさせて戴きました。八尋俊英氏（元経済産業省、現在株式会社日立コンサルティング社長）からは、様々な会社にご一緒させて頂き、刺激を受けました。

1993年2月から5年間に渡り活動を行った、情報処理学会データベースシステム研究会情報検索システム評価用データベース構築ワーキンググループの元委員の皆様には感謝致します。また、発足当時のデータベースシステム研究会主査をされていた田中克己先生（元京都大学教授）には、公募の提案や共同研究でお世話になりました。また、博士後期課程に入学後はDEIM2016発表時の座長として本論文（提案2）にコメントを頂き、関西データベースワークショップでも本研究にコメントを頂きました。心から感謝しております。

大阪大学情報科学研究科の尾上孝雄教授、本学情報科学研究科の岡田実教授には、私のミッションがプロデューサーに変わった時にホームネットワークプロジェクトなどの共同研究をお願いしました。本論文の関連パブリケーションの通り、成果を学術論文、国際会議、研究会発表に纏めており、深く感謝しております。

本学知能コミュニケーション研究室の松田真奈美秘書、多元ビッグデータプロジェクトの林美保アシスタントには、庶務の手続きや研究生活をサポート頂き、心より感謝しております。

本学知能コミュニケーション研究室の学生諸氏からは、ゼミ、輪講などにおいて、様々な刺激を受けました。特にビッグデータ班の皆様には感謝致します。

最後に、長い単身赴任から自宅に戻った私を支えてくれた妻に感謝するとともに、教育者であった亡き父と工学博士号の取得を誰よりも喜んでくれている母に心より感謝致します。

# 1. Introduction

## 1.1 Background

The rapid progress in the Internet and social media has enabled collecting and analyzing large amounts of unstructured text data. Also, every human activity will be accumulated as unstructured text data with the progress in the IoT. For example, the unstructured text data of watched TV programs, browsed electronic books, and viewed tweets will be accumulated. Discovering value from the data alone—not big but small in each domain—is limited because of the word sparsity problem in short text. However, new value could be created by analyzing them together with other domain data.

In 2013, learning the meaning of each word in a relationship with about 200 neurons using a shallow neural network showed that the inner product and the difference between learned word vectors could represent the word analogy and the direction of the word meaning [38, 39, 40]. Because the tool was published as word2vec, it has been widely used in the applications of natural language processing since 2013. Also, in 2014, a paragraph vector that extended the words to a document was proposed. It showed state-of-the-art potential at the time of the publication in various tasks of natural language processing [34]. Such word vectors or document vectors are called distributed representations, word embeddings, or document embeddings.

The problem with distributed representation learning is that understanding which dimension corresponds to which meaning becomes difficult because all dimensions are initialized randomly. When distributed representation learning is applied to big data analysis, distributed representations acquired in different domains should be used in common. Also, deep learning with distributed representations will likely be applied to value discovery from unstructured text data in different domains because it can extract the feature quantities automatically. However, even if all the data in different domains can be gathered and learned with deep learning, the results cannot be explained because the meaning of the extracted feature quantities is not known.

The definition of “readability” or “readable” included in this thesis title is that people can understand the meaning of large weighted features of distributed representations for words and unstructured text. If distributed representation learning could be achieved to the point that the meaning of features can be understood, enterprises could utilize it, and new applications could be created based on the readability.

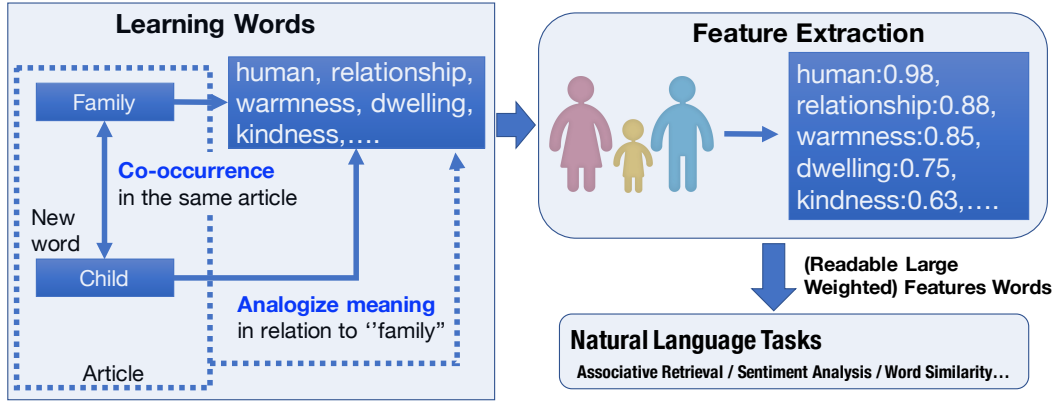


Figure 1. Research Target.

## 1.2 Research Target

This thesis describes our aim to achieve semantically readable distributed representation learning. The hypothesis was that people learn the meaning of words in association with more than 200 features. Suppose that 266 feature words are used here. For example, when people memorize the meaning of the word “family,” they learn its concept in association with features such as “human,” “relationship,” and “warmness.” When a new word “child” co-occurs with “family” in the same article, people relate the concept of “child” to the word “family” to analogize the meaning. Because people grasp the meaning of words related to 266 kinds of feature words, a large number of example texts are not necessary. The average vocabulary of Japanese people is about 20,000 words for 10-year-old children and about 50,000 words for adults. When viewing images, articles, and words about “family,” we need to be able to understand the meaning of large weighted features memorized in association with “family.” Neural networks can learn the concepts in the hidden layer, but people cannot understand the meaning of large weighted features.

The research target is shown in Figure 1. The word semantic vector, which we propose in this thesis, expresses the relationship between a core word and feature words as a binary value that is related or unrelated. The core words are for example the 20,000 words that 10-year-old children can memorize, and they mean the important words that can be analogized into new words. The feature words are, for example, 266 concepts that systematically cover conceptual classification, and many feature



words correspond to words such as “human,” “relationship,” or “warmth.” The word semantic vector is a distributed representation in which each dimension corresponds to a feature word. The word semantic vector dictionary lists feature words related to core words. The dictionary can be created using manual entries based on the logical and associative relationships between the core and feature words. Also, if the relationships are expressed binarily by a human expert, they can be produced independently of the fields, and individual differences are also small. However, because a vocabulary of  $2^{266}$  concepts can be possibly represented by the distributed representation, words with similar concepts are similar to the extent that they share the same set of feature words. Also, words of concepts to be discriminated can be separated to the extent that they have a different set of feature words [18, 52].

Logical and associative relationships of words are represented in the dictionary, while distributed representation learning like word2vec learns the context of words (surrounding words). Therefore, we expect to solve the word sparsity problem in short text by integrating both the dictionary and distributed representation learning instead of learning the context from random initial values like word2vec. In other words, even people can understand the concept of a word from a dictionary and learn how to use it from a few sentences. By adopting a mechanism that learns words like people, efficient word learning that integrates the dictionary into the distributed representation learning can be achieved.

Concerning the feature extraction from unstructured texts, unsupervised learning using neural networks can extract the feature quantities based on the learned word vectors. Applications of natural languages such as associative retrieval and sentiment analysis are achieved using these feature quantities. Large weighted feature words that are readable for each unstructured text will provide confirmation as to whether or not unsupervised learning is going well. Mining unstructured texts without given labels will become possible because unstructured texts with similar large weighted feature words are thought to be close in concept. Also, mining of data with unsupervised learning in different domains will be possible because 266 kinds of feature words are common. We will study whether or not the meaning of each dimension of distributed representation is maintained as a feature word even after unsupervised learning using neural networks.

Applications	Image Database for Designers	Reputation Information Extraction from Twitter	Social Media Mining
Problems	<b>Keyword Problem</b>	<b>Word Sparsity in Short Text</b>	<b>Uninterpretable Feature Quantities automatically extracted by Neural Networks</b>
Function by Readability	<b>Modification of Feature Words for queries</b>	<b>Error Analysis using Expanded Feature Words</b>	<b>Top 5 Weighted Feature Words are related to the Unstructured Text and Word</b>
Natural Language Tasks	Associative Retrieval	Sentiment Analysis/Word Similarity (SVM Classifier/Cosine Density)	
Feature Extraction	Indexing Images ( <b>Count base</b> )	Paragraph Vector for the Benchmark ( <b>Neural Network base</b> )	
Learning Words Seed Vectors	Bootstrapping ( <b>Count base</b> )	Paragraph Vector/Word2vec for Unstructured texts ( <b>Neural Network base</b> )	
Unstructured Texts	Binary Word Vectors	Deafault Random Initialization	<b>Feature Words: one-hot vectors</b> <b>Core Words: using Retrofitting</b>
<b>Word Semantic Vector Dictionary (Section 3.2, Section 6.5.1)</b>	Encyclopedia Texts	Unlabeled Tweets + <b>Expanded Feature Words</b>	1. Unlabeled Tweets 2. Diverse Unlabeled Tweets 3. Wikipedia Corpus
<b>Evaluation Benchmark (Chapter 4)</b>	3700 Core Words×266 Feature Words	20,330 Core Words×266 Feature Words	20,330 Core Words × 1. 266 Feature Words 2.,3. 264 Feature Words
	Small Benchmark (BMIR-J1, J2)	Single Domain Product A:4814 Product B:11,774	1. Single Domain (Product B) 2. Diverse Labeled Tweets:38K 3. Word Similarity Dataset
	<b>Proposal 1 (Chapter 3)</b>	<b>Proposal 2 (Chapter 5)</b>	<b>Proposal 3 (Chapter 6)</b>

Figure 2. Problem Definition.

### 1.3 Thesis Scope

The definition of the problem in this thesis is shown in Figure 2. The problem to be solved is the sparsity of words in short text and the uninterpretable feature quantities automatically extracted by neural networks. The sparsity of words means that many of the words in short text to be analyzed are not included in the vocabulary learning distributed representations from unstructured text data. Although many proposals such as word2vec [38, 39, 40], paragraph vector [34], GloVe [46], and FastText [5, 19] have been made since 2013 as distributed representation learning using shallow neural networks, the simplest word2vec/paragraph vector models are used as a means to achieve the research target. Also, as shown in the research target, feature extraction and natural language tasks are separated so as to confirm that large weighted feature words are readable. This doctoral dissertation consists of three proposals using a word

semantic vector dictionary and of two resources: the word semantic vector dictionary and evaluation benchmarks. The main technical differences between the three proposals in Figure 2 are as follows.

- Learning words and feature extraction is a word count base (Proposal 1) or neural network base (Proposal 2 and Proposal 3).
- In the neural network base, feature words of the word semantic vector dictionary are expanded in unstructured texts (Proposal 2) or used as seed vectors of the neural network (Proposal 3).

### **1.3.1 Proposal 1: Associative Image Retrieval**

Applying existing keyword retrieval to image retrieval causes some problems. Among them, database developers must describe photograph content in detail, and adequate retrieval is difficult.

These problems have been resolved by developing an associative retrieval technology using not keywords but word semantic vectors as a retrieval method with an associative function such as one a human being possesses [20, 21, 23, 22]. The word semantic vector expresses the relationship between a word and 266 feature words as a binary value that is related or unrelated for a human expert. The word semantic vector dictionary includes feature words related to each core word comprising 3700 important words in encyclopedia texts. More than 100,000-word semantic vectors were generated using statistical learning called a bootstrapping algorithm of article units based on manually created word semantic vectors of 3700 core words and encyclopedia texts. The learned word vector is also binary, and the bootstrapping algorithm is not a neural network base but a word count base, so the meaning of each dimension is perfectly maintained.

This proposal explains the experimental image retrieval system for 36,000 photographs with the word semantic vector dictionary made from the encyclopedia text. This system associates the words input by a user with the knowledge in the encyclopedia text and outputs ranked retrieval results.

The effectiveness of this associative retrieval method is shown by evaluating the content retrieval of images with a small benchmark and by making an adaptive learning function of word semantic vectors in a case in which a retrieval result is not the same

as a user’s subjective perspective would impose. One way to make the search result consistent with the user’s intention is to delete the feature word that does not fit the user’s intention for the word. This function can be only achieved because the word semantic vector has readability.

### **1.3.2 Proposal 2: Reputation Information Extraction from Twitter**

The paragraph vectors achieved state-of-the-art results on sentiment analysis at the time of publication [34]. The practical problem is that a large collection of documents is required for overcoming the sparsity of words in short text.

When applying reputation information extraction to Twitter sentences, we introduce the word semantic vector constructed manually to solve performance degradation due to the sparsity of the appearing word in Twitter [27]. The word semantic vector dictionary consists of 20,330 core words, which give the relationship with 266 feature words. We thought that applying the idea in Proposal 1 to the paragraph vectors can solve the problem on the sparseness of words. Even if the word sparsity problem exists in Twitter, we expect to learn the context information with the paragraph vector using feature words expanded from core words in tweets based on the word semantic vector dictionary.

The proposed method indicated a macro average F-score of 71.9 for positive and negative prediction in the 3-class classifications of positive, neutral, and negative in the benchmark of a specific smartphone product brand, which consists of about 12,000 tweets. The method exceeds the macro average F-score of the conventional method, paragraph vectors, by 3.2 points. Also, the method can perform error analysis using expanded feature words. Moreover, its effectiveness is larger even if a minimum benchmark consisting of a variety of products exists.

### **1.3.3 Proposal 3: Social Media Mining**

The problem with distributed representations generated by neural networks is that the meaning of the features is difficult to understand. Thus, tests and improvement in the qualities of natural language processing applications (e.g., sentiment analysis) are necessary because the distributed representations are not readable as is.

We propose a new method that gives a specific meaning to each node of a hidden layer by introducing a manually created word semantic vector dictionary into the initial

weights and by using paragraph vector models [25, 26]. We generate seed vectors, which are distributed representations between core words and 266 feature words, by recursively extending the word semantic vector dictionary using the retrofitting algorithms [13]. We conducted experiments to test the hypotheses and then evaluated the expandability of the method. Our experimental results demonstrated that the learned vector is better than the performance of the existing paragraph vector in the evaluation of the Twitter sentiment analysis task for a single domain benchmark. Also, we determined the readability of document embedding in a user test. A total of 52.4% of the top five weighted hidden nodes were related to tweets where one of the paragraph vector models learned the document embedding.

For the expandability evaluation of the method, we constructed a diverse, large-scale, and reliable sentiment analysis benchmark and improved the dictionary for the purpose of distributed representations [26]. We also conducted a word similarity task using a Wikipedia corpus to test the domain-independence of the method [26]. We found the expandability results of the proposed method are better than or comparable to the performance of the conventional method. Also, the objective and subjective evaluation support each hidden node maintaining a specific meaning. Thus, our method succeeds in improving readability and can be applied to mining of social media such as Twitter by using each feature word as a conceptual axis.

#### **1.3.4 Resource 1: Word Semantic Vector Dictionary**

In Proposal 1, a human expert constructed a word semantic vector dictionary consisting of 3700 core words and 266 feature words [20, 21, 23, 22]. Then, the number of core words constructed by the human expert was increased to 20,330 words to extend the application of associative retrieval [24]. In Proposal 3, we improved our dictionary with the aim of allowing third parties to reproduce our method. The number of feature words in the dictionary is 264.

#### **1.3.5 Resource 2: Evaluation Benchmark for Sentiment Analysis**

This dictionary is expected to be used as a new application field of word semantic vectors for extracting reputation information from Twitter. Sharp Corporation’s four-day time series analysis of product related tweets are shown in Figure 3. If we can extract the

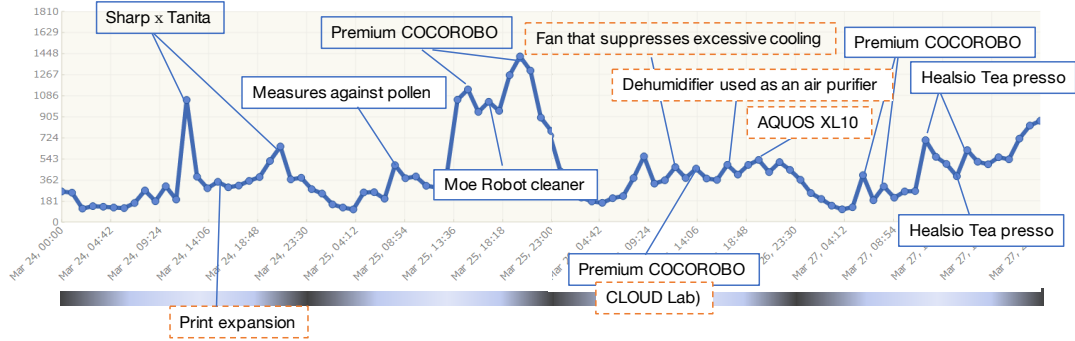


Figure 3. Time series analysis of product related tweets on Sharp Corp (3/24/2014 - 3/27). Topics surrounded by solid squares indicate excitement from user's tweets, and items surrounded by dotted squares are related to media.

reputation information of products on positive and negative with high accuracy, it will provide beneficial information for product planning and quality support. Therefore, evaluation benchmarks related to reputation information extraction were constructed, as follows.

First, we built two types of single domain benchmark for each smartphone product brand [27]. The benchmarks were used for the evaluation of Proposal 2. Also, the benchmark of one product brand was used to test the hypothesis of Proposal 3.

Next, we constructed a large-scale sentiment analysis benchmark with diversity [26]. The benchmark was used to evaluate the expandability of Proposal 3. This benchmark is scheduled to be released.

## 1.4 Rest of this thesis

The rest of this thesis is organized as follows (see also Figure 2).

**Chapter 2:** Text mining is the key technology to extract value from unstructured text. We first introduce two kinds of approach to construct commonsense knowledge base because our aim is to achieve commonsense knowledge to solve keyword problems and the word sparsity. Next, we describe manually created semantic lexicons, WordNet, commonly used in text mining, and an automatically generated sentiment lexicon. Also, we explain the idea of context vectors stimulated in this thesis and word2vec and

paragraph vector models as distributed representations. Finally, we explain SemEval of Twitter sentiment analysis shared tasks and describe the dataset and evaluation method.

**Chapter 3:** We propose a realistic method that encodes knowledge in an encyclopedia using a small manually created word semantic vector dictionary and encyclopedia text and associative image retrieval based on the knowledge. Also, we describe an adaptive learning function which can be only achieved because the word semantic vector has readability.

**Chapter 4:** We first introduce test collections for the evaluation of Japanese information retrieval systems, which was developed by a working group under the Special Interest Group on Database Systems of the Information Processing Society of Japan. I participated in the working group from the time of establishment in 1993 for the objective evaluation of our associative retrieval system. The distribution of the test collection had been discontinued. Also, we describe the single domain benchmark and diverse benchmark for the evaluation of sentiment analysis in Japanese Twitter.

**Chapter 5:** We propose a reputation information extraction method, which is used for product planning and quality support, from Twitter. We describe an integration method to learn feature words expanded by using the word semantic vector dictionary with the paragraph vector models. Also, we consider the error analysis using the expanded feature words and the evaluation of the small but diverse benchmark.

**Chapter 6:** We propose a new method of automatically learning readable distributed representations using the word2vec and paragraph vector models based on the word semantic vector dictionary for social media mining. Also, we describe the expandability evaluation of the method.

**Chapter 7:** We summarize the contribution of the thesis and discuss directions for future work.





## **2. Text Mining**

### **2.1 Introduction**

The subjects of this doctoral dissertation are “sentiment analysis in Twitter” and “associative image retrieval using knowledge of encyclopedia,” which are applications of text mining technology. In this thesis, we use word semantic vectors which are distributed representations to encode encyclopedia text. Attempts to construct two kinds of commonsense knowledge bases with the similar purpose have been continuing for many years. One is Cyc, and the other is ConceptNet. Section 2.2 introduces these commonsense knowledge bases.

The purpose of this thesis is to solve the problem of the sparseness of words and the readability of distributed representations by combining external dictionary and shallow neural networks. Therefore, Section 2.3 describes semantic lexicons and Section 2.4 describes distributed representations.

The standard data sets are essential to evaluate the progress of technology. Section 2.5 discusses the shared task of sentiment analysis in Twitter.

### **2.2 Commonsense Knowledge Bases**

We now describe Cyc, which is capable of highly accurate reasoning using ontology created by experts but is extremely difficult with adding information. Also, we describe ConceptNet, which is developed using crowdsourcing and is not accurate, but robust against errors.

#### **2.2.1 Cyc**

Cyc project is aimed at building a commonsense knowledge base and constructing a similar reasoning system like a human, started by Lenat in 1984 [35]. In Cyc, commonsense knowledge is registered manually by experts using a description language, which is an augmentation of first-order predicate calculus, called CycL. The Cyc Knowledge Base (KB) consists of terms and the assertions which relate those terms. These assertions include both simple facts and rules. The language is English only. The Cyc KB is divided into thousands of “contexts,” each of which is a collection of assertions that

share a common set of assumptions. As of July 2017, according to Cyc's web page<sup>1</sup>, the Cyc KB contains over five hundred thousand terms, including about seventeen thousand types of relations, and about seven million assertions relating these terms.

### 2.2.2 ConceptNet

ConceptNet [57, 56] started as a part of the crowdsourcing project Open Mind Common Sense(OMCS) launched in 1999 at the MIT Media Lab. OMCS is taking an approach to acquire commonsense knowledge by using the Internet from the general public including Japanese without using experts like Cyc. ConceptNet is an open, multilingual knowledge graph that connects words and phrases (terms) with labeled, weighted edges (assertions). The assertions are represented as triples of their start node (term), relation label, and end node (term). There are a core set of 36 relations in ConceptNet5.5 [56]. As of July 2017, according to ConceptNet's web page<sup>2</sup>, the knowledge base contains approximately 28 million statements (edges) and ten languages have core support, and 304 languages are supported in total. Using ConceptNet5.5, when searching for "Wedding" in Japanese, there is "Happiness" in the related term. However, when searching for "Wedding" in English, "Happiness" in English is not in the related term.

## 2.3 Semantic Lexicon

We now describe the general purpose WordNet created by experts and explain a Sentiment Lexicon automatically generated from the Twitter data for sentiment analysis tasks.

### 2.3.1 WordNet

WordNet [41], which was created at Princeton University, is the most commonly used lexical database for English. WordNet consists of synsets in that words are grouped by synonyms, and each synset corresponds to a concept. The statistical information of the database of WordNet 3.0 is as follows<sup>3</sup>.

- 117,659 concepts (the number of synsets)

---

<sup>1</sup><http://www.opencyc.org/>

<sup>2</sup><http://conceptnet.io/>

<sup>3</sup>Princeton University "About WordNet." 2010. <<http://wordnet.princeton.edu>>

- 155,287 words (Noun: 117,798, Verb: 11,529, Adjective: 21,479, Adverb: 4481)
- 206,941 word-synset pairs

The relationships between concepts (synsets) of nouns are as follows [41].

- Hyponymy (sub-name) and hypernymy (super-name) are transitive relations between synsets. This is the so-called “Is\_a” or “Class inclusion” relationship.
- Meronymy (part-name) and holonymy (whole-name) are complex semantic relations. This is the so-called “Part\_whole” relationship. WordNet distinguished component parts, substantive parts, and member parts.

There is a report [13] that the number of edges which connect words to synsets, hypernymy, and hyponymy is 934,705. Also, the Japanese version of WordNet has been developed based on synsets of the English version [8].

### 2.3.2 Sentiment Lexicon

Kiritchenko et al. generated a sentiment lexicon specialized in tweets from the corpus of positive or negative tweets automatically collected from a specific hashtag of Twitter [29]. Hashtag terms were chosen from “positive” and “negative” entries in Roget’s Thesaurus <sup>4</sup>. They were 30 positive terms such as #good and #excellent, and 47 negative terms such #bad and #terrible. They collected 2 million English tweets and deleted tweets that did not contain at least two content words from Roget’s Thesaurus. 775,000 tweets remained as Hashtag Sentiment Corpus. If there were even one of 30 positive hashtags, it was labeled as a positive tweet, and if there were even one of 47 negative hashtags, it was labeled as a negative tweet. Sentiment scores for each term were calculated from labeled tweets using PMI (Pointwise Mutual Information). The final sentiment lexicon consists of 39,413 unigrams, 178,851 bigrams and 308,808 non-contiguous pairs (unigram-unigram, unigram-bigram, and bigram-bigram pairs).

---

<sup>4</sup><http://www.gutenberg.org/ebooks/10681>

## 2.4 Distributed Representation

Each node corresponds to one concept, and two nodes are connected by labeled edges like aforementioned CyC, ConceptNet, and WordNet. Such representation is called *localist*. On the other hand, the definition of “distributed representation” is as follows [16, 17].

- Each neuron represents something. Therefore, it must be a *localist* representation.
- Distributed representation means a many-to-many relationship between two types of representations of concepts and neurons. Each concept is represented by many neurons, and each neuron participates in representing many concepts.

In this section, we explain the context vector, word2vec, and paragraph vector models as distributed representation.

### 2.4.1 Context Vector Representation

Gallant [14] proposed a method for representing context information based upon microfeatures proposed by Waltz and Pollack [61]. First, a feature space consisting of  $n$  common words or concepts is defined. Second, for each word  $k$ , a context vector,  $V^k$ , is defined to be an  $n$ -dimensional vector and interpret each element of  $V^k$  as follows [14].

- $V_j^k \approx \text{strongly positive}$  if word  $k$  is **strongly** associated with feature  $j$
- $V_j^k \approx 0$  if word  $k$  is not associated with feature  $j$
- $V_j^k \approx \text{strongly negative}$  if word  $k$  is **strongly** contradicts with feature  $j$

When a feature space consists of five features, e.g., [nature, city, noise, animal, green] and respective element values take an integer of five steps from -2 to +2, a feature vector of a word, e.g.,  $V^{\text{mountain}}$  might be expressed as [+2, -2, -2, +1, +1].

### 2.4.2 Word2vec and Paragraph Vector Models

It has been reported that Word2vec, which is invented by Tomas Mikolov et al., can construct vectors with similar weights for similar word and phrases when context information is learned as features using a neural network [38, 39, 40]. What is obtained

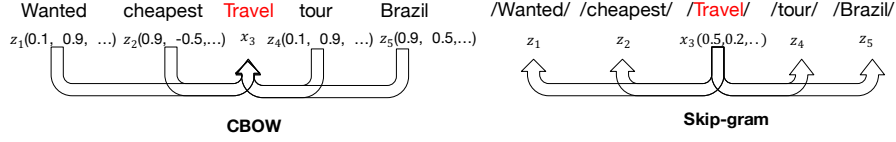


Figure 4. Word2vec learning images.

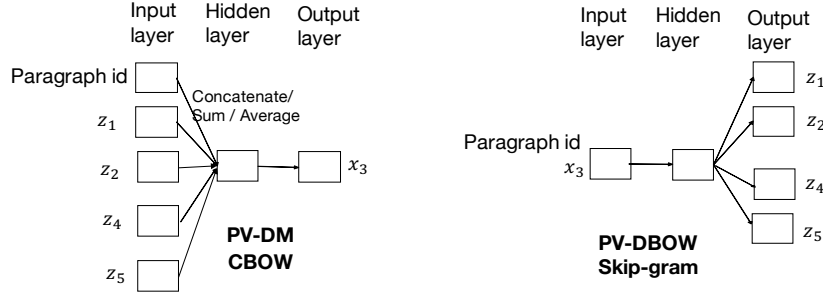


Figure 5. Word2vec and paragraph vector models.

by neural net learning is not only the semantic vectors of words but also the semantic analogy of words by adding and subtracting word vectors. For example, the word having the vector closest to “Tokyo - Japan + France” is “Paris.”

Here, we explain the distributed representation learning of two variants of word2vec models [7]. In the example sentence in Figure 4, the target word for which distributed representation is to be learned is “Travel,” and all words except “Travel” appearing in the example sentence are context words. In the CBOW (Continuous Bag Of Words) model of word2vec, if context words such as “Wanted,” “cheapest,” “tour,” and “Brazil” appear in the sentence, it updates the distributed representation of each word so as to be predicted whether or not the target word “Travel” appears in the sentence. In the word2vec models, two  $d$ -dimensional vectors are assigned to each word. in the case the  $i_{th}$  word appears as a target word, it uses the target word vector  $x_i$  of that word, and in the case that word appears as a context word, use the context word vector  $z_i$  of that word. On the other hand, in the Skip-gram model of word2vec, the objective is to predict the context word  $z_i$  appearing in the context using the target word  $x_i$ .

Figure 5 presents two variants of word2vec and paragraph vector models [34]. The distributed memory model of paragraph vectors (PV-DM) predicts the target word

vector, which is weights between hidden layer and output layer, of the center word  $x_3$  from the context vector, which is weights between input layer and hidden layer, obtained by adding a paragraph ID to input words within the context window. The CBOW model of word2vec does not add the paragraph ID to the input layer, but it is fundamentally the same as the PV-DM. The number of nodes in the hidden layer corresponds to the number of vector dimensions.

The paragraph vector with a distributed bag of words (PV-DBOW) learns the paragraph vector to predict the context word vectors of randomly selected surrounding words within the context window. Skip-gram of word2vec is used to learn the vectors of the target words in the PV-DBOW. In Skip-gram, the target word vector is learned so that the inner product of the target word vector and the context word vector of the surrounding words is larger than the inner product of the context word vector of words other than the surrounding words.

PV-DM and PV-DBOW randomly set initial values of word vectors and paragraph vectors. PV-DBOW randomly selects words in the window without considering the word order, PV-DM predicts the target word vector of the center word while sequentially moving the window, so that we can use the word order information for learning. PV-DM can also concatenate context word vectors of surrounding words in the hidden layer, in which case the word order is held as the context vector.

## 2.5 SemEval: Shared Tasks on Sentiment Analysis in Twitter

Association for Computational Linguistics holds the shared tasks named “Sentiment Analysis in Twitter” from 2013 [45, 51, 50, 44, 49]. SemEval-2017 Task 4 in the fifth year consists of the following five subtasks [49].

- **Subtask A:** Classification of 3-class (Positive, Negative, or Neutral) targeted at one tweet.
- **Subtask B:** Classification of 2-class (Positive, Negative) targeted at a given topic included in one tweet.
- **Subtask C:** Classification of 5-class (StronglyPositive, WeaklyPositive, Neutral, WeaklyNegative, and StronglyNegative) targeted at a given topic included in one tweet.

Table 1. Datasets of Subtask A from 2013 to 2017.

Dataset	Positive	Negative	Neutral	Total
2013-train	3662	1466	4600	9728
2013-dev	575	340	739	1654
2013-test	1572	601	1640	3813
2014-test	982	202	669	1853
2015-test	1040	365	987	2392
2016-train	3094	2043	863	6000
2016-dev	844	391	765	2000
2016-devtest	994	325	681	2000
2016-test	7059	3231	10,342	20,632
2017-train	19,902	7840	22,591	50,333
(Total-train)	(26,658)	(11,349)	(28,054)	(66,061)
(Total-dev)	(2413)	(1056)	(2185)	(5654)
2017-test	2375	3792	5937	12,284

- **Subtask D:** Distribution of 2-class (positive, negative) for a set of tweets targeted at a given topic.
- **Subtask E:** Distribution of 5-class (StronglyPositive, WeaklyPositive, Neutral, WeaklyNegative, and StronglyNegative) for a set of tweets targeted at a given topic.

From SemEval-2017 Arabic is added to all subtasks in addition to English. Also, the user profiles that posted tweets are provided. Subtask A has been held consecutively and gathering the most participant (more than 40 teams each year), since 2013. The evaluation benchmarks created in this thesis correspond to subtask B because they are the classification of each tweet targeted at topics. However, subtask B is given a topic for each tweet, but the benchmarks of this thesis have one topic or eight topics in advance. Rather, the benchmark of this thesis is closer to the dataset of subtask A which determines the polarity of the entire tweet.

### 2.5.1 Datasets

The datasets of subtask A is shown in Table 1. A new test set is offered to the participating team every year. For the test sets, more than 20,000 tweets and 12,000

Table 2. The confusion matrix. Cell XY stands for “the number of tweets that the classifier labeled X and the gold standard labels as Y.” P, N, U stand for Positive, Negative, Neutral, respectively [44]

Predicted	Gold Standard		
	Positive	Negative	Neutral
Positive	PP	PN	PU
Negative	NP	NN	NU
Neutral	UP	UN	UU

tweets were offered in 2016 and 2017 respectively, compared to the fact that a few thousands of tweets had been provided by 2015. On the other hand, the past training and development sets can be used. In 2014 and 2015 there was no new training and development set, but 10,000 tweets were added in 2016, and 50,333 tweets were added in 2017.

### 2.5.2 Evaluation Measure

We describe the macro average F-score that had been adopted from SemEval-2013 to SemEval-2016 as the primary evaluation measure of Subtask A which performs tweet polarity classification. Subtask A is a single-label multi-class (SLMC) classification task [44]. Each tweet must be classified as one of the following 3-class {Positive, Negative, Neutral}. The confusion matrix is shown in Table 2. Eq. 1 shows positive precision  $P_{pos}$  and Eq. 2 shows positive recall  $R_{pos}$ . Eq. 3 show the F-score  $F_{pos}$  for the positive class. Likewise, the F-score  $F_{neg}$  for the negative class can be calculated.

$$P_{pos} = \frac{PP}{PP + PN + PU} \quad (1)$$

$$R_{pos} = \frac{PP}{PP + NP + UP} \quad (2)$$

$$F_{pos} = \frac{2P_{pos}R_{pos}}{P_{pos} + R_{pos}} \quad (3)$$

The overall score, named macro average F-score, is computed as the average of the F-score for the positive and for the negative classes as shown in Eq. 4.

$$F = \frac{F_{pos} + F_{neg}}{2} \quad (4)$$



However, SemEval-2017 [49] adopted average recall, or AvgRec, which is recall averaged across the positive, negative, and neutral classes, as shown in Eq. 5 as the primary evaluation measure of the same subtask. And macro average F-score was demoted to the second evaluation measure.

$$AvgRec = \frac{R_{pos} + R_{neg} + R_{neutral}}{3} \quad (5)$$

The difference between them is whether or not to evaluate the neutral class directly. Because the dataset as shown in Table 1 has a significant bias among 3-class, it is considered that the average recall of 3-class was adopted. However, considering actual use, it is important that the extraction accuracy of positive and negative tweets is equally high, and even if neutral can be extracted with high recall, it will not be evaluated. Also, it is important to compare with past results in the same dataset, so macro average F-score becomes a substantial evaluation measure even in SemEval-2017. Therefore, in this thesis, the macro average F-score (hereinafter referred to as F-score) is adopted as the evaluation measure of sentiment analysis.

## 2.6 Conclusion

In this chapter, we reviewed the text mining technology. As commonsense knowledge bases, Cyc built by experts and ConceptNet built by crowdsourcing were introduced, and as semantic lexicons, WordNet constructed by experts, a sentiment lexicon generated automatically from Twitter were introduced. These are local (symbolic) representations. On the other hand, we showed the definition of distributed representations, the idea of context vectors of the distributed representation, and the word2vec / paragraph vector of distributed representation learning. Also, we described the shared tasks SemEval on sentiment analysis in Twitter, its dataset, and the evaluation measure.

In this thesis, we propose a distributed representation including context information as a semantic lexicon, named a word semantic vector dictionary. Also, we solve the sparsity of words on Twitter through an integration of the distributed representation learning and the semantic lexicon. Moreover, we improve the readability of distributed representations, which is a problem of distributed representation learning.



### **3. Associative Retrieval Using Knowledge in Encyclopedia Text**

#### **3.1 Introduction**

Many images are being digitalized due to expanding use of communication networks, there is a necessity for an image database that contains many of these digitalized images for which users can search using any query.

A method of keyword retrieval was previously applied to the retrieval of image data, such as paintings and photographs, by attaching the keywords expressing the image characteristics to the image data. However, applying keyword retrieval to the image retrieval means that database developers must select and attach many keywords to the image data to satisfy user demands.

Keyword retrieval involves the use of thesaurus deployment to acquire additional keywords. However, a thesaurus is a one-dimensional language concept as described in Section 2.3.1. For example, the words “happy” and “marriage” are considered to be different concepts. As a result, the image attached to the keyword “marriage” cannot be retrieved by inputting the query “happy people.” To resolve these problems, an attempt is made to give computers a common sense knowledge by using symbols to code all knowledge in an encyclopedia as described in Section 2.2.1. However, the encyclopedia stores a huge amount of knowledge, and arranging and describing this knowledge is very difficult.

Retrieval using impression words such as “refreshing” and subjective retrieval using words such as “photos of urban image” is inadequate because of differences in human sensibility and language conceptualization. To resolve these problems and perform subjective retrieval, a method of calculating a correlation between vectors acquired from image color information and the impression of words is proposed as a way to retrieve information using the natural language based on physical characteristics of the images [32, 31].

We propose a method that codes knowledge in an encyclopedia using a multidimensional expression (word semantic vector) of the concept that is associated with words geared to retrieve information. Keywords are not relied upon—instead an associative function (possessed by human beings) and subjective retrieval [23, 22] are imposed.

This chapter explains the experimental image retrieval system for 36,000 photographs with a word semantic vector dictionary constructed from an encyclopedia text. Because this image retrieval system is not restricted by keywords, a word semantic vector dictionary of 100,000 or more words can be constructed using an encyclopedia text. Also, there is a characteristic that there is no bias in the weighting of the meaning of words by being based on an encyclopedia rather than a thesaurus.

## 3.2 Creating a Word Semantic Vector Dictionary Using an Encyclopedia Text

This section explores a way to construct a word semantic vector dictionary with more than 100,000 words from the encyclopedia<sup>5</sup>. Each of 132,000 items has one explanatory paragraph (the merged text–item name and explanation will be called a record). 3700 core words were selected by frequency analysis for all encyclopedia text, and the word semantic vectors of these core words were created manually (by a human expert). Then we constructed a word semantic vector dictionary of more than 100,000 words that were included in the encyclopedia by using a bootstrapping algorithm with the word semantic vectors of these core words and a large amount of encyclopedia text.

### 3.2.1 Definition of Word Semantic Vector

The basic concept of word semantic vector is based on microfeatures proposed by Waltz and Pollack [61] and a context vector as described in Section 2.4.1. The word semantic vector is a vector representation of a relationship having many feature words. These feature words are n-category classifications of concepts, with each dimension of the vector a point on an n-dimensional vector space corresponding to one feature word and representing a meaning. For example, binary word semantic vector  $\mathbf{X} = (x_1, \dots, x_n)$  is expressed as Eq. 6:

$$x_l = \begin{cases} 0 & \text{if the feature word } l \text{ does not relate to the word} \\ 1 & \text{if the feature word } l \text{ relates to the word} \end{cases} \quad (6)$$

If the feature words are {human, sad, art, science, exciting, politics}, the word

---

<sup>5</sup>Britannica Electronic Reference Guide (CD-ROM version). TBS BRITANNICA CO. LTD., 1992

Table 3. Classification of feature words.

Six large classifications	Examples of 29 upper concepts	Examples of 266 feature words
Human·Life	Human Creature	Human, Name, Male, Female, Child Animal, Bird, Insect, Microbe, Plant
Human environment	Artificiality Traffic·Communication	Tool, Mechanical·Component, Building Communication, Traffic·Transportation
Natural environment	Area Nature	Place name, Country name, Japan, City Land, Mountain, Sky, Ocean
Abstract concept	Spirit·Psychology Abstract concept	Sense, Emotion, Happiness, Sadness State·Aspect, Change, Relationship
Physics·Substance	Motion Physical characteristics	Motion, Halt, Dynamic, Static Warmth, Weight, Lightness, Flexible
Civilization·Information	Humanities Science	Race, Knowledge, Speech Mathematics, Physics, Astronomy

Table 4. Grant criteria by logical relationship.

Logical relationship	Core words	Feature words
Class inclusion	Autumn	Season
Synonym relationship	Idea	Thought
Part-whole relationship	Leg	Human body

semantic vector of the core word “pilot” becomes (1, 0, 0, 1, 1, 0). In those papers [61, 14], the relationship between words and each feature word is expressed with multi values according to its relational strength. The relationships are expressed in binary form here. Word semantic vectors are created independently of the fields because the relationships are binarily expressed.

### 3.2.2 Selection of Feature Words and Criteria for Manual Entries

We must select a combination of feature words to classify the content of the encyclopedia. We select the 266 feature words belonging to six large classifications and 29 upper concepts as in Table 3. Criteria for the manual entries of the feature words for the core words are as follows:

Table 5. Grant criteria by associative relationship.

Core words	Feature words
Love	Kindness, Warmth
Up	Economy, Video
Leg	Car, Traffic·Transportation

- Make feature word entries using logical and associative relationships. The logical relationship is a direct connection between the core words and the feature words as in Table 4. The associative relationship is a connection of which one is reminded by associations as in Table 5.
- The large classifications and upper concepts of the feature words are for references of the classification. The criteria for the manual entries of the feature words are feature words themselves. For example, the feature word “warmth” is classified under the upper concept “physical characteristics.” However, the feature word “warmth” is made the entry for the core word “love” by an association of concepts involving “warmth of heart” (more typically in English, “heartwarming”).

### 3.2.3 Bootstrapping Algorithm of Word Semantic Vectors

Making entries of 266 feature words for all words in the encyclopedia is not realistic. Therefore, we have developed a method called bootstrapping algorithm based on the following hypotheses.

**Hypothesis 1.** If a record includes more than a constant (for example, five) number of core words, appropriate context information of this record is expressed by the record semantic vector, which is the weighted sum of the word semantic vectors of these core words.

**Hypothesis 2.** The word semantic vectors can express appropriate context information of these words by the weighted sum of the semantic vectors of the records including these words.

Figure 6 is a processing flow of the bootstrapping algorithm. Words are extracted

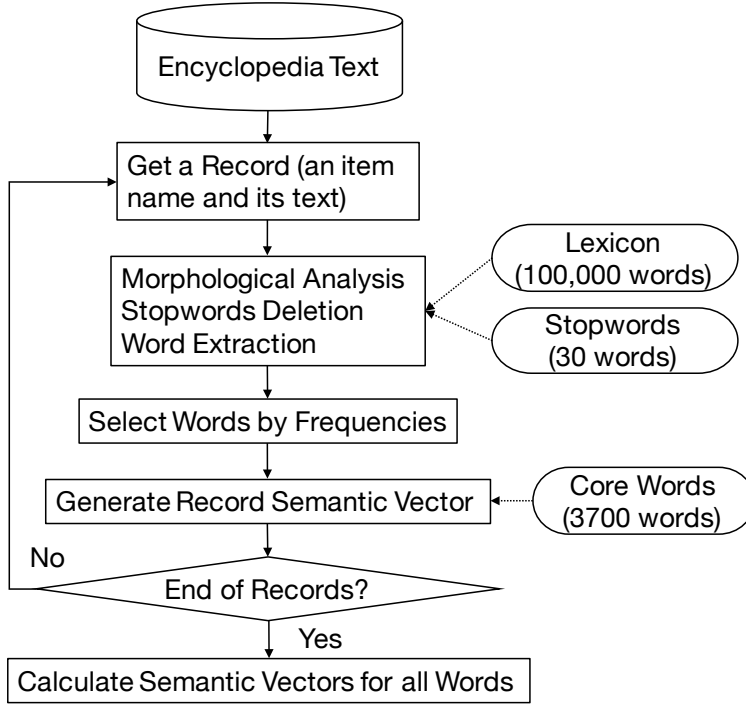


Figure 6. Flow of bootstrapping algorithm for word semantic vectors.

from each record by a morphological analysis, and stop words are deleted. Then, the semantic vector  $RSV_i$  of the record  $i$ , including more than a constant number of core words, is calculated based on Hypothesis 1 as in Eq. 7:

$$RSV_i = k \sum W_j P_{ij} (a_1 x_{j1}, \dots, a_n x_{jn}) \quad (7)$$

Here  $W_j$  is the weight of the word  $j$  in a text database,  $P_{ij}$  is the frequency of the word  $j$  in the record  $i$ ,  $(a_1, \dots, a_n)$  is the weight vector of each feature word  $(x_{j1}, \dots, x_{jn})$  is the word semantic vector of the word  $j$  and  $k$  is the coefficient for normalizing the record vectors to a constant length. The weight  $W_j$  of the word is defined by the reciprocal of the occurrence probabilities of the word  $j$  in the text database as in Eq. 8:

$$W_j = \log (N/df_j) \quad (8)$$

Here  $N$  is the number of all records and  $df_j$  is the number of the record including the word  $j$ . Eq. 8 shows that some words factor more importantly (more heavily weighted)

in the retrieval of records. The weight of the feature word is the reciprocal of the occurrence probabilities of each feature word of the core words as in Eq. 9.

$$a_l = \log (NC/f_{cl}) \quad (9)$$

Here  $a_l$  is the weight of the feature word  $l$ ,  $NC$  is the number of all core words and  $f_{cl}$  is the number of core words made in the entry for feature word  $l$ . Eq. 9 shows that feature words, when made entries for many core words, are not as effective for semantic classification of the words. The entries of feature words for these words must be done for all 266 feature words to employ the 266-dimensional word semantic vector effectively. Eq. 7 calculates the semantic vector by modulating the word semantic vectors of the core words in the record with feature word weights to turn down feature word inclination using the bootstrapping algorithm, and by normalizing the sum of the multiplied vectors and the weights of the words in the text database by the word frequencies as words weighted in the records. The weighted sum  $T'_j$  of the semantic vectors of the records, including the word  $j$ , is calculated by Eq. 10 for every word  $j$  (including the core words) extracted from the record  $i$ :

$$T'_j = \sum_i P_{ij} RSV_i \quad (10)$$

After the semantic vectors of all records are generated by Eq. 7 and the word semantic vectors of all words are calculated by Eq. 10,  $T'_j$  is normalized to the word semantic vector  $T_j$  of the word for which the number of entries of the feature words is within a constant range (e.g., between 8 and 25).

### 3.3 Image Retrieval Using the Word Semantic Vector

We explain here a prototype of the digital photo catalog system, which adapts the word semantic vectors constructed from the encyclopedia for the captions of the photographs; it also is capable of retrieving photographs associatively with only a few keywords. It is expected that designers will be the first users of this digital photo catalog system and so we have developed the prototype on the personal computer “Apple Macintosh” that is used widely by the designers.



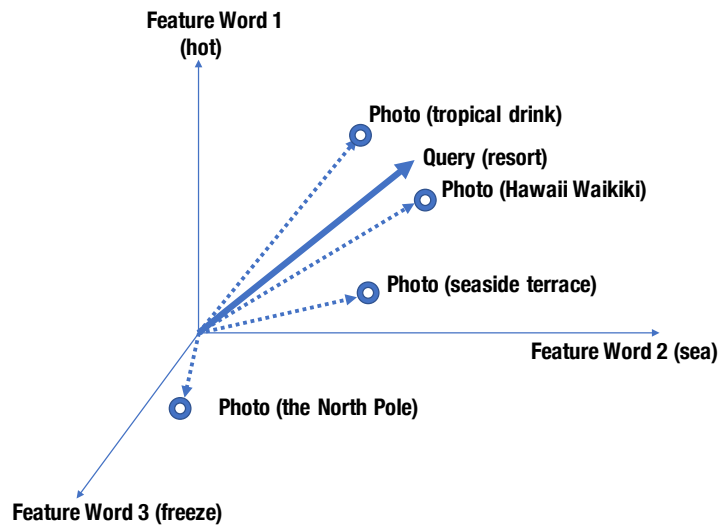


Figure 7. Image retrieval in semantic vector space.

### 3.3.1 Outline of the Digital Photo Catalog System

This system retrieves the images using close semantic vectors and a semantic vector that has been inputted in the form of a query using natural language; the ranked results are then displayed. The image semantic vectors are constructed from these image captions. Figure 7 shows this conceptual image. The photographs and query are settled on a three-dimensional vector space using as an example the features words of “hot,” “sea,” and “freeze.” In this example, the system ranks highly the photographs “Hawaii Waikiki” and “tropical drink” as close to the query word “resort.”

**3.3.1.1 Image data** Image data used in this digital photo catalog system include 36,000 photographs on a laser disc<sup>6</sup>. Photographs on this laser disc are digitalized to 320 x 240 pixels and compressed to an average one-twentieth of size by JPEG. The total size of these image data is 433 Mbytes.

**3.3.1.2 Text data** Each photograph is attached not only to a title, a classification code, and an area code but also to an image code such as gorgeousness, romance, and

<sup>6</sup>JPRC Photo Disc. Japan Photo Research Center, Co., Inc. (JPRC), 1988

Table 6. Captions to describe images.

Image title	Caption
Telephone box	England, London, night, red, street corner, telephone box
Matterhorn	Switzerland, summer, blue sky, train, mountain, matterhorn

Table 7. Semantic vectors for images.

Image title	Feature words
Telephone box	4: Tool, machine 3: Communication, solid, speech, ... 2: Act, shape, design, ... 1: Europe, country, name, money, ...
Matterhorn	4: Mountain, height, Europe, ... 3: Overseas, forestry, state, ... 1: Car, blue, static, ...

amazement, and a theme code shows the subject of each image. The retrieval system's database stores the image captions, which are titles and text data decoded from all these codes attached to the images. Table 6 provides an example of these image captions. The total size of these text data is 3.4 Mbytes.

**3.3.1.3 Indexing of images** First, the word semantic vector dictionary constructed using encyclopedia text is adapted for the image captions, and image semantic vectors are made. Then, the image database is constructed by registering the image data and the image semantic vectors. The adaptation of the word semantic vector dictionary for the image captions modifies the weights, and the word semantic vectors of the words included in the captions using the bootstrapping algorithm, the word semantic vector dictionary is used in place of a core word dictionary and image captions are used instead of encyclopedia text in Figure 6. The word semantic vectors constructed from the encyclopedia text is used for the words not included in the image captions. We assume that the semantic inference of a sentence is defined completely by the semantic clues of the words included in the sentence. We define the sentence vector as the sum of

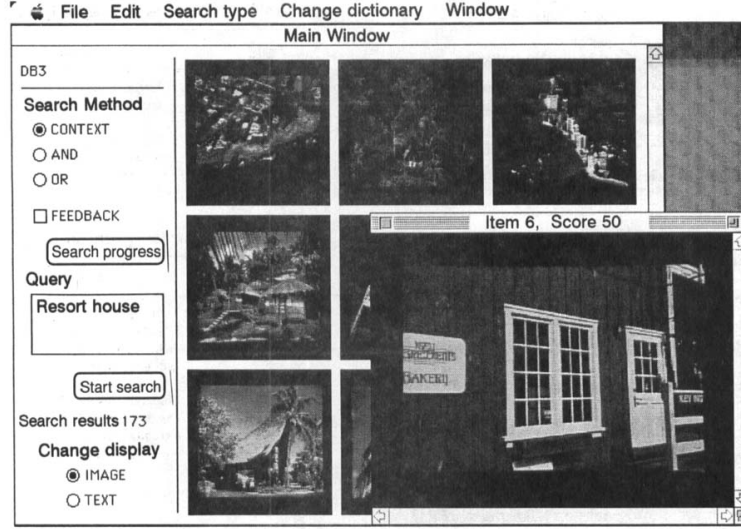


Figure 8. An example of image retrieval (Query “Resort House”). (Photos are from JPRC Photo Disc.)

all word semantic vectors included in the sentence. Namely, sentence semantic vector  $X$  is calculated by Eq. 11:

$$X = \sum_{k=1}^m p_k T_k \quad (11)$$

Here  $T_k$  ( $1 \leq k \leq m$ ) is the word semantic vector of the word included in the sentence,  $p_k$  is the word weight and  $m$  is the number of words included in the sentence. Then, only  $X$  is normalized to a constant length (depending on the situation) because the direction of  $X$  is the only important one. Table 7 is an example of image semantic vectors.

**3.3.1.4 User interface** Figure 8 gives us an image retrieval example. This system can do relevance feedback and retrieve text database. Table 8 shows the functions of each button on the Figure 8 example. Relevance feedback is a way of retrieving images that are close to the semantic vector of the image selected by a user. The system can display full-size images by clicking the retrieved images. Figure 8 presents the retrieved result of the query “Resort House.” The full-size image ranked at the 6th location is

Table 8. Buttons on the main window of the digital photo catalog system.

CONTEXT	Retrieval only using semantic vectors
AND	Retrieval using semantic vectors for records satisfying Logical AND of words extracted from query
OR	Retrieval using semantic vectors for records satisfying Logical OR of words extracted from query
START	Start of retrieval
IMAGE	Display of image retrieval results
TEXT	Display of text retrieval results
FEEDBACK	Relevance of feedback

also displayed. Figure 9 shows the text of these retrieval results and the image caption ranked at the 6th location. This caption does not include the word “resort,” but this record is retrieved at a high rank because the semantic vectors “Marina del Rey” and “resort” are similar to each other. Figure 10 is an example of the relevance feedback with a semantic vector of the image ranked at the 6th location. These retrieval results are all photographs of “House at Marina del Rey,” except for the one ranked at the 7th location (which is also a house in the United States).

### 3.3.2 Evaluation of the Digital Photo Catalog System

This retrieval system ranks all 36,000 images by calculating the distance (inner product) between the semantic vectors of the query and each image. This interface can display the top 200 of these images.

We evaluated the precision of the image content retrieval in the following way. First, we extracted for evaluation some sample data from the image data and manually tested the queries to get correct answers for these sample data. We evaluated by inputting these queries for the sample data and checking as to whether the correct answers appeared in the retrieval results. The image semantic vectors do not necessarily represent the subjective characteristics of these images accurately because these vectors are mechanically constructed from image captions. Thus, any evaluation must be based on a subjective analogy to evaluate this image retrieval system. The test collection of these queries and their correct answers for the sample data were made then by looking

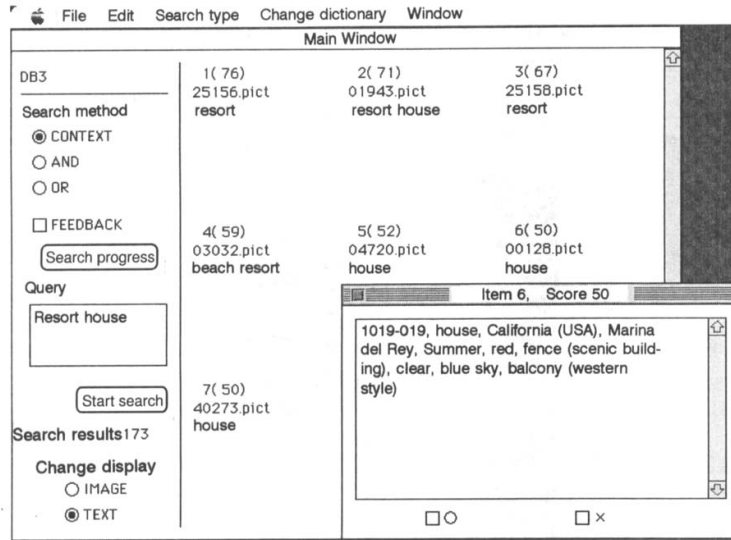


Figure 9. An example of text retrieval (Query “Resort House”).

at only images but not image captions. Selecting images corresponding to a prepared query or a query corresponding to an image is complex and inefficient because word queries must correspond to subjective impressions of visual images. On the other hand, comparing the analogies subjectively using images is easy. Thus, when we made the test collection for evaluation, we classified the images to some groups so that images judged to be similar to each other were classified as part of the same group; we then made the queries corresponding to each group. We made three queries for each group because the evaluation results might change significantly with these queries. We classified 120 images into 14 groups using subjective analogy; we obtained 42 queries. Table 9 is an example of the caption of a representative image and three queries in each group for two groups in the benchmark group.

This evaluation method determines whether an adequate image is retrieved by inputting a vague impression for image retrieval. We retrieved images by inputting 42 queries for 120 image data, and at least one image belonging to the corresponding group to the query is retrieved in the upper three ranks for 38 queries. The other four queries cannot be retrieved in the upper three ranks because the image caption differs from the subjective characteristic of the image, and the feature (semantic vector) made the entries for the words in the query different from a subjective human perspective. In

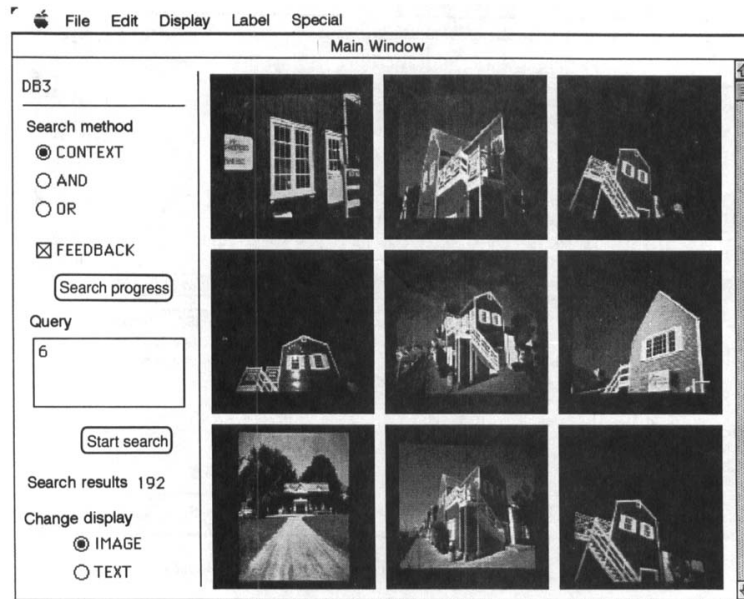


Figure 10. An example of image retrieval using relevance feedback. (Photos are from JPRC Photo Disc.)

this case, we can adapt retrieval results to reflect individual subjectivity by using the learning functions of the word semantic vectors from users who are referred to in the next section. We did relevance feedback by using the representative image captions in each group, and the images of the sample group were then retrieved in the upper ten ranks at a rate (recall) of 75 percent. From these evaluation results, we confirmed the effectiveness of using a semantic vector in image retrieval.

### 3.3.3 Learning Function by User

We have made the following three learning functions to adapt the image semantic vectors to the retrieval environment of each user.

#### (i) Replacement of word semantic vectors

A user is not satisfied with the retrieval by a query, word A, and the retrieval result by another query, word B, satisfies the retrieval by the query, word A. In this case, replacing the word semantic vector of the word A by the word semantic

Table 9. Examples of a caption for the representative image in each group and its queries.

Caption for representative image	Query
Evening scenery of Manhattan, New York, Brown, building, building street, High-rise building, scenery building	1. high-rise building, city 2. big city with many buildings 3. panoramic view of big city
Countryside and fog, Switzerland, summer, green, pastoral, Farm village, scenery nature, refreshing	4. mountain, village, nature 5. green mountain area 6. countryside mountain area

vector of the word B improves the retrieval result by the query, word A. For example, if the images related to literature are retrieved in high ranks by the word “cat (in Chinese character)” because of the association of “I Am a Cat (the title of a novel),” and cat images are retrieved in high ranks by the word “cat (in katakana, Japanese syllabary),” replacing the word semantic vector of the word “cat (in Chinese character)” by the word semantic vector of the word “cat (in katakana)” improves the retrieval result of the word “cat (in Chinese character)” to the images related to “I Am a Cat” and the cat images.

(ii) Modification of word semantic vectors

A user is not satisfied with the retrieval result by a query, word A. In this case, modifying the entries of the feature words for the word semantic vector of the word A improves the retrieval result by the query, word A. For example, if many photographs of “mountain” and “beach” are retrieved in high ranks with the word “countryside,” and these images do not satisfy the user’s demand for “countryside,” removing the entries of the feature words “Mountain” and “Ocean” for the word semantic vector of “countryside” improves the retrieval result in high ranks by the word “countryside” to the images of “mountain” and “beach.”

(iii) Modification of word semantic vectors by marking ○ or × to retrieval results

A user is not satisfied with the retrieval result by a query, word A. In this case, first, the user marks ○s for good retrieval results and ×s for unsatisfactory retrieval results. Next, the system normalizes the sum of semantic vectors of the

retrieval results marked with  $\bigcirc$  and the sum of semantic vectors of the retrieval results marked with  $\times$ . Then, the system calculates the difference between the normalized vector marked with  $\bigcirc$  and the normalized vector marked with  $\times$ . If the value of each feature word of the difference vector is greater than threshold value  $t$ , the system makes a feature word entry. If the value of each feature word of the difference vector is less than threshold value  $u$ , the system removes the feature word entry. Thereby, the images marked with  $\bigcirc$  are retrieved in higher ranks by the word A, and the images marked with  $\times$  are retrieved in lower ranks. In the result, retrieval results with word A are improved.

We confirm that image retrieval results by queries of one word can be modified to satisfy the user by using these three modification functions of the word semantic vectors by the user. In these three modification functions, marking  $\bigcirc$  or  $\times$  is the simplest method to improve the retrieval results using user-subjectivity. However, because the feature word modification of the words by the adaptive learning influences the semantic vectors of the images (including those words in the captions), the user must consider the meaning of each feature word when modifying these feature words.

### 3.4 Conclusion

In this chapter, we proposed here a way to construct a word semantic vector dictionary representing encyclopedia knowledge by using a vector. We have shown its effectiveness by experimentally creating the image database using a word semantic vector dictionary. We confirm that retrieval based on encyclopedia knowledge can be done if the information of the photograph captions is used by adapting the word semantic vector dictionary containing more than 100,000 words (from the encyclopedia) to the captions of 36,000 photographs. Using this word semantic vector dictionary is effective for retrieval of multimedia information because the database can be made easily; unexpected but useful results are included in the retrieval of multimedia information.

In comparison with existing research [61, 14] that use the multidimensional representation of the concept associated with languages, this work uses the multilayered classification of feature words, the clarification of criteria for entries of these feature words for core words, the use of the multidimensional representation for coding encyclopedia knowledge, and the application of the coded knowledge to index multimedia



information.

In this chapter, 3700 important words were selected as the core words from the encyclopedia texts, and a human expert assigned relationships with 266 feature words to construct a word semantic vector dictionary. Because the validity of associative retrieval using a word semantic vector dictionary was confirmed, we expanded the target area of unstructured texts to newspaper articles, World Wide Web, instruction manuals. In response to the expansion of the target area, the number of core words expanded from 3700 words to 20,300 words [24].

Lastly, we will discuss the word semantic vector dictionary also used in the other two proposals of this thesis as follows.

- (i) The difference in sense concerning core words by individuals.

Regarding the logical relevance of feature words with the core words, it is considered that there are few individual differences because binary values were given by a human expert who is a lexicographer based on the clear criteria as shown in Table 4. Regarding the associative relevance of feature words with the core word, we think that individual differences will occur. However, because feature words of queries are readable, you can customize by modifying feature words that do not fit your sense as shown in Section 3.3.3.

- (ii) On the appropriateness of 266 feature words.

For the core words of 3700 words and the extended 20,300 words, at least one feature word, nine feature words on average are given. Words with a high level of expertise such as scientific and technical terms are only given one feature word like a field name. We think that it is more appropriate to treat such scientific and technical words as keywords than to deal with semantically.

- (iii) On how to automatically increase core words.

As shown in Chapter 2, WordNet shows the logical relevance between words, ConceptNet shows the associative relevance between words. Using these local representations, we will consider how to expand the current 20,300 core words in the future.



## **4. Evaluation Benchmark**

### **4.1 Introduction**

In Chapter 3, we constructed a word semantic vector dictionary of over 100,000 words from the encyclopedia text and developed an image database for 36,000 images and their captions based on the dictionary. However, the benchmark constructed for evaluating associative images retrieval is 42 queries and 14 groups of the correct answer sets for 120 image data. The benchmark is not enough as an evaluation of information retrieval system.

At that time, because there was no standard benchmark for evaluating Japanese information retrieval system, we created the first test collection, named BMIR-J1 [28, 11, 10], under the Information Processing Society’s working group. In section 4.2, we will mainly describe the first full-benchmark, named BMIR-J2, in its successor [30, 53].

As for sentiment analysis in Twitter, as shown in Section 2.5, English datasets are being developed and accumulated with SemEval. However, because there is no Japanese dataset, we have constructed two types of single domain small benchmarks and a large-scale benchmark with diversity in Section 4.3.

### **4.2 Test Collection for Evaluation of Japanese Information Retrieval Systems**

In Europe and the US, a variety of standard test collections are available for objective evaluation of information retrieval (IR) systems [15]. However, in spite of the growing interest in IR research in Japan, no test collection for Japanese existed until 1993. Unlike European languages, no explicit word boundaries exist in Japanese text. Moreover, Japanese has several character classes, each of which fulfills different linguistic functions. Because of these language specific problems, Japanese test collections are indispensable for fostering research and development of Japanese IR systems. In 1993, we formed a working group under the Special Interest Group on Database Systems of the Information Processing Society of Japan (IPSJ SIGDBS), to design and construct Japanese test collections. In 1996, a preliminary version called BMIR-J1 was distributed to fifty sites. BMIR-J1 consisted of three basic parts: 600 documents, 60 queries, and the correct answers, which were from 5 to 30 documents, given to each query. Our

working group proposed five functions that the best IR system should have and marked functions that were needed to get the correct answers to each question. The benchmark could be applied using these major characteristics to evaluate various kinds of retrieval methods.

Although BMIR-J1 was a small preliminary collection, we collected many useful comments from the monitor users. We enlarged the collection size and revised the queries and relevance assessments based on comments from BMIR-J1 users. Distribution of BMIR-J2, the first complete Japanese test collection, started in March 1998. It contained sixty queries and the IDs of 5080 newspaper articles in the fields of economics and engineering. The queries were also classified into five categories based on the functions the system is likely to use to interpret them correctly and to retrieve relevant texts. This collection had two levels of relevance, topically relevant and partially relevant.

In designing a test collection, we must consider which types of IR systems are applicable, what kinds of texts should be included, and how relevant texts should be selected. On the basis of our experience, we discuss these design issues.

#### **4.2.1 Overview of BMIR-J2**

**4.2.1.1 Text selection** Initially, we considered text sources such as patent descriptions and technical papers for use in the collection. However, we settled on newspaper articles because they are generally available on CD-ROMs and are widely used. Thus, BMIR-J2 uses only articles from the Mainichi newspapers. In response to requests from the BMIR-J1 users, we focused on the fields of economics and engineering.

**4.2.1.2 Query development** A total of sixty queries were developed in BMIR-J2. Each query consists of a natural language phrase describing a user's needs and of additional comments augmenting the phrase. In the following English translation, the first line, showing a query phrase, is followed by two narrative lines:

Q:F=oxoxo :“Utilizing solar energy”

Q:N-1: Retrieve texts mentioning uses of solar energy.

Q:N-2: Include texts concerning generating electricity and drying things with solar heat.

**4.2.1.3 Query categorization** Information retrieval often requires a deep understanding of the user's needs. However, merely matching words appearing in the query with those in the texts is sometimes enough to retrieve relevant texts. Categorizing queries according to the functions that the system uses to process them is worthwhile because a system's architecture ranges from simple word matching to rich natural language processing. This variety of categories allows BMIR-J2 users to select queries that their systems will be able to deal with effectively. BMIR-J2 provides five categories, shown as "F=oxoxo" in the previous example. Each digit, denoted as "○" (necessary) or "×" (unnecessary), represents the following functions, starting from the left:

- **The basic function:** The relevant texts can be retrieved simply using words extracted from the query or their thesaurus expansion.
- **The numeric range function:** The system will need to handle a numeric range description such as "a reduction of more than one thousand employees."
- **The syntactic function:** Analyzing a syntactic relationship among query words will help understand the query.
- **The semantic function:** A semantic analysis will be required to understand the query. In a query, "a trend in the facsimile market," a system must determine how to identify a description as a "trend" because the word "trend" may not appear in the text.
- **The world knowledge function:** World knowledge will be required to process the query. In a query, "a joint business operation between companies in different types of industries," the system must know that the companies belong to different types of industries. Such information is often missing in the texts or the system's lexicon.

**4.2.1.4 Relevance assessments** The relevancy of articles for each query was assessed in the following four steps in general [12]:

1. Pre-screening of possibly relevant texts: A Boolean query was made manually in such a way that most relevant texts were likely to be retrieved. The results from one or two IR systems were merged.

2. Relevance assessments by database searchers: Several database searchers first assessed the relevancy from the results obtained in step 1. They also cross-checked their work.
3. Relevance assessments by one of the WG members: Relevant texts of each query from step 2 were checked and corrected by one of the WG members.
4. Relevance assessments by another WG member: Relevant texts from step 3 were cross-checked and corrected by another WG member.

## **4.2.2 Design issues and discussion**

**4.2.2.1 Types of test collection** BMIR-J2 allows a batch-wise evaluation of IR systems. Need is increasing for other types of test collection, such as by text categorization and interactive testing with users. These issues should be examined further.

**4.2.2.2 Collection size** How many texts and queries, as well as how many relevant texts for each query, are necessary to make an IR system evaluation statistically sound? Queries with few relevant texts tend to have a great influence on the overall system performance.

**4.2.2.3 Text source and domains** Texts from limited domains and specific sources enable a controlled system evaluation. Texts from various domains and sources, on the other hand, enable an evaluation that is close to real-world situations. The choice of text selection is valuable, but other issues such as copyright of text sources and getting enough funding for the license, are often dominant. To get around the copyright issue, we included a list of article IDs, rather than full texts, in the distribution set.

**4.2.2.4 Relevance assessment process** As recent test collections go, BMIR-J2 is not particularly large [15]. Even with 5080 articles and sixty queries, checking all possibilities manually is already overwhelming work. Thus, pre-screening relevant texts using IR systems was necessary in developing our collection. In order not to miss relevant texts in pre-screening, a pooling method using IR systems with different architectures is desirable. Relevance results should be cross-checked by at least two

people. Our experience led us to use cross-checking, which gave us a chance to adjust relevance assessment criteria and to keep them consistent among WG members.

### 4.3 Japanese Twitter Sentiment Analysis Benchmark

In this section, we describe the evaluation benchmarks for the purpose of extracting opinions of individuals that are used for product planning and quality support from Twitter.

First, we introduce two type of single domain benchmarks. The product brands of two kinds of smartphones (hereinafter referred to as Product A manufactured by a company A, and Product B manufactured by a company B) are targeted.

Next, we explain a sentiment analysis benchmark consisting of 38,576 tweets of 8 categories of products, services, and organizations for Japanese Twitter.

#### 4.3.1 Single Domain Benchmark

We used crowdsourcing to label each tweet for each product brand for benchmark construction. There are four kinds of labels as follows.

- **Positive:** Positive opinions are stated for the target product brand in a tweet.
- **Negative:** Negative opinions are stated for the target product brand in a tweet.
- **Neutral:** Opinions on the target product brand in a tweet are neither positive nor negative.
- **Unrelated:** Opinions on the target product brand are not stated.

We collected tweets with search expressions for 13 months from October 2014 to November 2015 regarding smartphones A and B. We collected tweets based on the product name for Product B, but we conducted the AND search with topics on the smartphone for Product A because there are other product genres with the same product brand name. After sampling to about 10%, we deleted tweets only of katakana, and bots. We allocated five workers to each tweet using crowdsourcing and assigned labels for approximate 35,000 tweets. The cost for labeling is about 20,000 yen, and the benchmark can be constructed with a realistic budget.

Table 10. Configuration of the benchmark.

Dataset	Positive	Negative	Neutral	Total	Unrelated
Product A					
Training set	1122 (35%)	1023 (32%)	1065 (33%)	3210	
Dev. set	280 (35%)	256 (32%)	266 (33%)	802	
Test set	280 (35%)	256 (32%)	266 (33%)	802	
Total number	1682	1535	1597	4814	8216
Product B					
Training set	3654 (41%)	2375 (27%)	2802 (32%)	8831	
Dev. set	608 (41%)	396 (27%)	467 (32%)	1471	
Test set	609 (41%)	396 (27%)	467 (32%)	1472	
Total number	4871	3167	3736	11,774	5998
Unlabeled tweets (No label)					
560,853					

Table 11. Statistical information of each benchmark.

	Product A	Product B
Number of tweets	4814	11,774
Number of unique words	8782	11,901
Percentage of “word occurrence frequency one time”	54.8%	54.3%
Average number of words per tweets	14.5	10.0
Number of tweets containing core words	4606	10,352
Average number of core words per tweet	5.4	3.3
Average number of feature word types per tweet	26.3	19.1

Table 10 shows the configuration of benchmarks constructed using crowdsourcing. The evaluation benchmark for Product A consisted of 4814 tweets (training set: 3210, development set: 802, test set: 802), the benchmark for Product B consisted of 11,774 tweets (training set: 8831, development set: 1471, test set: 1472). Table 11 shows the statistical information on the evaluation benchmarks after pre-processing. The average number of words per tweet was 14.5 words for Product A, 10.0 words for Product B and the average number of core words extracted from tweets were 5.4 words per tweet for Product A and 3.3 words per tweet for Product B. This indicates that the number of tweets for Product A’s benchmark is as small as about 40% for Product B, but the



Table 12. Number of labels given to tweets by crowdsourcing

Categories	Number of tweets	Positive	Negative	Neutral	Positive&Negative	Unrelated	Number of labels
Smartphone A	130,650	2906	5188	16,054	594	68,158	92,900
Smartphone B	482,036	5655	9531	51,900	603	18,884	86,573
Smartphone C	1,155,034	3543	6176	45,568	408	28,844	84,539
Robot cleaner A	11,664	741	311	6894	41	4371	12,358
Robot cleaner B	307,156	954	1089	20,654	55	48,092	70,844
Printing service	275,097	3887	3484	30,176	241	35,514	73,302
Maker A	187,584	744	4421	40,950	75	26,358	72,548
Maker B	169,532	1503	937	13,624	80	54,891	71,035
Total number	2,718,753	19,933	31,137	225,820	2097	285,112	564,099

Table 13. Japanese Twitter Sentiment Analysis Benchmark

Dataset	Positive	Negative	2 class total	Neutral	Unrelated	Total
Training set	10,100	15,618	25,718	137,089	180,186	342,993
Dev. set	2,525	3,904	6,429	34,272	45,046	85,747
Test set	2,525	3,904	6,429	34,272	45,046	85,747
Total number	15,150	23,426	38,576	205,633	270,278	514,487
No label	2,204,266					

sentence length is long.

Tweets with the same number of votes on multiple labels in the 1st place and the “unrelated” labels were removed from the experiments in Chapter 5 and Chapter 6. In addition to this benchmark, there are about 560,000 unlabeled tweets before crowdsourcing. These were only subjected to preprocessing from the tweets collected Product A and B, eliminating duplicates with the benchmarks, and used for the initial learning of word vectors for the purpose of reducing the sparsity of words.

#### 4.3.2 Diverse and Large-scale Benchmark

We constructed a large-scale benchmark with diversity using crowdsourcing to test the expandability of our method. Table 12 shows the number of tweets collected for each category and the number of tweets labeled by crowdsourcing. We collected tweets with keywords such as product brands for 13 months from October 2014 to November 2015 regarding smartphones A and B and for 13 months from January 2015 to February 2016 regarding other categories. We labeled the tweets of each category

using crowdsourcing. Five workers were assigned to each tweet, and labels were given by majority vote. In the case of ties in the majority vote, multiple labels were assigned to one tweet. Five kinds of labels were used, as follows.

- **Positive:** Positive opinions are stated for specific features of a target category in a tweet.
- **Negative:** Negative opinions are stated for specific features of a target category in a tweet.
- **Neutral:** Opinions on a target category in a tweet are neither positive nor negative in the sense of the aforementioned.
- **Positive&Negative:** Both positive and negative opinions are stated in a tweet.
- **Unrelated:** Opinions on a target category are not stated.

Here, the point was to clarify positive and negative judgment criteria as “specific features” to perform crowdsourcing work efficiently on large-scale tweets. Therefore, the first two examples as we will show in Section 6.4.4, which have been usually considered positive, became neutral in the labeling this time. Also, the latter three cases that express their opinions on specific features “sound” and “charging” are subjects of positive and negative judgment.

Table 13 shows the Japanese Twitter sentiment analysis benchmark created based on the result of labeling by crowdsourcing. We classified tweets that are given multiple labels including “positive & negative” labels in the “No label.” The 2-class classifications of positive and negative were part of a diverse and large-scale benchmark of 38,576 tweets, comparable with the datasets of SemEval as shown in Section 2.5.1. In addition, the reliability of the benchmark was high because the positive and negative classification standards were clear. Furthermore, because the specific features of products, services, and organizations were stated in the tweet, the test was considered to be a benchmark to extract tweets that are useful for product planning and quality support.

## 4.4 Conclusion

One of the unique features of BMIR-J1 and BMIR-J2 is that the queries are classified into five groups according to interpretation levels needed for processing these queries. English test collection do not have these groups in themselves. In our associative retrieval, the evaluation of semantic function was performed mainly. Also, we extracted frequently-used important words from newspaper articles and increased the word semantic vector dictionary constructed by a human expert from 3700 words to 20,300 words [24]. Because the purpose of each information retrieval system is different, the existence of a test collection that can be evaluated by the function is extremely important.

As for the diverse and large-scale benchmark for Japanese Twitter, although this benchmark is reliable in the 2-class classifications, it can also be used for the 3-class classifications including neutral. However, because tweets such as merely “like,” “dislike,” “want,” “not needed” about the target products, services and organizations were included in neutral this time, the benchmark became very unbalanced data in the 3-class classifications. Therefore, it is a challenging task to increase the macro average F-score of the positive and negative prediction in the 3-class classifications. As an actual application, it would be impossible to respond to requests for product planning and quality support unless it is possible to classify positive and negative tweets with high accuracy after filtering the unrelated tweets and neutral tweets from this benchmark.



## 5. Reputation Information Extraction from Twitter

### 5.1 Introduction

In the enterprise, the necessity of extracting reputation information from Twitter is increasing to grasp the customer's voice about new products and their quality promptly. A method is required to clarify the opinions of individuals from Twitter whose sentence length is short and to extract positive and negative opinions on their products with high accuracy.

It is considered that distributed representation learning like paragraph vectors as shown in Section 2.4.2 can be utilized as a method of calculating the similarity of tweets to achieve this objective. Also, the paragraph vectors achieved state-of-the-art results on sentiment analysis at the time of publication [34].

When learning a paragraph vector based on words included in one tweet, the sparseness of words becomes a problem. For example, in the Twitter data used in the experiment in Section 5.4, the whole vocabulary is approximately 93,000 words (64% is two occurrences or less), and one tweet is 7.6 words on average. Except for the product name and words of two occurrences or less, it is necessary to estimate the context from about two words. However, if words are sparse, the accuracy of the tweet's similarity becomes worse, because learning of a paragraph vector requires at least a four-word window length (the number of words of context information).

In contrast to the distributed representations obtained by learning, we proposed word semantic vectors, constructed using a human expert with context information as features in Chapter 3. The word semantic vector expresses the relationship between a word and feature words as a binary value that is related or unrelated. The feature word corresponds to each dimension of the word semantic vector, and it consists of 266 conceptual classifications. The core words are composed of 20,330 important words extracted using frequency analysis of Japanese newspapers and encyclopedias. The word semantic vector dictionary is a dictionary listing feature words related to core words. We thought that applying the idea as proposed in Chapter 3 to the paragraph vector can solve the problem on the sparseness of words. We propose an integration method to learn feature words expanded using the word semantic vector dictionary with a paragraph vector model. Context information can be added to short tweets by expanding feature words using the dictionary.

We explain concretely why the accuracy of sentiment analysis improves by integrating the word semantic vector and the paragraph vector. The paragraph vector is used for learning of tweets and learning of feature words expanded as context information. In a tweet “Recommend the new Product B,” the paragraph vector of the tweet and the word vectors of the three words, “recommend,” “new” and “Product B,” can be learned by the PV-DM and PV-DBOW models. The PV-DM uses information of the word order, and the PV-DBOW does not use information about the word order, as described in Section 2.4.2. In the word expansion using the dictionary, the common feature words “trend-popularity,” “value-quality,” “excellent,” “positive,” and “strong” are added to the Tweet as context information for the core words “new,” and “recommend.” Also, particular feature words of each core word are added to the tweet. The ability to consider the logical and associative relevance in this way is an advantage of integrating the word semantic vectors and the paragraph vectors. The context information updates the word vectors using the PV-DBOW model and learns vectors similar to the tweet’s paragraph vectors. If this tweet is a tweet with a positive label, a classifier will be learned so that this tweet is classified as positive by supervised learning at the latter stage in the case the words “new,” and “recommend” appear in a tweet. As a result, feature words that did not exist in the original tweet are also updated to the word vectors that are classified as positive. For example, in the case there is no training data of a tweet “progress by Product B as expected,” it can be classified as positive by the proposed method. Because the both core words “progress” and “expect” are expanded to the feature words “excellent,” “positive” and “strong.” The sparseness of words can be improved by adding feature words as context information without collecting large tweets, compared to learning tweets with paragraph vectors.

We created reputation information extraction benchmarks from Twitter to test the effectiveness of the proposed method with actual data as described in Section 4.3.1. Experimentally we evaluate whether the accuracy of sentiment analysis can be improved by integrating the learning of feature words expanded as context information with the paragraph vector, comparing with the tweet paragraph vector only.

## 5.2 Related Work

We describe mainly the feature expansion method using semantic lexicons presented at SemEval as described in Chapter 2, compared with the proposed method. Twitter-

Hawk [4] experimented with features expansion using WordNet [41]. In addition to the text and the lexicon features, the features were expanded from WordNet and the sentiment analysis was performed using the linear kernel SVM. However, the F-score decreased by about 2 points, compared with the case without using WordNet. WordNet is a one-dimensional classification of the concept of a word, and the word is given a relationship with a synonym group of 117,000 as described in Section 2.3.1. When WordNet is used for feature expansion, it requires 117,000 dimensions, which makes it a very sparse feature vector. It is considered to be the reason for degrading the accuracy of sentiment analysis. On the other hand, the word semantic vector of the proposed method is a 266-dimensional vector representation of the logical and associative relation with 266 kinds of feature words. In TwitterHawk, the extended feature vector is directly input to the linear SVM, but in the proposed method the word vector is updated so that expanded feature words learn context information. As a result, in the proposed method, the F-score improved by about 3 points, compared with the case without word expansion.

NRC Canada [42], which showed the highest score in SemEval-2013, constructed a classifier using a linear kernel SVM for over millions of features including N-gram, sentiment lexicons (three types of manually created lexicons, two types of Twitter’s automatically generated lexicons). As a result, the F-score was 69.02, indicating that about 5 points of the F-score were improved by the sentiment lexicons automatically generated from 775,000 tweets as described in Section 2.3.2.

INESC-ID [1] using a structured Skip-gram model similar to word2vec learned 600-dimensional word vectors with unlabeled tweets of 52 million with no need for the feature engineering such as NRC-CANADA. The classifier also learned the mapping from 600 dimensions to the ten-dimensional subspace with the labeled data using the same neural network. The test set of SemEval-2013 was 72.09, and SemEval-2015 was 65.21, both showing the highest level in SemEval-2015.

NRC-Canada used feature vectors of more than 1 million dimensions, and INESC-ID used 52 million tweets for pre-learning of word vectors. Considering actual use of reputation information extraction, these can be said to be unrealistic from the viewpoint of economic rationality. The reason why large-scale features and tweets are necessary to improve the accuracy of sentiment analysis is due to the sparseness of words on Twitter. In the proposed method, the F-score of the same level was obtained using

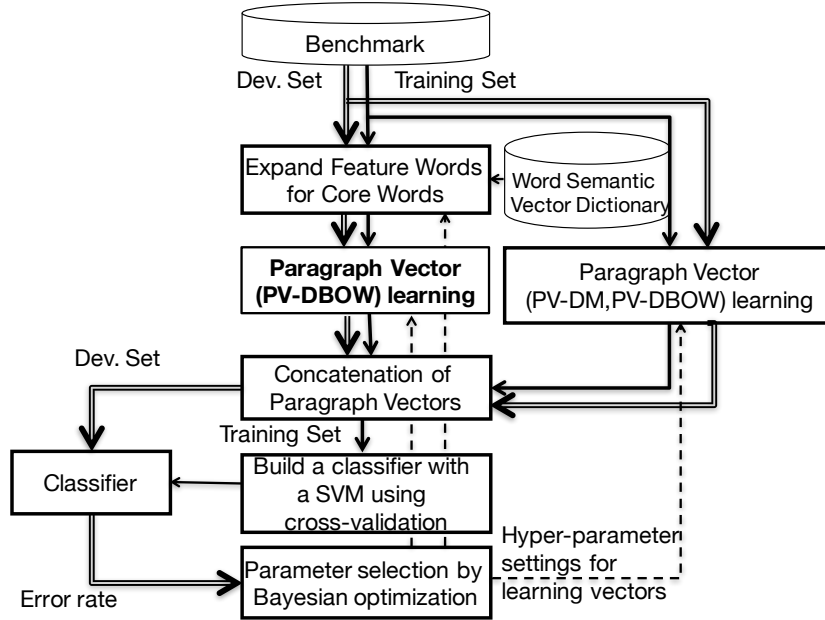


Figure 11. Overall flow of the proposed method.

about 0.1 percent's features of NRC-CANADA, about one percent's unlabeled tweets of INESC-ID. It is considered to be the effect of learning the context information of a specific field even if the words are sparse, by integrating the word semantic vector with the paragraph vector.

## 5.3 Proposed Method

### 5.3.1 Reputation Information Extraction Method from Twitter (Proposed Method)

In this section, we propose a reputation information extraction method used for product development and quality support.

In this chapter, we propose an integration method to learn words expanded using the word semantic vector dictionary with a paragraph vector model to solve the problem of word sparsity in Twitter. We aim to learn the paragraph vector, which is difficult to understand adequately in Twitter of short sentence length, using the method of expanding the core words extracted from the tweets to the feature words.

Figure 11 shows the flow of the proposed method. The solid arrows indicate the



flow of the training set. The training set consists of “labeled” tweets and a large number of unlabeled tweets. Double arrows indicate the flow of development set (tweets and labels), and dashed arrows indicate parameter adjustments necessary for learning paragraph vectors. Specifically, after preparing for the training set and development set, which is an evaluation benchmark, we construct the proposed method by the following procedure.

**[Step 1]** Word vectors are learned using two types of paragraph vectors (PV-DM, PV-DBOW) for unlabeled tweets of the training set. At the same time, the core words in the unlabeled tweets are expanded to the feature words using the word semantic vector dictionary, and the word vectors are learned by using the PV-DBOW model. This procedure (Step 1) can be omitted.

**[Step 2]** In learning the paragraph vector, the initial values of the word vectors are set to the word vectors learned in Step 1. When Step 1 is omitted, the initial values of the word vectors are default random setting. Next, the paragraph vectors (PV-DM, PV-DBOW) of the tweets with “label” in the training set are learned. Also, the paragraph vectors (PV-DBOW) are learned by expanding the core words in the tweets with “label” to feature words. However, core words expanded into feature words remain as learning objects. We combine these three types of paragraph vectors for each tweet and create a tweet feature vector. We build a tweet classifier with a support vector machine (SVM) using the labels of each training tweet as training data. The parameters of the SVM are determined by quadrant cross-validation and grid search to construct a classifier. In learning the training set, the vector of feature words is also updated in the PV-DBOW model.

**[Step 3]** Three types of paragraph vectors for the tweets of the development set are learned based on the word vectors learned by the training set in Step 2. Each tweet vector and its label of the development set is entered into the classifier, and the error rate  $[=100-(F_{pos}+F_{neg})/2]$  is measured. Bayesian optimization automatically adjusted the parameters of the paragraph vector learning so that the output of the objective function, which is the error rate, is minimized [55].

After Step 2 and Step 3 are repeated until the error rate converges, the hyper-parameter of the paragraph vector learning is determined.

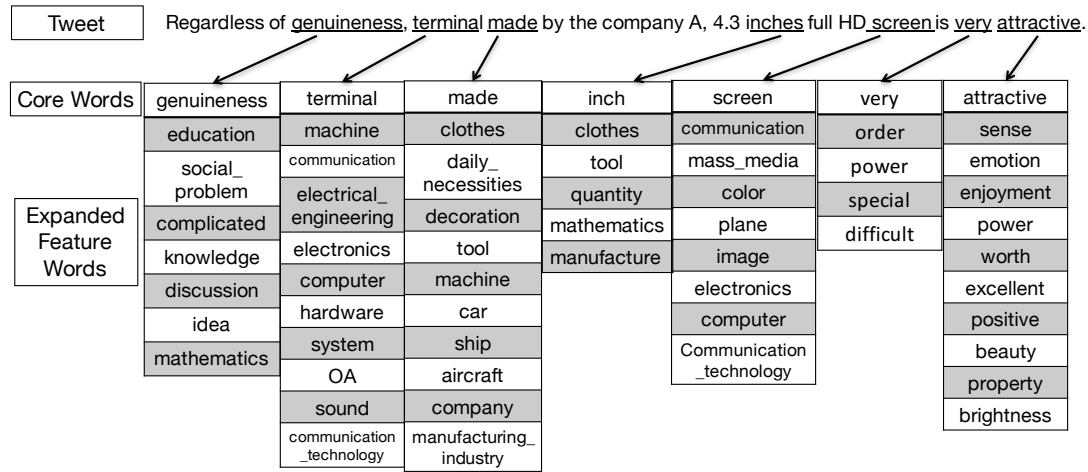


Figure 12. An example of extracted core words from a tweet and expanded feature words from the core words.

### 5.3.2 Learning Paragraph Vectors Using Expanded Feature Words

An example in which core words in tweets are expanded into feature words is shown in Figure 12. In the tweet shown in Figure 12, it contains seven core words “genuineness,” “terminal,” “made,” “inch,” “screen,” “very,” “attractive.” Because the hypothesis 1 of the bootstrap algorithm in Section 3.2.3 is satisfied, the semantic vector of the paragraph is considered to be able to learn appropriate context information from the semantic vectors of the core words (a combination of feature words). Here, PV-DBOW is used for constructing the semantic vector of paragraph (tweet) and updating semantic vectors of the words of hypothesis 2.

In the proposed method, the concept of word order disappears due to the word expansion, so the expansion of core words in the proposed method does not meet the PV-DM model. On the other hand, context information of tweets is added by the word expansion, so it is in good agreement with the PV-DBOW model which predicts context word vectors of surrounding words randomly from the target word. For this reason, the proposed method uses the PV-DBOW model to learn feature word-expanded tweets.

Core words are extracted from tweets after morphological analysis of tweets. However, the part of speech of core words to be expanded into feature words and the upper limit of the number of feature words to be expanded are determined as hyperparameters

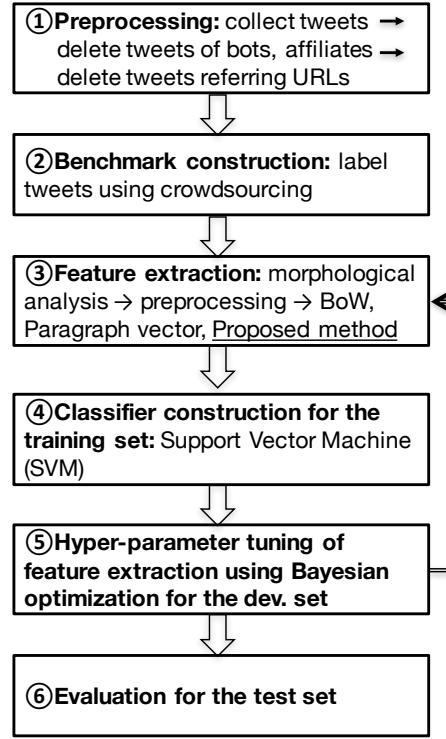


Figure 13. Experimental procedure of the extraction of the reputation information from Twitter.

of paragraph vector learning by considering the error rate of supervised learning in the later stage.

## 5.4 Experiment of Extracting Reputation Information from Twitter

In this section, we will explain the evaluation experiment conducted to test the effect of the proposed method.

### 5.4.1 Procedure

The experimental procedure for extracting the reputation information from Twitter is shown in Figure 13.

① In the preprocessing, we collected tweets based on keywords related to each product brand and excluded the tweets which contain bots, affiliate words, citation

URLs, and retweets.

② For the benchmark construction, we used crowdsourcing to label each tweet for each product brand. Details of the benchmark were described in Section 4.3.1.

③ In the feature extraction, a paragraph vector, a paragraph vector using word expansion by the word semantic vector dictionary of the proposed method, and Bag of Words (BoW) with dimensions of vocabularies extracted from all tweets as baselines were created as feature expressions. The following preprocessing was carried out for the tweets beforehand.

- Delete username (@user) and newline.
- Delete punctuation marks and symbols [“:”, “,”, “(”, “)”, “{”, “}”, “[”, “]”, “. ”, “, ”, “ ”, “#”]
- English character strings were unified into lowercase letters

Japanese morpheme analysis MeCab<sup>7</sup> and its dictionary mecab-ipadic-NEologd<sup>8</sup>, which expanded it by millions of new words and named entities from language resources on the Web, were used to extract words from tweets.

④ In the classifier construction for the training set, cross-validation was performed using SVM, and various parameters of SVM were determined. The experimental method is described in section 5.4.2.

⑤ In the Hyper-parameter tuning of feature extraction using Bayesian optimization for the development set, parameters of feature extraction were adjusted to minimize the error rate of the development set and fed back to the ③ feature extraction. The parameters to be adjusted were the window length, vector length, the number of learning times, the upper limit of the number of feature words to be expanded, and so on. We stopped the feedback loop when the error rate of the development set converged. When learning the training set we updated word vectors of the PV-DBOW, but in the development set, we learned only the paragraph vectors. Because the number of tweets was small, we did not update the word vectors of the PV-DBOW.

⑥ In the evaluation for the test set, the macro average F-score (F-score) described in Section 2.5.2 was evaluated by 3-class classifications of the test set by using the

---

<sup>7</sup><http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html>

<sup>8</sup><https://github.com/neologd/mecab-ipadic-neologd>

parameter of the feature extraction optimized based on the development set. In the evaluation for the test set, the development set was not included in the training set, and two types of evaluations were performed, one was updating the word vectors of the PV-DBOW, and the other was not updating the word vectors.

### 5.4.2 Experimental Method

#### (1) Evaluation experiment of baseline (BoW)

The feature representation of the tweet was set as a vector (BoW) whose dimensions were the vocabularies of both the training and development set of each product and the training and test set, and the frequencies of the vocabularies extracted from each tweet were used as the values of the vector.

For constructing a classifier by BoW and SVM, scikit-learn <sup>9</sup> which is a machine learning library for Python was used. The values of the kernel function and hyperparameters of SVM were decided by quadrant cross-validation and grid search for BoW of the training set. Evaluation of the development set and the test set was performed using the classifier of the constructed SVM.

#### (2) Evaluation experiment of paragraph vector

First, we learned word vectors based on approximately 560,000 unlabeled tweets. Then we learned the paragraph vector of the training set for each product. Finally, we learned paragraph vector of the development set. For the learning of the paragraph vector, gensim <sup>10</sup> which is a library of the topic model for Python was used. Regarding the various parameters of the paragraph vector and the vector generation method of the hidden layer described in Section 2.4.2, the range of appropriate values was determined by preliminary experiments and parameters for paragraph vector learning were determined using Bayesian optimization tool Spearmint <sup>11</sup>. Spearmint is designed to perform automatic experiments by adjusting multiple parameters so that the value of the objective function is minimized by as few iterations as possible [55]. The objective function here is the error rate of the development set.

---

<sup>9</sup><http://scikit-learn.org/stable/>

<sup>10</sup><http://radimrehurek.com/gensim/>

<sup>11</sup><https://github.com/HIPS/Spearmint>

The SVM classifier was constructed based on the quadrant cross-validation and grid search for the paragraph vectors of the training set by using scikit-learn. However, because the word vector was also learned during the cross-validation for the training set, the experiment setting differs from the evaluation for the development set. The error rate of the development set was measured using the constructed SVM classifier and the paragraph vector of the development set. Then the parameters of the next paragraph vector learning were selected by the Spearmin, and automatic experiments were repeated until it was judged that the error rate of the development set had converged. The convergence condition of the error rate was set as the case where the automatic experiment was repeated 100 times or more, and the minimum value was not updated in the error rate measurement in the automatic experiment more than 20 consecutive times.

### (3) Evaluation experiment of the proposed method

Against the evaluation experiment using the paragraph vector described in (2), feature word expansion of the core word included in the tweet described in Section 5.3 was performed for learning unlabeled tweets. Here, we expanded up to 7 feature words for core words of nouns, adjectives, adjective verbs, and verbs. Nine feature words were assigned on average for each core word in the classification order shown in Table 3. In the case of the candidates of feature word expansion were seven words or more, seven feature words were selected in order from the top of the classification. The reason is as follows. The occurrence probability of feature words classified as the “Human-environment,” “Abstract-concept,” “Physics-Substance,” and “Civilization-information” is high in the tweets of the product’s reputation. However, feature words classified as “Human-environment” and Abstract-concepts,” which is located at the center of the alignment order of feature words, are considered to be particularly important in sentiment analysis. Because unlabeled tweets were before crowdsourcing, so many meaningless tweets were included. There were about 109,000 tweets not including the core words of the aforementioned part of speech. We also learned word vectors including expanded feature words. Regarding learning of unlabeled tweets, parameters such as the number of expansion of feature words were determined by preliminary experiments using the training sets and development sets of Product A and Product B. We adopted the default value

Table 14. Evaluation results1: macro-averaged F score for predicting positive and negative tweets in the dev. set and the test set (the error rates).

	ProductA			ProductB		
	Dev.	Test	Test (Adapt)	Dev.	Test	Test (Adapt)
BoW	59.3 (40.7)	62.9 (37.1)		66.7 (33.3)	68.2 (31.8)	
PVEC	63.9 (36.1)	65.4 (34.6)	65.7* (34.3)	67.3 (32.7)	68.3 (31.7)	68.7** (31.3)
Proposed Method	66.4 (33.6)	67.7 (32.3)	67.7*** (32.3)	72.1 (27.9)	70.9 (29.1)	71.9**** (28.1)
vs.BoW	*p=3.9e-12 < 0.05			**p=0.004 < 0.05		
vs.PVEC	***p=4.1e-07 < 0.05			***p=5.6e-13 < 0.05		

of each parameter. Then whether or not to expand the noun of the core word of the unlabeled tweet and the upper limit of the number (4 or 7 or 14) of feature words to be expanded adopted the values that make the F-score of the development set the best. We expanded the core words to the feature words for the training and development set of each product and learned the paragraph vectors with the aforementioned word vectors as the initial vectors. As with parameter tuning necessary for paragraph vector learning, the upper limit of feature words expanded from each core word was also determined using the Spearmin as a parameter. The PV-DM and PV-DBOW in (2) and the PV-DBOW with feature word expansion were combined and used as the features for the construction of SVM classifier.

### 5.4.3 Results

Table 14 shows the F-score and the error rate in the three-class classification. Because the paragraph vector randomly acquires tweets from the benchmark every learning, different feature vectors are generated even with the same condition setting. Therefore, because the classification accuracy of SVM fluctuates, we calculated the average of 5 tests for the development set and the average of 10 tests for the test set. In the test set learning, the case where the word vectors of PV-DBOW were not updated (in the case of the “Test” column of Table 14) and the case where the word vectors of PV-DBOW

were updated (in the case of the “Test (Adapt)” column of Table 14) were evaluated.

In the evaluation of the development set, the proposed method exceeded the paragraph vector (PVEC) (hereinafter referred to as PVEC) by 2.5 points for Product A and 4.8 points for Product B. The relative value of the error rate of Product A: 6.9% ( $= 2.5 / 36.1$ ) and Product B: 14.7% improved in the proposed method. As shown in Table 11 in Section 4.3.1, it is considered that the word expansion by the proposed method was more effective in the improvement of the error rate of Product B because the benchmark of Product B has a higher sparsity of words than that of Product A. In both the PVEC and proposed method the evaluation result of the test set shows that the F-score in the case where the word vectors of PV-DBOW were updated (in the case of the “Test(Adapt)” column of Table 14) was equal or improved as compared with the case where the word vector was not updated (in the case of the “Test” column of Table 14). When updating the word vector of the test set, the F-score of the proposed method exceeded the F-score of the PVEC by 2.0 points of Product A and 3.2 points of Product B. With the relative value of error rate, Product A: 5.8% and Product B: 10.2% were improved by the proposed method. Although the improvement rate of the test set by the proposed method is lower for both products than that for the development set, the rate of decrease for Product B is larger. The reason for this will be discussed in the next section.

Word vector update of the test set improved by 1 point of the F-score (the relative value of the error rate 3.4%), which was the most effective, in the proposed method of Product B, compared with the case where the word vector was not updated. The reason is that Product B has a large effect of the word expansion, so it increases to the F-score of 72.1 in the development set and is considered to be over-learning. Because feature words are limited to 266, we confirmed that over-learning problems could be solved by adapting to the context information of the test set even with fewer tweets.

Meanwhile, the improvement rate of the PVEC for the baseline BoW that does not hold the word order in Product A was 4.6 points (the relative value of the error rate of 11.3%) for the development set and 2.8 points (the relative value is 7.5%) for the test set. Also, the improvement rate in Product B was 0.6 points (the relative value 1.8%) for the development set and 0.5 points (the relative value 1.6%) for the test set. In Product A, the PVEC significantly improves comparing with the BoW, and the effect of retaining the word order can be confirmed. However in Product B, the PVEC shows



Table 15. Parameters of feature extraction and their values for Product A and Product B.

Parameters	Min-Max	Product A		Product B	
		Proposed Method	PVEC	Proposed Method	PVEC
Training set: # of iterations for learning (learning rate)	5~15 (0.007~0.025)	12 (0.007)	5 (0.024)	15 (0.007)	15 (0.007)
Test set: # of iterations for learning (learning rate)	5~15 (0.007~0.025)	5 (0.018)	15 (0.021)	15 (0.010)	5 (0.014)
Dimensionality of feature vectors	100~400	142	400	400	400
Upper limit of expanded feature words	7~17	17		7	
Expanded feature words on nouns	0:No, 1:Yes	1		0	
Window size of PV-DBOW(Proposed Method)	6~18	6		6	
Hidden layer of PV-DM	sum	sum	sum	sum	sum
Window size of PV-DM	4~12	4	4	12	5
Window size of PV-DBOW	5~15	5	5	5	5
Word occurrence frequency threshold	2	2	2	2	2
Downsample threshold for words	1e-03	1e-03	1e-03	1e-03	1e-03
Negative Sampling	0~10	10	10	10	10

a significant difference to the BoW, but the difference is small. As shown in Table 11 of Section 4.3.1, this evaluation result is considered to be because the benchmark of Product A consisted of 45% more average word number per one tweet than that of Product B, and the tweet’s sentence length is long.

Table 15 shows parameters of the feature extraction, the fluctuation width (Min-Max), and the parameter settings adopted for Product A and Product B, in the proposed method and PVEC. “Word occurrence frequency threshold” indicates a threshold for deleting low-frequency words from the vocabulary, and “Downsample threshold for words” indicates a threshold for probabilistically reducing high-frequency words. For these thresholds, we created a vocabulary when learning unlabeled tweets, and that setting is used. Also, we can not change the unlabeled tweet settings (the sum of the input vectors) as to whether to set the hidden layer of the PV-DM as the sum of the input vectors or as the concatenation. “Negative sampling” is the number of words that sample other words for the surrounding words in a tweet. Regarding parameters other than the fixed value, we examined the allowable range of parameters based on the error rate of the development set to avoid the problem of local optimization in Bayesian optimization. At first, we fixed to the default value (the recommended value of the tool or thesis) in advance and shook each parameter in turn. Based on this preliminary

Table 16. The number of succeeded tweets and failed tweets by the proposed method in Product A.

Answer	Dev. set		Test set (Adapt)	
	Incorrect → Correct	Correct → Incorrect	Incorrect → Correct	Correct → Incorrect
Positive	28	20	31	13
Negative	10	12	13	24
Neutral	41	23	31	17

Table 17. The number of succeeded tweets and failed tweets by the proposed method in Product B.

Answer	Dev. set		Test set (Adapt)	
	Incorrect → Correct	Correct → Incorrect	Incorrect → Correct	Correct → Incorrect
Positive	126	99	82	120
Negative	109	82	112	72
Neutral	125	112	134	90

experiment, the fluctuation width (Min-Max) of each parameter was decided.

#### 5.4.4 Consideration

We examine the effect of the expansion of feature words in the proposed method. Table 16 (Product A) and Table 17 (Product B) show the number of tweets whose PVEC’s incorrect answers were changed to correct answers and the number of tweets whose PVEC’s correct answers were changed to incorrect answers by the proposed method. The proposed method affected sentiment analysis of about 16% in Product A, however, of about 45% in Product B. As shown in Table 11 of Section 4.3.1, because the average word number per a tweet of Product A has more 4.5 words than that of Product B, Product A is highly unlikely to be affected by the expansion of feature words. Also, in Product A, the expansion of feature words in the development and test set shows that it is effective for the classification of positive and neutral classes.

In Product A, feature words with the high frequency of occurrences recalling positive or negative cases and the number of tweets to which they were given were shown below.

- **Positive:** positive:1794, newness:1607, excellent:1489, trend-popularity:929,

brightness:624, beauty:356

- **Negative:** negative:1088, difficult:590, coarseness:411

The fact that the number of tweets given feature words that recall negative cases is lower than that of positive cases is considered to be a small factor of the expansion of feature words in the classification of negative cases. On the other hand, in Product B, all positive, negative and neutral answers were improved in the development set. However, in the test set, the negative and the neutral answers further improved, but the positive answers resulted in the negative effect of the expansion of feature words. We will investigate the reasons. In the positive and negative classification of Product B, typical feature words in the succeeded tweet group (incorrect answer  $\rightarrow$  correct answer) by the proposed method and its typical feature words in the failed tweet group (correct answer  $\rightarrow$  incorrect answer) are shown in the development and test set respectively in Table 18. In each tweet group, the occurrence frequency of each feature word was normalized by the number of tweets of each group, and feature words with high occurrence ratios to the pair group were extracted and arranged in the order of frequency in each group.

In the following, we analyze typical feature words for each pair in Table 18 to investigate the reason why the incorrect answers of the conventional method were changed to the correct answers by the proposed method, and the reason why the correct answers of the conventional method were changed to the incorrect answers by the proposed method.

- **Positive: Dev. set (incorrect  $\rightarrow$  correct):** Typical feature words were the expansion of feature words for words such as “good, beautiful, like, new, recommend, expect, progress” and these feature words contributed to the positive sentiment determination.
- **Positive: Dev. set (correct  $\rightarrow$  incorrect):** Typical feature words were the expansion of feature words for words such as “wastefully, mistakenly, impossibly, lost, scared, leaning.” There were many tweets which are difficult to judge positive sentiment beyond the context, such as the negation form and the topics of other products.
- **Positive: Test set (incorrect  $\rightarrow$  correct):** There were many tweets related to “buy” (typical feature words “quantity, economy, cheap, tax\_system”) and other

Table 18. The typical feature words in the succeeded tweets and the failed tweets by the proposed method in Product B (their occurrence ratio to their pair group).

Answer	Dev. set		Test set (Adapt)	
	Incorrect → Correct	Correct → Incorrect	Incorrect → Correct	Correct → Incorrect
Positive	emotion(2.0)	power-degree(2.7)	positive(2.2)	machine(4.4)
	positive(2.2)	negative(2.4)	quantity(2.1)	daily_necessities(3.3)
	emotional(2.6)		economy(2.3)	tool(3.1)
	morality-ethics(3.6)		cheap(2.4)	facility(5.0)
	power(2.5)		tax_system(2.4)	negative(3.2)
	trend-popularity(2.1)		trend-popularity(3.9)	complicated(4.4)
Negative	negative(1.6)	quantity(2.9)	negative(6.3)	change(2.2)
	machine(2.0)	positive(3.7)	property(3.1)	newness(14)
	facility(2.0)	economy(3.0)	order-regularity(2.6)	positive(1.6)
	coarseness(2.4)	cheap(3.0)	coarseness(4.8)	
	movement(2.3)	tax_system(3.0)		
	disease(2.3)	morality-ethics(4.0)		

typical feature words were the expansion of feature words for words such as “beautiful, secure, fashionable, convenient, enviable.” The tendency of tweets was similar to the development set (incorrect → correct).

- **Positive: Test set (correct -> incorrect):** There were a lot of tweets related to “use” (typical feature words “machine, daily\_necessities, tools, facilities”) and other typical feature words were the expansion of feature words for words such as “difficult, dislike, inconvenient, sorry” and so on. The tendency of tweets was similar to the development set (correct → incorrect).
- **Negative: Dev. set (incorrect -> correct):** There were a lot of tweets related to “use.” Another typical feature word “disease” was the expansion of feature words for words such as “dangerous, severe, upset, abnormal, weak.” Also, “movement” was the expansion of feature words for words such as “split, repeat, rapid” and so on. These feature words contributed to the negative sentiment determination.
- **Negative: Dev. set (correct -> incorrect):** There were many tweets related to “buy” and “positive, morality-ethics” was the expansion of feature words for

words “good.” There were many tweets which are difficult to judge positive sentiment beyond the context, such as the negative form and the topics of other products.

- **Negative: Test set (incorrect -> correct):** There were many tweets related to defects, and the typical feature words “negative, order-regularity, property” were the expansion of feature words for word “selfish,” and these feature words contributed to the sentiment determination with negative.
- **Negative: Test set (correct -> incorrect):** There were many tweets related to model change, typical feature words were the expansion of feature words for words such as “change, new,” the tendency of tweets was similar to the development set (correct → incorrect).

As a result of the analysis, the sentiment error of PVEC could be correctly judged by the proposed method in the four cases of “satisfied,” “buy,” “use,” and “defect,” regardless of the development set or the test set or the positive answers or the negative answers. In any case, it was found that the sentiment judgment succeeded by supplementing the context information of the tweet by the expansion of feature words.

On the other hand, the correct sentiment judgment of PVEC could be judged wrongly by the proposed method in the four cases of “unsatisfied,” “use,” “buy,” and “change from.” In any case, because they were negative forms of words influencing sentiment or topics of other products, it turned out that many cases were difficult to cope with the PV-DBOW which does not take the word order into consideration. Compared with the PVEC, this is considered to be due to the relatively weakened effect of the PV-DM in the proposed method.

#### **5.4.5 On the effectiveness of the proposed method on a benchmark with diversity even if it is minimal**

In this chapter, we showed the effectiveness of the proposed method using the benchmark of Product B consisting of about 12 thousand tweets and the benchmark of Product A whose total number of tweets is less than half that of Product B but whose average word number per tweet is about 45% more. However, if the number of products is two, the evaluation is not sufficient. Therefore, for a variety of products, we created a small experimental benchmark to evaluate the effectiveness of the proposed method. The

Table 19. Configuration of the experimental benchmark.

Product	Training set			Test set		
	Positive	Negative	Neutral	Positive	Negative	Neutral
Smartphone						
Product A	7	9	8	2	2	2
Product B	5	5	6	2	2	2
Product C	1	5	2	0	1	0
Others	0	0	9	0	0	3
Robot cleaner						
Robot A	6	2	1	1	1	1
Robot B	4	3	3	2	0	0
Convenience store printing service						
Service A	2	2	1	1	1	0
Service B	1	5	2	0	1	0
Maker						
Maker A	8	5	1	2	2	0
Maker B	3	3	2	1	1	0
Total	37	39	35	11	11	8

configuration of the experimental benchmark is shown in Table 19. Based on smartphone, robot cleaner, convenience store printing service, manufacturer’s classification, each classification consists of 2 to 4 categories. "Others" of the smartphone consists of the tweets that refer to multiple smartphone product brands, including Products A to C, all being neutral labeled. Maker consists of tweets about the reputation regarding Maker A and B and the products of both companies. The entire benchmark consists of 141 labeled tweets (111 tweets for the training set and 30 tweets for the test set).

For the evaluation, the initial value of the word vector was set as a random value without using unlabeled tweets. The evaluation was carried out on the proposed method learned with the expansion of feature words for the core words in the tweets using the PV-DBOW model and the conventional method learned using only the PV-DBOW model for tweets. In other words, using only the vectors of PV-DBOW (proposed method) and PV-DBOW (conventional method) as the feature extraction with the aim of evaluating the learning effect of the expansion of feature words by the proposed method, we constructed an SVM classifier with leave-one-out cross-validation for the training set. Table 20 shows the parameter settings for the feature extraction determined based on the training set. According to the setting of “Downsample threshold for words” there was no reduction of high-frequency words in the conventional method, almost all

Table 20. Parameters of feature extraction and their values for the proposed method and the conventional method.

Parameters	Proposed Method	Conventional Method
Training set: number of iterations for learning	6000	600
Test set: number of iterations for learning	1500	300
Dimensionality of feature vectors	100	100
Upper limit of expanded feature words	7	
Expanded feature words on nouns	1	
Window size	30	15
Word occurrence frequency threshold	2	2
Downsample threshold for words	1e-06	0.1
Negative Sampling	10	10

Table 21. Evaluation results2: F scores(SD) in the three class and the 2-class classifications by PV-DBOW (the conventional method) and PV-DBOW (the proposed method).

	3-class classification F-score(SD)	2-class classification F-score(SD)
Conventional Method	50.3( $\pm$ 3.0)	64.6( $\pm$ 3.7)
Proposed Method	60.1*( $\pm$ 3.3)	73.8**( $\pm$ 4.7)
vs.Conventional Method *p=3.4e-17 < 0.05		**p=2.2e-11 < 0.05

words were subject to reduction in the proposed method. For this reason, although the number of iterations for learning in the proposed method was very large, it means that the combination of different words was learned every time.

The evaluation results are shown in Table 21. In addition to the F-score of the 3-class classifications in the evaluation result, the F-score of the 2-class classifications was also evaluated. Because the neutral of the test set is biased towards the smartphone, the evaluation was done by the training set and test set with only 2-class of positive and negative. The proposed method outperformed the conventional method by 9 to 10 points in both the 3-class classifications F-score and the 2-class classifications F-score. Because the standard deviation was large, the average of 30 trials was adopted. An example of a tweet that was incorrect sentiment judgment in the conventional method and correct sentiment judgment in the proposed method is shown below. An example of a tweet that was wrong sentiment analysis in the conventional method and correct

sentiment analysis in the proposed method is shown below.

- Network print service is very *convenient*<sub>1</sub>. [state·aspect, positive, easy, kindness, customer]<sub>1</sub>

Like the analysis of Product B shown in the previous section, the expansion of feature words contributed to the positive sentiment judgment. Also, an example in which the sentiment analysis failed in the proposed method is shown below.

- I printed a *trial*<sub>1</sub> with convenience store print service A, but I do not get *satisfactory*<sub>2</sub> *green*<sub>3</sub> ... [power-degree, worth, accurate, action, behavior, reaction, material]<sub>1</sub> [state·aspect, positive, rational, idea]<sub>2</sub> [plant, traffic-transportation, environment, positive, color, green, agriculture]<sub>3</sub>

The negative form of this tweet is inverted in polarity, and as you can see in the analysis in the previous section, the PV-DBOW (the proposed method) alone can not cope with the negative form. We confirmed that this tweet is judged as negative by concatenating with the PV-DM.

## 5.5 Conclusion

In this chapter, we proposed an integration method to learn feature words expanded by using the word semantic vector dictionary with the paragraph vector models to solve the problem of word sparsity in Twitter.

We first proposed a reputation information extraction method used for product development and quality support.

Next, an experimental system for extracting reputation information using a BoW as a baseline, the paragraph vector, and the proposed method was constructed and evaluated. As a result, the proposed method, the expansion of feature words, was superior to the paragraph vector as the word sparsity was high. We confirmed that the problem of over learning on the development set of the benchmark having word sparsity is resolved when adapting the word vector to the test set. For the BoW not considering the word order, it showed that the paragraph vector has superiority as the sentence length is longer.

Also, the proposed method showed that it could perform error analysis using expanded feature words.



Moreover, we showed the effectiveness of the proposed method using a small experimental benchmark consisting of various products.

Finally, we will discuss the limitation and solution of the proposed method.

- The following two points were clarified as a result of error analysis using expanded feature words.
  - The PV-DBOW (the proposed method) alone can not cope with the negative form and complicated context such as topics of other products.
  - The effect of the PV-DM considering the word order is relatively weakened in the case where the word sparsity is high.

As a result of the error analysis of Product B in Section 5.4.4, it is confirmed that the F-score is improved to 72.1 by concatenating two PV-DMs in the proposed method. In the future, it is conceivable that weights of PV-DM, PV-DBOW, PV-DBOW (the proposed method) can be adjusted as parameters depending on the benchmark.

- Because the meaning of words varies depending on the context, it is conceivable that the core words to be expanded to feature words changes in the context. However, core words with multiple meanings are given feature words corresponding to all meanings, so core words can not be selected according to the context. On the other hand, because feature words are classified under six large classifications as shown in Table 3 in Chapter 3, there is a possibility of selecting a large classification of feature words to be expanded according to the context. In this experiment, feature words are selected up to the upper limit of expanded feature words in order from the top of the classification. In the future, it is considered not to select from the top of the classification but to select from the large classification suitable for the current context in order.



## **6. Semantically Readable Distributed Representation Learning and Its Expandability**

### **6.1 Introduction**

The problem in engineering with the distributed representations of words and paragraphs is that the meaning of the distributed representations is difficult to understand. Thus, tests and improvement in the qualities of natural language processing applications (e.g., sentiment analysis) are necessary because the distributed representations are not readable as is.

In this chapter, we propose a new method of automatically learning readable distributed representations using the word2vec and paragraph vector models based on the word semantic vector dictionary [25]. Word semantic vector dictionaries are more like distributed representations rather than semantic lexicons like WordNet [41] and FrameNet [2] because each core word is defined as a fixed-length dense vector. In this chapter, the 266 feature words are taken as the hidden nodes of each model. Then, the initial weights between the input word and each hidden node, which is a seed vector, are given based on the dictionary. We investigated whether or not the meaning of each hidden node is maintained, even after learning is done by the neural networks related to core words. Also, the meaning of the hidden nodes is maintained to some extent in new words. If the learning is insufficient, then the weights after learning will naturally correlate with the initial weights. However, the accuracy of sentiment analysis using distributed representations after learning by the proposed method is better than the paragraph vector performance in a single domain benchmark. We present our evaluation of the readability of document embedding in a user test conducted through crowdsourcing. A total of 66.1% and 52.4% of the given feature words were related to tweets where one of the paragraph vector models learned the document embedding for the top weighted hidden node and the top five weighted hidden nodes, respectively.

We demonstrated the readability of the document embedding in a single domain benchmark. However, testing the reproducibility and expandability of the proposed method is important. We improved the dictionary and a diverse Japanese sentiment analysis benchmark for this purpose. The dictionary consists of 264 feature words—not 266—and 20,330 core words. The diverse benchmark consists of 38,576 tweets labeled

positive or negative for eight categories as shown in Section 4.3.2. The evaluation results show that the performance of sentiment analysis is better than or comparable to the conventional method’s while maintaining the correlation between the word vectors learned with 3 million tweets and the seed vectors based on the dictionary. Moreover, to evaluate the domain-independence of the proposed method using the Wikipedia corpus and a Japanese word similarity dataset, we found that synonyms have similar vectors while the top five weighted feature words of the core word after learning are related to the core word. Therefore, our method improves the readability of the distributed representations, which are the weights of each hidden node for words and paragraphs. Also, it can be applied to mining of social media such as Twitter by using each feature word as a conceptual axis.

## 6.2 Related Work

Research on the integration of external semantic lexicons and distributed representation learning has been active. The following three research directions are being studied.

- **Pre-processing.** Feature word expansion on Twitter as proposed in Chapter 5 belongs to this direction [27]. Tweet2vec [60] trained the CNN-LSTM encoder-decoder model on 3 million randomly selected tweets populated using data augmentation techniques, which are useful for controlling generalization error for deep learning model. Data augmentation techniques refer to replicating tweets and replacing some words with their synonyms using WordNet [41].
- **Learning process.** RC-NET [62] is built upon the Skip-gram model [39] whose objective function is extended by incorporating both the relational knowledge (like is-a, etc) and the categorical knowledge (like synonyms) as regularization functions. Bollegala et al. proposed a global word co-occurrence prediction method [46] using the semantic relations in WordNet as a regularizer [6]. Our proposal in this chapter belongs to this direction.
- **Post-processing.** Retrofitting [13] is a technique for fitting learned word vectors to semantic lexicons. In this chapter, we used this technique to create initial weights of core words.

Experiments have shown that the precision of distributed representations of words has qualitatively improved the best in [6] and that the accuracy of sentiment analysis has improved in [60]. Both showed that they were state-of-the-arts using standard datasets on word similarity and sentiment analysis. Similar studies have been done using topic models based on LDA [3]. Topical Word Embeddings(TWE) [36], in which topical word refers to a word taking a specific topic, have been proposed to measure contextual word similarity by extending the Skip-gram model [39]. And TWE outperformed the Skip-gram model in word similarity tasks. TWE is also applied to tweet topic classification tasks and performs better than paragraph vectors [47]. However, no reports on the relevant literature describe an attempt to give meaning to each hidden node.

One model of paragraph vectors (PV-DBOW) [34] uses pre-trained word embeddings that reportedly improve task performance [33]. Although this thesis shows the possibility of learning proper document embeddings with good initialization of word embeddings, it does not demonstrate the possibility of interpretation of hidden nodes.

Proposals have been made to give meaning to each dimension of word representations obtained from random initial values like word2vec without external semantic lexicons. First, in the matrix factorization derived from not random initial values but corpus co-occurrence patterns, the interpretability of each dimension has been proposed to increase by introducing sparsity and nonnegativity into the vector representation of words, denoted as non-negative sparse embedding (NNSE) [43]. For word2vec, a proposal has first been made to enhance the interpretability of dimensions by online learning of the Skip-Gram model while keeping the word embeddings non-negative, denoted as the online interpretable word embeddings-improved projected gradient (OIWE-IPG) [37], which was influenced by NNSE. Also, the Sparse CBOW model has been proposed to learn sparse word representations by introducing the sparse constraint [58]. The interpretability of each dimension is tested using a word intrusion task [43]. The dataset consists of the top 5 weighted words of each dimension and one other word with noise from the lower half of that dimension, which is ranked higher in other dimensions. The interpretability is evaluated using the detection precision of the intruders by humans [43, 37]. A method for automatically evaluating the interpretability without the intervention of people has also been proposed [58]. An objective evaluation is difficult because no common dataset and evaluation measure

exist, but both the proposals using word2vec are compared with NNSE in the word intrusion task. These results show that OIWE-IPG is better than NNSE and that NNSE is better than Sparse CBOW. OIWE-IPG has the best performance on a word similarity task in the case of 300 dimensions, and it is superior to Skip-gram, RNN, and NNSE. The differences and advantages of our method are shown in the following for these methods of interpreting the meaning of each dimension of representations using the upper weighted group of words.

- For example, the Wikipedia Corpus has over 500,000 word types. In these methods, people need to analogize the meaning of each dimension from upper weighted words by sorting weights of more than 500,000 word types in each dimension. In our method, the meaning of each dimension is evident because it is represented by 266 classified conceptual classifications.
- In these methods, the meaning of each dimension changes for each corpus to be learned. However, our method is common to each corpus.
- In the word similarity task, these methods are superior to the conventional method, but they are not evaluated by natural language processing applications (e.g., sentiment analysis). Our method can interpret the meaning of each dimension of not only word representations but also document representations while giving performance comparable to the conventional method in the word similarity task also in the natural language processing application.

## 6.3 Proposed Method

### 6.3.1 Hypothesis

In this section, we present a test of the hypothesis that the meaning of each hidden node is maintained using three approaches even after learning word vectors: ①assigning specific meaning to each hidden node, ②giving the strength of the semantic and associative relationship with each hidden node as the initial weights of important words, and ③using word2vec and paragraph vector models.

We will explain why we think the meaning of the hidden layer is maintained, referring to Figure 14. First, we used the multilayer classification of 266 feature words as hidden nodes to cover the content of the encyclopedia as shown in Table 3 in Chapter 3.

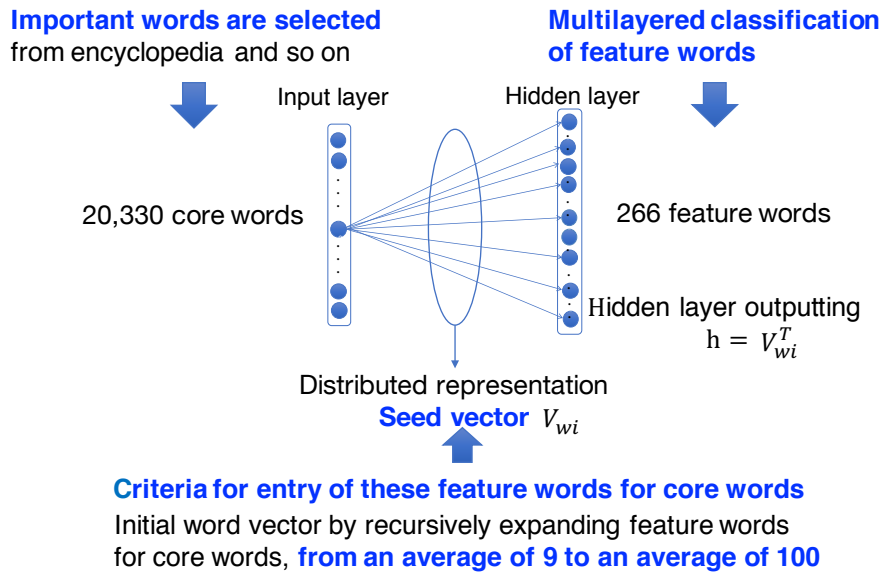


Figure 14. Why do we think that the meaning of the hidden layer is maintained?

Second, 20,330 important words were extracted using frequency analysis of Japanese newspapers and encyclopedia texts as core words. We clarified the criteria of giving feature words to core words based on logical relevance as in Table 4 and associative relevance as in Table 5 in Chapter 3. We then generated seed vectors, which are distributed representations between core words and 266 feature words, by recursively extending the word semantic vector dictionary using the retrofitting algorithms discussed in the next section. As a result, the number of feature words given to core words increased from 9 words on average to 100. In the seed vector, numerical values were entered in dimensions corresponding to these feature words, and values corresponding to other dimensions (average of 166 dimensions) were zero. Third, in the skip-gram model of word2vec, the seed vector of the target word became the output vector of the hidden layer, and a vector obtained by multiplying the output vector of the hidden layer by a fixed number was added or subtracted to the seed vector by backpropagation. Therefore, the weights of the seed vector for the core word were adjusted to predict the context words, but only the weights with the given and retrofitted feature words were affected.

The neural network learns the concept automatically for the hidden layer. Thus, the weights may adapt to the context with the concept maintained by pre-setting the

appropriate conceptual classification to be learned to the nodes of the hidden layer and by giving the suitable initial value.

### 6.3.2 Model Setting for Testing the hypothesis

In this section, we describe the setting to encode the initial weights of the core words based on the strength of the relationship with each feature word, and we describe our test of the hypothesis using Skip-gram as an example.

A method has been proposed for generating a word vector by recursively expanding a definition sentence for a word in a dictionary [59]. The word semantic vector dictionary can be regarded as defining a core word with 266 types of feature words. Because feature words are also core words, recursive extension is necessary. However, convergence occurs when the feature word is expanded several times because the definition sentence of the core word is limited to 266 words. Also, a method has been proposed for retrofitting word vectors according to related words in a dictionary [13]. In that method, we generate a seed vector of the core word by recursively expanding the dictionary using retrofitting tools<sup>12</sup>.

When building a vocabulary from the corpus, the initial vectors of the following two kinds are created first.

- The 266 feature words are added to the vocabulary as one-hot vectors with dimensions corresponding to respective feature words set to 1.
- Other initial word vectors including core words extracted from the corpus are 266-dimensional zero vectors.

The retrofitting algorithm aims at bringing word vectors closer to the relationship of the word entries of the lexicon as post-processing of learning of word vectors [13]. We applied this algorithm for retrofitting the aforementioned initial word vectors, which are 266-dimensional one-hot or zero vectors, into the word semantic vector dictionary. The retrofitting algorithm is shown as the following online update [13]:

$$\mathbf{q}_i = \frac{\sum_{j:(i,j) \in E} \beta_{ij} \mathbf{q}_j + \alpha_i \hat{\mathbf{q}}_i}{\sum_{j:(i,j) \in E} \beta_{ij} + \alpha_i} \quad (12)$$

---

<sup>12</sup><https://github.com/mfaruqui/retrofitting>



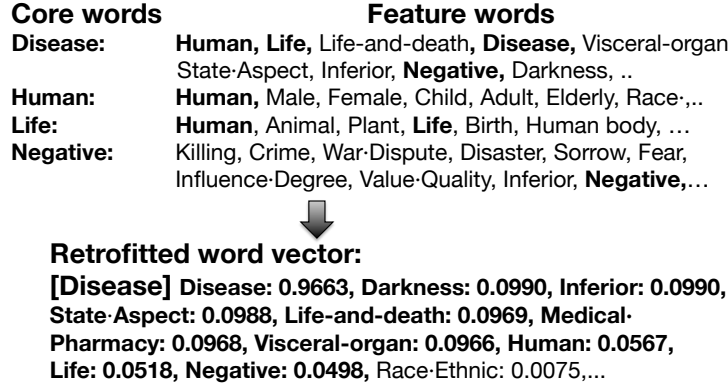


Figure 15. Example of retrofitting “Disease.”

$\mathbf{q}_i$  is the retrofitted word vector for the core word  $w_i$ ,  $\hat{\mathbf{q}}_i$  is the aforementioned initial vector for  $w_i$ , and  $\alpha_i$  is the weight of the initial vector; currently it is set to the number of given feature words  $w_j$  for  $w_i$ .  $\mathbf{q}_j$  is the retrofitted word vector for the given feature word  $w_j$ , and  $\beta_{ij}$  is the weight of the given feature word  $w_j$  for the core word  $w_i$ ; currently, the weight  $\beta_{ij}$  is set to 1. Eq. 12 multiplies the initial vector  $\hat{\mathbf{q}}_i$  of the core word  $w_i$  by the weight  $\alpha_i$ , adding the vectors obtained by multiplying the retrofitted vector  $\mathbf{q}_j$  of the given feature word  $w_j$  by the weight  $\beta_{ij}$  and by dividing it by the sum of both weights. Running the procedure for about ten iterations increases the relationship between each core word and 266 feature words from an average of 9 to an average of 100. The relationship is increased for each of the core words to expand the feature words given to the core word recursively.

Figure 15 presents an example of retrofitting “Disease,” which is a feature word and core word in the dictionary. The points of this algorithm are the following.

- The retrofitted word vector is close to the original vector. In the case of “Disease,” the original vector is a one-hot vector.
- When the feature words assigned to a retrofitted core word are not expanded as core words, the weights of the feature words are almost equal. When expanded, the weights decrease according to the number of feature words to be expanded.

In the vocabulary, each word has two vectors. One is an input vector, which is the weights between the input node and each hidden node, and the other is an output vector,

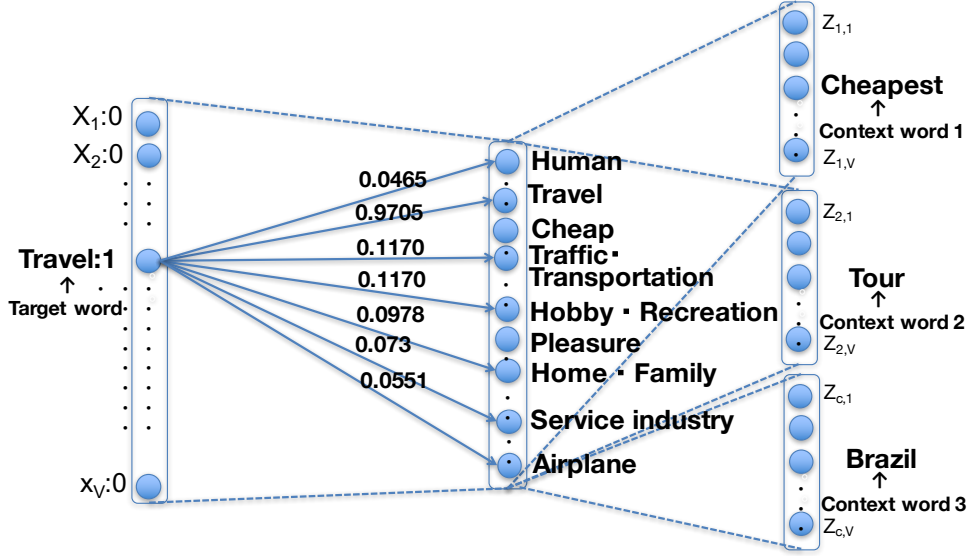


Figure 16. Skip-gram model setting for testing.

which is the weight between each hidden node and the output node. The retrofitted word vector was used as the seed vector of the input vector. The initial weights of the output vector were set to 0, which is the default setting of gensim's doc2vec library<sup>13</sup>.

Figure 16 presents an example of the Skip-gram setting for testing the hypothesis. The input layer specifies the target word. The output layer consists of three context words appearing around the target word. The hidden layer comprises the nodes corresponding to 266 feature words. The weights of the target word for each hidden node are retrofitted weights. Each weight is updated by back propagation so that the probability of predicting the context words increases when the target word is input. The objective function is the following [39, 48].

$$E = -\log \sigma(\mathbf{v}'_{\mathbf{w}}^T \mathbf{h}) - \sum_{w_j \in W_{neg}} \log \sigma(-\mathbf{v}'_{w_j}^T \mathbf{h}) \quad (13)$$

The hidden layer outputting  $\mathbf{h}$  is  $\mathbf{v}_{w_i}^T$ .  $\mathbf{v}_{\mathbf{w}}$  is the input vector whose initial vector is generated by Eq. 12, and  $\mathbf{v}'_{\mathbf{w}}$  is the output vector of the word  $w$ .  $W_{neg}$  is the set of words for negative sampling. The output vector is updated as follows [48].

$$\mathbf{v}'_{w_j}^{(new)} = \mathbf{v}'_{w_j}^{(old)} - \eta \left( \sigma(\mathbf{v}'_{w_j}^{(old)T} \mathbf{h}) - t_j \right) \mathbf{h} \quad (14)$$

<sup>13</sup><https://radimrehurek.com/gensim/models/doc2vec.html>

Table 22. Hyper-parameter settings for learning word vectors.

Hyper-parameters	Values
Dimensionality of the feature vectors	266
Number of iterations over the corpus	20
Learning rate	Initial:0.025, Minimum:0.0001
Window size	5
Downsample threshold for words	1e-5
Number of negative sampling words	15

Table 23. Example of retrofitted and learned word vectors for a core word that is a feature word itself.

Generation method	feature words and weights arranged in descending order
Retrofitted vector for “ <b>travel</b> ”	<b>travel</b> :0.97, traffic·transportation:0.12, hobby·recreation:0.1, home·family:0.1, service industry:0.1, airplane:0.06, human:0.05, car:0.05, overseas:0.05, Japan:0.05,
learned vector for “ <b>travel</b> ” by PV-DM	<b>travel</b> :1.41, <b>machine</b> :0.65, <b>image</b> :0.61, company:0.55, state·aspect:0.52, traffic·transportation:0.5, hobby·recreation:0.43, education:0.40, facility:0.38, behavior:0.36,
learned vector for “ <b>travel</b> ” by PV-DBOW	<b>travel</b> :1.45, time:0.45, custom:0.44, clothes:0.43, state·aspect:0.43, Europe:0.42, low:0.42, <b>image</b> :0.41, public system:0.40, <b>machine</b> :0.40,

where  $t_j$  is 1 when  $w_j$  is the context word and 0 otherwise. The initial output vector  $\mathbf{v}'_{\mathbf{w}}$  is 0. Thus, the output vectors of the context words become close to the input vector, which is the seed vector, of the target word.

## 6.4 Experiments

In these experiments, we examined the relationship between sentiment analysis using a single domain benchmark and readability of tweet embedding in a user test. We also tested the hypothesis on whether or not weights obtained based on learning and weights based on the dictionary are correlated in a closed test and an open test, compared with a control test.

Table 24. Example of retrofitted and learned word vectors for a core word that is not a feature word.

Generation method	feature words and weights arranged in descending order
Retrofitted vector for “screen”	<b>computer</b> :0.42, machine:0.42, communication tech.:0.42, mass media:0.42, electronics:0.40, <b>plane</b> :0.22, <b>color</b> :0.22, communication:0.22, advertisement:0.04,
learned vector for “screen” by PV-DM	<b>computer</b> :0.83, machine:0.78, <b>image</b> :0.78, state-aspect:0.73, company:0.68, <b>plane</b> :0.61, education:0.55, book:0.55, facility:0.52, <b>color</b> :0.50,
learned vector for “screen” by PV-DBOW	state-aspect:0.68, <b>plane</b> :0.50, <b>computer</b> :0.45, <b>image</b> :0.34, human body:0.32, relationship:0.31, chemistry:0.31, civil engineering-architecture:0.29, tool:0.26, <b>color</b> :0.26,

We used the evaluation benchmark of sentiment analysis for Product B and the 560,853 unlabeled tweets as shown in Table 10. in Section 4.3.1. For the 560,853 unlabeled tweets, only noises such as the URL and the account name were deleted. Also, MeCab and its dictionary mecab-ipadic-NEologd were used to extract words from tweets. The number of words extracted from the corpus five or more times was 30,468, while the number of retrofitted core words was 6,814 words.

#### 6.4.1 Learning Word Vectors by Our Method and Evaluation of Correlation Coefficients

First, the word vector was updated using two variants of paragraph vector models with unlabeled tweets only using gensim’s doc2vec library. On the basis of the accuracy of the sentiment analysis of the final stage, we decided the values of hyper-parameters for paragraph vector learning of the conventional method. The hyper-parameter settings for learning the corpus are shown in Table 22. Our method used the same hyper-parameter settings. Here, the size of the feature vectors was adjusted to the number of feature words, 266. When the number of dimensions of the feature vectors exceeded 266, we could set the initial value 0 or the random number for the part exceeding 266 in our method. However, no difference occurred in accuracy between 266 dimensions and 300 dimensions for the corpus in the paragraph vector of the conventional method.

Table 25. Evaluation results 1: Correlation coefficients between initial and learned word vectors.

	Control Test	Closed Test	Open Test
PV-DM/CBOW	0.224	0.608	0.340
PV-DBOW/Skip-gram	0.211	0.642	0.395

Thus, we utilized 266 dimensions. Both the PV-DM and PV-DBOW have the same hyper-parameter settings. We used the sum of the input vectors for the hidden layer of the PV-DM for the same reason as with the hyper-parameter settings.

Table 23 shows an example of retrofitted and learned word vectors for a core word “travel,” which is a feature word itself. The retrofitted word vector for “travel” was similar to a one-hot vector. The weight of the feature word “travel” of the learned word vector “travel” was more than twice the weight of other feature words in the PV-DM and more than three times the weight of other feature words in the PV-DBOW. Because our method learned the word vectors with a smartphone corpus, “machine” and “image” had higher weights in both of the learned word vectors. Table 24 shows an example of retrofitted and learned word vectors for a core word “screen,” which is not a feature word. Feature words with the top eight weights of retrofitted vectors for “screen” are those given to the core word “screen.” Of these, the feature word “computer,” “plane” and “color” had higher weights in both of the learned word vectors. Although “image” is not a feature word assigned to the core word “screen,” it gained a high score in both learning methods with the smartphone corpus.

Table 25 presents correlation coefficients between retrofitted vectors and learned vectors in the closed and open test, compared with those of the control test. The control test shows the correlation between the word vectors after learning by the conventional method and the initial vectors, the closed test shows the correlation for the core words used for learning by the proposed method, and the open test shows the correlation for the core words not used for learning by the proposed method as follows.

**Control test:** We selected the core words (814 words) in the top 2% high-frequency words (2343 words) for the evaluation because high-frequency words had a stronger influence on tweet vector learning than low-frequency words. We evaluated the correlation coefficients between the initial vectors as a control test using default random

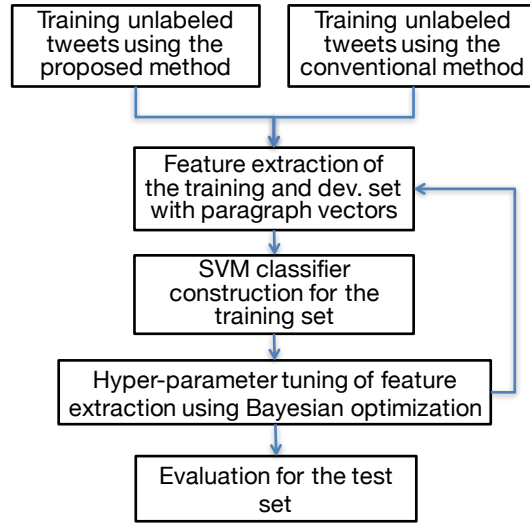


Figure 17. Procedures of sentiment analysis.

initialization and the learned vectors.

**Closed test:** For the 814 core words, we combined all elements of retrofitted word vectors with 0.013 or more as one vector and similarly learned word vectors. A feature word with a value of 0.013 or less corresponded to a relationship according to a two-step recursion with the core word. Therefore, we excluded feature words having a value less than 0.013 from the calculation of the correlation coefficient because the relationship with the core word is not high. Then, we calculated the correlation coefficients of the two vectors. The results showed a stronger correlation compared with that of the control test.

**Open test:** Let the word vectors learn for the unlabeled tweets excluding the aforementioned 814 retrofitted core word vectors. Subsequently, we calculated the correlation coefficient between the aforementioned 814 retrofitted core word vectors with 0.013 or more and the corresponding 814 learned word vectors. The results showed a weak correlation.

#### 6.4.2 Evaluation of Sentiment Analysis

Second, we evaluated the tasks of sentiment analysis using the paragraph vector of the conventional method and our method. The experimental procedures are presented

Table 26. Evaluation results 2: Macro-average F-score for predicting positive and negative tweets in 3-class sentiment analysis.

	Dev. Set	Test Set
Conventional Method	68.6	68.8
Our Method	70.5*	70.2**
vs. Conventional Method *p=0.0006<0.05    **p=1.9e-05<0.05		

in Figure 17. The only difference between the methods was the initial weights when learning the unlabeled tweets. Specifically, the evaluation steps were the following.

**[Step 1]** In our method, we updated word vectors by having it learn unlabeled tweets with the PV-DM and PV-DBOW based on the retrofitted word vectors, as described in the previous section. In the conventional method, we used the same settings of the hyper-parameters shown in Table 22 for the PV-DM and PV-DBOW based on standard random initialization.

**[Step 2]** In learning the paragraph vector of the training and dev. set, let the word vectors learned in Step 1 be the initial value of the word vectors. We combined the PV-DBOW and PV-DM for each tweet and created a tweet feature vector. We built a tweet classifier with a support vector machine (SVM) using the labels of each training tweet as training data.

**[Step 3]** Each tweet vector and its label of the development set was entered into the classifier, and the error rate  $[=100-(F_{pos}+F_{neg})/2]$  was measured. Bayesian optimization automatically adjusted the parameters of the paragraph vector learning so that the output of the objective function, which is the error rate, was minimized [55]. After Step 2 and Step 3 were repeated until the error rate converged, the hyper-parameter of the paragraph vector learning was determined.

As presented in Table 26, we found that the evaluation results of our method were better than those of the conventional method in the macro-average F-score of positive and negative prediction in three-class classification.

Title: Please rate the ten words given to the tweet with ○ ×. There are 15 tweets.  
Tweets that are evaluating smartphones and ten words are given. If these words represent contents of tweets, please add ○, if they are unlikely to be related, please add ×. An example of work is as follows.

Tweet: “Indeed, it is the era of product B.”

Word:	User rating
Trends·Popularity:	○
Positive:	○
Education:	×
Past:	×
Future:	○

In this example, ○ is entered in “Trends·Popularity, Pojitive, Future” from associative relevance with “era of ~.” Please enter ○, × in a sense like an associative game. Also, if a word that exactly matches the theme of the tweet is displayed like the following tweet and word combination, please add ◎.

Tweets: “Because we are managing music with software A, I listen to it by product B...”

Word:	User rating
Music:	◎

▽ Approval condition: In the selection of ○ and ×, it is not judged that they were given randomly like all ○ or all ×.

Figure 18. Example of a task by crowdsourcing.

### 6.4.3 User Test for Readability

Finally, we conducted a readability user test of hidden nodes for the learned tweet vectors. We prepared feature words for hidden nodes with top ten weights for 30 tweets, which were estimated to be positive or negative by using the PV-DM and PV-DBOW individually. The top ten feature words were given for each tweet, and 30 user testers were asked through crowdsourcing whether or not each tweet was associated with the ten feature words. An example of a task by crowdsourcing is shown in the figure 18. One task consists of 15 tweets. Each user tester can perform any of the tasks but cannot do the same task twice. This user test can absorb individual differences by letting each user tester answer whether or not each tweet is related to each top ten feature word and by taking the average of 30 user testers.

Table 27 shows what percentage of the Top 1, 5, and 10 weighted hidden nodes of PV-DBOW or PV-DM tweet vectors that were classified as either positive or negative were related to the tweets. For the PV-DBOW, the percentage of the given feature words were related to the tweets where one of the paragraph vector models learned



Table 27. Evaluation results 3: Readability of hidden nodes and macro-average F-score of the corresponding 2-class sentiment analysis.

Tweet Vectors	Readability of hidden nodes			Sentiment Analysis
	Top1	Top5	Top10	F-score
PV-DBOW Positive	65.5%	56.1%	46.6%	86.8
PV-DBOW Negative	66.8%	48.7%	41.5%	78.3
PV-DBOW All	66.1%	52.4%	44.1%	82.5
PV-DM Positive	56.4%	46.2%	36.9%	80.2
PV-DM Negative	67.3%	43.8%	37.8%	74.7
PV-DM All	61.9%	45.0%	37.3%	77.5
Control Test for Positive		16.1%	15.4%	80.2
Control Test for Negative		13.9%	13.9%	74.7
Control Test for All		15.0%	14.6%	77.5

the document embedding was 66.1% for the top weighted hidden node and was 52.4% for the top five weighted hidden nodes. The PV-DM results were 61.9% for the top weighted hidden nodes and 45.0% for the top five weighted hidden nodes. Overall, the PV-DBOW readability was better than that of PV-DM, though the PV-DM results tended to be more readable for negative tweets compared with those of the PV-DBOW. The third results show a control test. The control test assessed how user testers scored on five or ten feature words randomly chosen for the tweets classified as either positive or negative using PV-DM because no conventional methods give the meaning of features of tweet embeddings. A comparison with the control test showed that our method apparently improves the readability of distributed representation learning.

Table 27 also shows the macro-average F-score in 2-class sentiment analysis using the corresponding positive and negative vectors for reference. A common trend was evident in the readability of the hidden nodes and sentiment analysis.

For the five cases of the paragraph vector (PV-DBOW) in the aforementioned readability user test, each tweet and a list in descending order of the weight of the feature words are shown below.

**Product B is amazing!** [power, strong, human, worth, state-aspect, facility, education, positive,]

**Product B is a godsend!** [state-aspect, thought, human, relationship, life and death, existence, power, strong,]

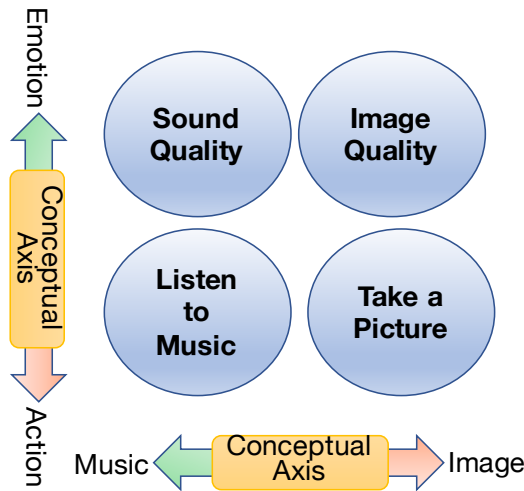


Figure 19. Application image of social media filtering using the conceptual axis.

**Oh, after all, the sound of Product B is something good and deep.** [sound, advertisement, state·aspect, image, **emotion**, worth, **music**, power, sense,]

**While listening to the music with Product B, I was surprised with how good the sound quality was.** [state·aspect, **music**, facility, action, ethics, service industry, **emotion**, quantity,]

**Even if Product B is fully charged, the LED remains lit.** [brightness, machine, **luminescence**, state·aspect, computer, activity, essence,]

For the first two similar tweets, the common feature words “power,” “strong,” and “human” had higher weights. Other tweets with the PV-DBOW tweet vectors that were classified as positive and the three feature words with the top ten weights were as follows.

**"Product B is the best." "Product B is the most attractive."**

In the following two tweets on the sound quality of smartphones, the common feature words “music” and “emotion” had higher weights. In the last tweet on charging and LED lighting, the feature words “brightness” and “luminescence” had higher weights. Also, importantly, clear differences emerged in the top feature words of these three groups’ tweets.

The results of this readability user test revealed the following points.

- By looking up the top five weighted feature words of the learned tweet vectors, you can determine whether or not the learning is proceeding well. Therefore, the readability can be used for quality assessment of distributed representation learning.
- The conceptual axis can further filter the results of the sentiment analysis for social media mining. Each feature word itself or a combination of feature words becomes a conceptual axis. The image of social media filtering using the conceptual axis is shown in Figure 19. Tweets can be visualized with a meaningful arrangement along the conceptual axis by constructing the conceptual axis with feature words.

## **6.5 Expandability Evaluation**

Because we found support for the effectiveness of our method with a single domain benchmark, we constructed a diverse benchmark described in Section 4.3.2 to evaluate the expandability of our method. Also, we improved our dictionary with the aim of allowing third parties to reproduce our method. In this section, we describe the improvement of the dictionary and report the evaluation results of the diverse benchmark. In addition to the sentiment analysis task, a domain-independent test of our method with a word similarity task was also performed using the Wikipedia corpus.

### **6.5.1 The Improvement of the Word Semantic Vector Dictionary**

The dictionary aims to increase the readability of word and document embeddings by giving feature words as nodes of the hidden layer of the neural networks. The requirements of the dictionary for this purpose are as follows.

- Because the feature words correspond to each dimension of the paragraph vector, the number of feature words must be a multiple of 4 with good memory efficiency.
- In the case of nodes of the hidden layer, the weights of feature words given to many core words increase; therefore, such feature words might be deleted.

Table 28. Top feature words given to core words

Feature words	Number of core words
state·aspect	5,188
relationship·relation	4,460
power·degree	3,217
order·regularity	2,689
strong	2,645
positive	2,455

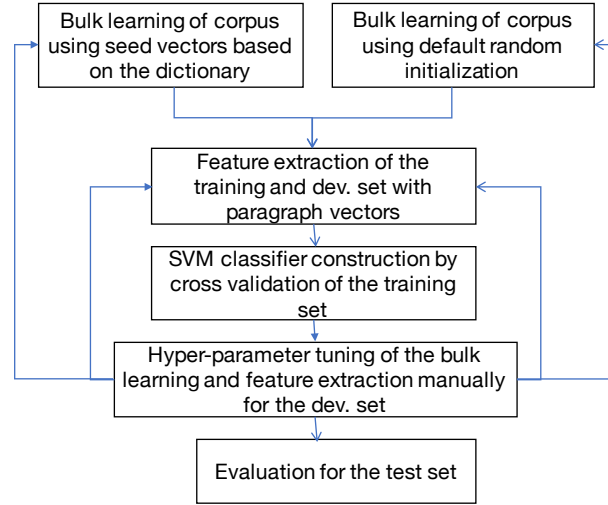


Figure 20. Procedures of sentiment analysis for the expandability evaluation.

Table 28 shows the top feature words given to the core words in Japanese. The feature words “state·aspect” and “relationship·relation” belonging to “abstract concept” in Table 1 are given to many core words. In particular, “state·aspect” is a feature word with larger weights in all examples shown in Section 6.4, which indicates that the influence of the feature word is too strong. Feature words below those shown in Table 28, when used for word expansion, play an important role in estimating positiveness [27]. Therefore, “state·aspect” and “relationship·relation” were deleted, and the number of feature words was set to 264 kinds, which is a multiple of 4.

Table 29. Hyper-parameter settings for learning word and paragraph vectors.

Hyper-parameters	PV-DM	PV-DBOW
Number of iterations in the dictionary (input vector)	10	10
Number of iterations in the dictionary (output vector)	1	1
Number of iterations for bulk learning	20	60
Number of iterations for feature extraction	200	60
Word occurrence frequency threshold	10	3
Window size (Conventional Method)	2 (5)	5 (5)
Downsample threshold for words	1e-6	1e-5

Table 30. Statistical information on the learning of the Twitter corpus

Items	Values
Number of examples	3.1 M examples
Number of uniq words1	82,031 (more than 10 examples appeared)
Size of word corpus1	76.7 M (more than 10 examples appeared)
Number of core words1	11,174 (more than 10 examples appeared)
Number of uniq words2	164,549 (more than 3 examples appeared)
Size of word corpus2	77.1M (more than 3 examples appeared)
Number of core words2	14,018 (more than 3 examples appeared)

### 6.5.2 Evaluation Using the Diverse and Large-scale Benchmark

We made seed vectors of Japanese core words using the dictionary and conducted an evaluation on 2-class classifications of the diverse and large-scale Japanese Twitter sentiment analysis benchmark described in Section 4.3.2 using two types of paragraph vector models. The evaluation flow of sentiment analysis using the benchmark is shown in Figure 20. The differences from the procedure shown in Figure 17 are that the hyper-parameter tuning was done manually without using Bayesian optimization and that hyper-parameters of bulk learning including seed vectors construction from the dictionary were also adjusted based on the macro-average F-score of the development set. We divided the corpus for bulk learning and corpus for feature extraction and cross-validation as follows.

**Corpus for bulk learning:** Total 3.1 M lines (No label: 2.2M tweets, No label (old benchmark): 0.56M tweets, Training set: 0.34M tweets, The dictionary: 20K core

Table 31. Evaluation results 4: Correlation coefficients between seed vectors and learned word vectors.

	PV-DM		PV-DBOW	
	Input Vector	Output Vector	Input Vector	Output Vector
After bulk learning	0.541	0.682	0.494	0.526
After feature extraction	0.423	0.686	0.494	0.526

words and their feature words)

**Corpus for feature extraction and cross-validation:** 2 class training set: 25,718 tweets, 2class dev. set: 6429 tweets, 2class test set: 6429 tweets

The hyper-parameter settings are shown in Table 29. The seed vectors of the output vectors were also set from the dictionary in this evaluation because the macro-average F-score of the dev. set was improved. The number of iterations in the dictionary was one, so the seed vectors of the output vectors were non-symmetric with the input vectors. We utilized the sum of the input nodes regarding the hidden layer of PV-DM because of the better macro-average F-score of the dev. set. Statistical information on the learning of the Twitter corpus is shown in Table 30. The upper half of the Table 30 shows statistical information of the corpus, in which the word occurrence frequency threshold is 10, for PV-DM and the lower half shows statistical information of the corpus, in which the word occurrence frequency threshold is 3, for PV-DBOW.

Table 31 presents the correlation coefficients between seed vectors and learned vectors in the closed test after bulk learning of the corpus and after feature extraction of the training and dev set. As for PV-DM, as shown in Figure 1, because the tweet ID was added to the context words, the word vectors were also updated when the feature extraction of tweets was done. However, as for PV-DBOW, the update of word vectors was stopped when the feature extraction of tweets was done because the extraction and the update of word vectors are independent. Therefore, the correlation of the word vectors of PV-DBOW is lower than that of the single domain benchmark shown in Table 25, but they still correlate with the seed vectors. With regard to PV-DM, the correlation of input vectors decreased as the scale of benchmarks increased, but it was considerably higher than the correlation of control tests shown in Table 25, and the correlation of output vectors was maintained. Therefore, in both the PV-DBOW and

Table 32. Evaluation results 5: Macro-average F-score for predicting positive and negative tweets in 2-class sentiment analysis.

	PV-DM		PV-DBOW	
	Conventional Method	Our Method	Conventional Method	Our Method
Dev. set	86.1	87.1*	88.85	89.01**
Test set	85.4	86.5***	88.02	88.19****
vs. Conventional Method *p=0.0019<0.05, ***p=0.0072<0.05      **p=0.17>0.05, ****p=0.13>0.05				

PV-DM, the meaning of the nodes in the hidden layer was maintained.

Table 32 presents the macro-average F-score in 2-class sentiment analysis. A comparison of the results of the single domain small benchmark shown in Table 27 revealed that the F-score improved by 9 points in PV-DM and by 5.69 points in PV-DBOW in the new benchmark. In the evaluation of the new benchmark, our method is still better than the conventional method in the PV-DM, and it is comparable to the conventional method in the PV-DBOW. Also, as shown in the following examples, the top five to ten weighted feature words were found to be related to tweets of diverse categories concerning robot cleaners, printing services, and makers also having higher weights.

**Robot cleaner B, Wonderful! The air in the room got really clean.** [dwelling, family, facility, environment, structure, newness, machine,]

**I am feeling Maker A's love for robot cleaner A and ...**

[customer, kindness, dwelling, art, peace, sound, company, human,]

**It is really useful to create booklets with the printing service at convenience stores.** [book, power, worth, newness, manufacture, education,]

**Oh wow! Maker B is allowed to marry same-sex within the company? It is wonderful that companies promote it!** [company, commerce, worth, public\_system, economy, international\_relation, social\_activity,]

Next, we examined the relationship between readability, which here is the correlation coefficients between seed vectors and learned word vectors, and the task accuracy. Table 33 shows the relationship between the correlation coefficients after bulk learning and feature extraction, and the task accuracy of the dev. set, which is the macro average F-score, for the PV-DM. Also, Table 34 shows it for the PV-DBOW. In the PV-DM,

Table 33. Relationship between the correlation coefficients after feature extraction and the task accuracy of the dev. set for the PV-DM.

Number of Bulk Learning	iter=0	iter=1	iter=5	iter=10	iter=15	iter=20
Macro Average F-score	84.5	84.7	85.9	86.7	86.8	87.1
Correlation Coefficients (Input Vector)	0.845	0.831	0.751	0.650	0.534	0.423

Table 34. Relationship between the correlation coefficients after feature extraction and the task accuracy of the dev. set for the PV-DBOW.

Number of Bulk Learning	iter=0	iter=1	iter=2	iter=10	iter=40	iter=60
Macro Average F-score	86.06	87.90	88.44	88.69	88.80	89.01
Correlation Coefficients (Input Vector)	1.0	0.598	0.571	0.502	0.500	0.494

which learns the word order information, the readability monotonously decreases according to the number of bulk learning of 3.1 M corpus, and the task accuracy peaks at 20 iterations. A trade-off relationship exists between readability and task accuracy up to the 20 iterations. However, the learning is rapidly performed by two iterations of bulk learning in the PV-DBOW, which learns the context information. A trade-off relationship also exists between readability and task accuracy, up to two iterations, but after that, the change in readability is small until 60 iterations. The realistic number of bulk learning is ten iterations for the PV-DM and two iterations for the PV-DBOW because both the readability and task accuracy are balanced.

Furthermore, we clarify the effect of the seed vectors from the word semantic vector dictionary by examining the accuracy of the task without collecting a significant amount of corpus. Table 35 shows the task accuracy of the development set for the proposed method, which uses only seed vectors from the word semantic vector dictionary, and for the conventional method, which uses random initial vectors without a significant amount of corpus, for the PV-DM. Also, Table 36 shows it for the PV-DBOW. In the PV-DM, the macro average F-score improved by 3.6 points using the word semantic vector dictionary. In particular, the recall rate of positive prediction with a smaller number of data was improved by 10%. However, in the PV-DBOW, the macro average F-score was improved by 3.2 points using the dictionary. In particular, the precision of



Table 35. No Bulk Learning for PV-DM.

	Our Method			Conventional Method		
	Precision	Recall	F-score	Precision	Recall	F-score
Negative Prediction	86.4%	90.0%	88.2	81.5%	92.1%	86.5
Positive Prediction	83.5%	78.1%	80.7	84.7%	67.8%	75.3
Macro Average	85.0%	84.1%	84.5 (+3.6)	83.1%	80.0%	80.9

Table 36. No Bulk Learning for PV-DBOW.

	Our Method			Conventional Method		
	Precision	Recall	F-score	Precision	Recall	F-score
Negative Prediction	87.6%	91.2%	89.4	85.9%	87.8%	86.8
Positive Prediction	85.5%	80.1%	82.7	80.4%	77.7%	79.0
Macro Average	86.6%	85.7%	86.1 (+3.2)	83.2%	82.8%	82.9

positive prediction improved by 5.1%. In the distributed representation learning, the context of a large number of data, which is the negative tweets in the present benchmark, is preferentially learned, but the meaning of the word and the context are considered to be harmonized using the dictionary.

### 6.5.3 Domain Independent Test

The original word semantic vector dictionary was developed to embed the knowledge of the encyclopedia as proposed in Chapter 3. Recently, Wikipedia<sup>14</sup>, which is a free encyclopedia, provides a Japanese domain independent corpus. Also, the performance of word embeddings is usually evaluated in a word similarity task. Moreover, the first word-similarity dataset in Japanese has been published [54]. The dataset consists of four parts of speech, i.e., adjectives, adverbs, verbs, and nouns, and it contains more than 4500 word pairs including rare words in addition to common words. We evaluated the word similarity task using the dataset for word embeddings learned with the Wikipedia corpus. In addition, the words of each part of speech were randomly chosen to show feature words with the top n weights and examples of the top n synonyms using word embeddings of our method.

<sup>14</sup><https://en.wikipedia.org/wiki/Wikipedia>

Table 37. Statistical information on the learning of the Wikipedia corpus

Items	Values
Number of paragraphs	1.19 M articles
Number of uniq words	0.55M (more than 10 articles appeared)
Size of word corpus	366.23M (more than 10 articles appeared)
Number of core words	15,866 (more than 10 articles appeared)
Required memory	2.94G
Required time for 3 iters	44 min (46 min) (Our Method (266 features)), 39 min (Conventional Method)

A word similarity task evaluated word embeddings constructed by our method and the conventional method both learning the Wikipedia corpus using the PV-DBOW/Skip-gram model. In the Japanese word similarity dataset<sup>15</sup>, each pair of words had an average point of similarity given by ten human annotators. The similarity between word pairs by each method was estimated by the cosine measure of the word vectors. If multiple words with word vectors were extracted from one word in the dataset, the weighted sum of the vector of each word was calculated. The weights of the constituent words were set to decrease in accordance with the occurrence order for adverbs and verbs, and the constituent words were set to have equal weights for adjectives and nouns. Also, all constituent words must have word vectors. The performance was evaluated by calculating the Spearman rank correlation coefficient between the word similarity in the Japanese word similarity dataset and the word similarity estimated by each method.

We removed all tags from the Wikipedia corpus<sup>16</sup> and modified it to one article per line. Learning too much context information for the word similarity task is not ideal, unlike the sentiment analysis task, so the number of learning iterations over the corpus in PV-DBOW was set to three (Epoch 3). We confirmed that Epoch 3 gives the best performance in the word similarity task in both the conventional method and the proposed method. Statistical information on the learning of the Wikipedia corpus is shown in Table 37. By changing 266 feature words to 264 feature words, the learning time improved by about 2 minutes. The difference in the learning time with the conventional method is mainly due to an increase in file IO.

The correlation coefficients shown in Table 38 were for a closed test between the seed vectors based on the core words of 15,866 words appearing in more than ten

<sup>15</sup><https://github.com/tmu-nlp/JapaneseWordSimilarityDataset>

<sup>16</sup><https://dumps.wikimedia.org/jawiki/latest/>

Table 38. Evaluation results 6: Correlation coefficients (closed test)

	Input Vectors	Output Vectors
PV-DBOW/Skip-gram	0.432	0.476

Table 39. Evaluation results 7: Word similarity task using Wikipedia corpus.

Part of Speech	Conventional Method	Our Method
Adjective	0.3470	0.3334
Adverb	0.2311	0.2415
Verb	0.3323	0.3431
Noun	0.2888	0.2840
Macro Average	0.2998	0.3005*

vs. Conventional Method \* $p=0.91>0.05$

articles in the Wikipedia corpus and the word vectors learned with the Wikipedia corpus. Because the size of the word corpus of the Wikipedia corpus is about five times that of the Twitter corpus, the correlation coefficient decreased, but the learned word vector was still correlated with the seed vector.

Table 39 presents the evaluation results on the word similarity task using the Wikipedia corpus. The evaluation results of our method were comparable to those of the conventional method in the Spearman rank correlation coefficient.

Both of the evaluation results showed that the core words learned with the Wikipedia corpus are correlated with the seed vectors based on the dictionary and that the evaluation results of our method are comparable to those of the conventional method in the word similarity task. Table 40 shows examples of the top  $n$  weighted feature words and the top  $n$  similar words for core words of each part of speech randomly selected. The boldface feature words are feature words that are given to the core words in the dictionary and that remained after learning the Wikipedia corpus. For example, the feature word of a special word such as “education” had a very high weight in a technical term such as “degree.” For the top five weighted feature words of an adverb such as “incomprehensible,” which is a rare word, three feature words remained from the dictionary, and two reasonable feature words were learned from the context. For a verb such as “get drunk,” only one feature word “food” remained, but others were appropriate feature words given in learning from the context. Top  $n$  similar words

Table 40. Example of Top n weighted feature words and similar words for core words

Core words	excellent (素晴らしい)	incomprehensible (不可思議)	get drunk (酔う)	degree (学位)
Top n weighted feature words	<b>worth</b> : 0.658 <b>positive</b> : 0.517 power-degree: 0.499 machine: 0.473 physics: 0.341	<b>idea</b> : 0.729 image: 0.679 <b>religion</b> : 0.534 <b>phenomena</b> : 0.450 sense: 0.427	traffic-transportation: 0.686 emotion: 0.642 <b>food</b> : 0.644 idea: 0.562 power-degree: 0.484	<b>education</b> : 1.763 <b>public_system</b> : 0.777 <b>special</b> : 0.727 possess: 0.668 physics: 0.548
Top n Similar Words based on the word vectors	素晴らしい: 0.757 すばらしい: 0.754 素晴らしいかつ: 0.744 すばらしく: 0.667 (thus far readings or inflection forms) perfect: 0.640	strange: 0.685 mystery: 0.633 eerie (不気味): 0.622 unusual: 0.617 knowledge (人智): 0.616 Bilocation (超常現象): 0.607	drunk: 0.733 drunken sickness: 0.705 gloom (憂さ): 0.703 excessive drinking: 0.702 heavy drinking: 0.695 merry drinker: 0.686	master: 0.849 doctor's degree: 0.827 bachelor's degree: 0.806 master's degree: 0.803 ph.d.: 0.793 bachelor: 0.784

Table 41. Word similarity test using Wikipedia corpus for the evaluation of 264 feature words and 266 feature words

Part of Speech	Num.	266 feature words	264 feature words
Adjective	814	0.3284	0.3334
Adverb	874	0.2304	0.2415
Verb	1409	0.3399	0.3431
Noun	1082	0.2828	0.2840
Macro Average	4179	0.2954	0.3005*

vs. 266 feature words  $0.05 < *p = 0.095 < 0.10$

based on word vectors are appropriate in any of the aforementioned cases. Therefore, our method is domain-independent.

Next, the effect of changing the number of feature words of the dictionary from 266 to 264 was investigated using the dataset. Table 41 shows the results of Spearman rank correlation coefficients in the case of 266 feature words and 264 feature words by learning with the Wikipedia corpus and using the word similarity dataset. Spearman rank correlation coefficients showed a tendency of superiority, with 264 feature words improving against that of 266 feature words in all parts of speech. The results show that the word semantic vector dictionary approaches the human sense by deleting only the two feature words of “state-aspect” and “relationship-relation,” which were given to many core words, and by changing the feature words from 266 to 264.

Furthermore, a comparison was made with the case where only the dictionary was

Table 42. Word similarity test for the evaluation of the dictionary and distributed representation

Part of Speech	Num.	Our Method	Num.	Dictionary only
Adjective	814	0.3334	171	0.2676
Adverb	874	0.2415	89	0.6569
Verb	1409	0.3431	346	0.4004
Noun	1082	0.2840	501	0.1523
Macro Average	(4179)	0.3005	(1142)	0.3693

used in the evaluation of the same dataset. Table 42 shows the Spearman rank correlation coefficient using only the word semantic vector dictionary in the word similarity dataset and that using the word vector by our method (the distributed representation learning of the Wikipedia corpus with the word semantic vector dictionary as seed vectors). In the case of using only the dictionary, the Spearman rank correlation coefficient with people is higher with the adverb and the verb, and that of the distributed representation learning increases with the noun and the adjective. These results show that the context is more important than the words themselves for the meaning of nouns and adjectives.

## 6.6 Conclusion

In this chapter, we proposed a novel method to give specific meaning to each node of a hidden layer in neural networks using a word semantic vector dictionary to enable the readability of word and document embeddings.

First, using a single domain sentiment analysis benchmark, we found that the evaluation results of our method were better than those of the conventional method in the macro-average F-score. Also, we tested the readability of tweet embeddings in a user test. A total of 52.4% of the top five weighted feature words were related to tweets.

Next, we improved the dictionary, which is suitable for distributed representation, and constructed a large-scale sentiment analysis benchmark as described in Section 4.3.2. In the evaluation of the benchmark, we found that our method is still better than the conventional method in the PV-DM and that it is comparable to the conventional method in the PV-DBOW. Compared to the results of the single domain benchmark,

the F-score improved by 9 points in PV-DM and by 5.69 points in PV-DBOW. Also, we found that the top five to ten weighted feature words were related to tweets of diverse categories.

Also, word similarity tasks using the Wikipedia corpus were evaluated. The evaluation results of our method were comparable to those of the conventional method in the Spearman rank correlation coefficient. Moreover, we found the top five weighted feature words to be related to the core words.

Moreover, our experimental results demonstrated that weights obtained based on learning and weights based on the dictionary are more strongly correlated in a closed test and more weakly correlated in an open test, compared with the results of a control test. As the word corpus used for learning expanded, the correlation with the seed vector decreased, but the correlation was maintained as high as about twice that of the default random setting.

Finally, we discuss what kind of properties the distributed representation learns and how it relates to the word semantic vector dictionary created by a human expert. In a diverse and reliable sentiment analysis benchmark, the distributed representation obtained using the word semantic vector dictionary as seed vectors was superior to the distributed representation obtained from the random initial values by 3.2 points or more in the macro average F-score. However, by using the seed vectors that learned the 3.1 M corpus iteratively enough, the difference in the F-score between both of the methods became slight. This result suggests that if the distributed representation in the sentiment analysis task is learned ideally for both the corpus amount and the number of iterations for learning, all the knowledge created by a human expert can be included. However, from the viewpoint of the readability of a distributed representation, we found that a comparable F-score and readability can be achieved with fewer iterations for learning using the word semantic vector dictionary.

## 7. Conclusion

### 7.1 Summary of Contributions

In this thesis, we introduced the word semantic vector dictionary based on the hypothesis that people memorize the concepts of words, articles, images, and tweets in association with feature words, as shown in the research target of section 1.2. The contributions of this thesis are as follows.

- We proposed a method to construct a word semantic vector dictionary representing encyclopedia knowledge by not using neural networks but inventing a bootstrap algorithm in Chapter 3. The contributions of this proposal are four points as follows: the multilayered classification of feature words, clarification of criteria for entry of these feature words for core words, use of multidimensional representation for coding of encyclopedia texts, and the application of coded knowledge for indexing images. Also, we proposed the learning function of word semantic vectors by the user, which is achieved because of the semantically readable distributed representations of words.
- We proposed an integration method to learn feature words expanded using a manually created word semantic vector dictionary with a paragraph vector model to solve the problem of word sparsity in Chapter 5. We showed the integration method improved the F-score of sentiment analysis by 3.2 points by learning the context information of a particular field even if words are sparse. Also, we showed that expanded feature words for Tweets could be used for error analysis of sentiment analysis. Moreover, we showed the effectiveness of the proposed method on a small but diverse benchmark, which consists of 141 labeled tweets, without using any unlabeled tweets. The proposed method outperformed the conventional method by 9 to 10 points in both the 3-class and the 2-class classifications F-score.
- We proposed a new method to achieve semantically readable distributed learning in Chapter 6. The key idea is to assign each feature word in the word semantic vector dictionary to one hidden node in neural networks. The weights of the neural networks are initialized by recursive extension of the dictionary. When

training unlabeled text data using neural networks, each learned feature is usually a black box. Because the proposed method learns the weights of preselected features, we tested the readability of tweet embeddings. A total of 52.4% of the top five weighted feature words were related to tweets. Also, we showed the expandability of the proposed method by using a diverse Japanese sentiment analysis benchmark and by conducting a word similarity task using the Wikipedia corpus.

- We have constructed a large-scale and diverse benchmark for evaluating reputation information extraction from Japanese Twitter in Section 4.3 and the word semantic vector dictionary in Chapter 3 and Section 6.5.1 so that a third party can reproduce our method.

## 7.2 Future Direction

- **Optimization of feature words:** The proposed method improved the readability of distributed representations because these distributed representations of words and paragraphs learned by neural networks are weights for each hidden node with a specific meaning in Chapter 6. Future research will be conducted to optimize 264 feature words. Learning the Wikipedia corpus with our method and displaying the top n weighted Wikipedia articles of each feature word will enable clarifying important feature words and trivial feature words in user tests.
- **Applying our method to deep neural networks:** State-of-the-art sentiment analysis in Twitter is based on ensembles of Convolutional Neural Networks (CNNs) and Long Short Term Memory Networks (LSTMs) presented at SemEval 2017 [9]. 100 million unlabeled tweets were prepared to pre-train word embedding using word2vec/fasText/GloVe. Distant training of word embedding was done with CNNs using a distant dataset of 5 million positive and 5 million negative tweets. The expansion of feature words in Chapter 5 will be tested whether or not our method is also valid for CNNs **without collecting such big data** because CNNs is considered mainly to learn context information.
- **Constructing English version of the dictionary:** First, we translated 20,330 core words from Japanese to English words or English phrases using the neural



machine translation API<sup>17</sup>. The reason for using neural machine translation rather than a Japanese-English dictionary was to translate difficult words extracted from encyclopedia or newspapers into English words and phrases used every day. As a result, they were translated into about 14,000 unique English words/phrases including translation errors. We employed three proofreaders using crowdsourcing to make the proofreading request of translated English words and phrases while referring to given feature words. The problem with the English version is that because we only merged the results of three proofreaders, it contains translation errors including misspellings, and English core words translated from various Japanese core words are given all the original feature words. The English version is planned to be improved after publication. We will also determine whether or not our method can apply to other languages using the standard English datasets of sentiment analysis tasks, word similarity tasks, and word analogy tasks in the both proposal of Chapter 5 and Chapter 6.

- **Readable performance improvement:** We will evaluate whether the meaning of each hidden node in Chapter 6 is more preserved by introducing regularization [62, 6] of learning process using our dictionary.
- **Creating new applications:** We believe that the performance of social media mining could be improved by our method without any annotated data because the performance of sentiment analysis and the readability of document embedding shows similar trends in Chapter 6. It would be more useful to create new applications that do not require labeling with a simple model. Also, the proposed method is not limited to the reputation information extraction in Chapter 5, and it will be useful in improving the accuracy of natural language processing tasks where the sparseness of words becomes a problem.
- **Solving document analogy questions in the feature word space:** Similar to solve the word analogy questions in the word2vec, if the document analogy questions in the feature word space are compelling, its applicability will expand. For example, if you want to find “meal” information from Kyoto sightseeing tweets, you could get the target tweet by adding the favorite feature word “food” and subtracting the undesirable feature word “mountain.”

---

<sup>17</sup><https://www.microsoft.com/en-us/translator/>

- **Interactive data visualization for the feature word space:** The readability of the top five weighted feature words given to each tweet of a short sentence is about 52%, but if you display tweets in the descending order of weight of any feature word, you could get useful tweets. Therefore, a visualization tool for arranging tweets along the conceptual axis with feature words on a two-dimensional surface as shown in Figure 19 becomes necessary for filtering social media.
- **Taking good points of the dictionary and distributed representation learning:** In the case of using only the dictionary for the word similarity task, the Spearman rank correlation coefficient with people is higher with the adverb and the verb, and that of the distributed representation learning with the Wikipedia corpus increases with the noun and the adjective as shown in Table 42. However, the evaluation results of our method were comparable to those of the conventional method as shown in Table 39. Therefore, it is conceivable to reduce the number of iterative learning of the corpus in adverbs and verbs for leaving a significant influence of the dictionary.
- **Toward automatic optimization of 264 feature words:** Using the Wikipedia corpus and topic models based on LDA [3], soft clustering of the 15,866 core words extracted from the corpus would be performed for 264 topics. We will then try to assign each label (feature word) to each topic by comparing the probability of the core word for each topic and the seed vectors based on the word semantic vector dictionary. Also, we would like to consider a methodology to construct the dictionaries fully automatically when applied to new languages using WordNet and ConceptNet, which are multilingual.

## References

- [1] R.F. Astudillo, S. Amir, W. Lin, M. Silva, and I. Trancoso. Learning Word Representations from Scarce and Noisy Data with Embedding Subspaces. In *Proc. of ACL-IJCNLP*, pages 1074–1084, 2015.
- [2] C.F. Baker, C.J. Fillmore, and J.B. Lowe. The Berkeley FrameNet Project. In *Proc. of ACL-COLING*, pages 86–90, 1998.
- [3] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.
- [4] W. Boag, P. Potash, and A. Rumshisky. TwitterHawk: A Feature Bucket Approach to Sentiment Analysis. In *Proc. of SemEval-2015*, pages 640–646, 2015.
- [5] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching Word Vectors with Subword Information. *Transactions of the ACL*, 5:135–146, 2017.
- [6] D. Bollegala, A. Mohammed, T. Maehara, and K. Kawarabayashi. Joint Word Representation Learning Using a Corpus and a Semantic Lexicon. In *Proc. of AAAI*, pages 2690–2696, 2016.
- [7] D. Bollegala, N. Okazaki, and T. Maehara. *Machine Learning of Web Data*. Kodansha, 2014.
- [8] F. Bond, H. Isahara, S. Fujita, K. Uchimoto, T. Kuribayashi, and K. Kanzaki. Enhancing the Japanese WordNet. In *Proc. of The 7th Workshop on Asian Language Resources, in conjunction with ACL-IJCNLP*, 2009.
- [9] M. Cliche. BB\_twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs. In *Proc. of SemEval-2017*, 2017.
- [10] I. Keshi et al. Overview of Benchmark for Japanese IR System Ver. 1.0 (BMIR-J1) (in Japanese). In *Proc. of IPS of Japan SIG Notes DBS106-19*, volume 106, pages 139–146, 1996.
- [11] K. Matsui et al. Test Collection for Information Retrieval Systems from the Viewpoint of Evaluating System Functions. In *International Workshop on Information Retrieval with Oriental Languages*, pages 42–27, 1996.

- [12] T. Kitani et al. BMIR-J2 - A Test Collection for Evaluation of Japanese Information Retrieval Systems (in Japanese). In *Proc. of IPS of Japan SIG Notes DBS114-3*, volume 114, pages 15–22, 1998.
- [13] M. Faruqui, J. Dodge, S.K Jauhar, C. Dyer, E. Hovy, and N.A. Smith. Retrofitting Word Vectors to Semantic Lexicons. In *Proc. of NAACL-HLT*, pages 1606–1615, May–June 2015.
- [14] S. I. Gallant. A Practical Approach for Representing Context and for Performing Word Sense Disambiguation Using Neural Networks. *Neural Computation*, 3(3):293–309, 1991.
- [15] D. Harman. Panel: Building and Using Test Collections. In *Proc. of SIGIR*, pages 335–337, 1996.
- [16] G.E. Hinton. Distributed Representations. Technical report, Carnegie-Mellon University, 1984.
- [17] G.E. Hinton. Learning Distributed Representation of Concepts. In *Proc. of Annual Conference of the Cognitive Science Society*, pages 1–12, 1986.
- [18] G.E. Hinton, J. L. McClelland, and D. E. Rumelhart. *Distributed Representations*, pages 77–109. MIT Press, 1986.
- [19] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of Tricks for Efficient Text Classification. In *Proc. of EACL*, pages 427–431, 2017.
- [20] I. Keshi, H. Ikeuchi, and K. Kuromusha. Associative image retrieval using knowledge in encyclopedia text. *Systems and Computers in Japan*, 27(12):53–62, 1996.
- [21] I. Keshi, H. Ikeuchi, and K. Kuromusha. Associative Image Retrieval Using Knowledge in Encyclopedia Text (in Japanese). *IEICE Trans. on Information and Systems (Japanese Edition)*, J79-D2(4):484–491, 1996.
- [22] I. Keshi, H. Ikeuchi, and Y. Obuchi. Encyclopedia Text Database Creation using Semantic Vectors (in Japanese). In *Proc. Advanced Database System Symposium '93*, 1993.

- [23] I. Keshi, T. Inui, and K. Ishikura. Associative Retrieval of Very Large Document Databases (in Japanese). In *I.E.I.C.E. Technical Report*, volume 92, pages 73–80, 1993.
- [24] I. Keshi, K. Kuromusha, R. Sato, A. Kawamura, H. Shimizu, H. Miyakawa, A. Ito, A. Matsuoka, H. Takezawa, and M. Konya. Interface Agent for Digital Information Appliances (in Japanese). *Sharp Technical Journal*, 77(9):15–20, 2000.
- [25] I. Keshi, Y. Suzuki, K. Yoshino, and S. Nakamura. Semantically Readable Distributed Representation Learning for Social Media Mining. In *Proc. of the 2017 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 716–722, August 2017.
- [26] I. Keshi, Y. Suzuki, K. Yoshino, and S. Nakamura. Semantically Readable Distributed Representation Learning and Its Expandability Using a Word Semantic Vector Dictionary. *IEICE Trans. on Information and Systems*, 2018 (Submitted).
- [27] I. Keshi, Y. Suzuki, K. Yoshino, G. Neubig, K. Ohara, T. Mukai, and S. Nakamura. Reputation Information Extraction from Twitter Using a Word Semantic Vector Dictionary. *IEICE Trans. on Information and Systems (Japanese Edition)*, J100-D(4):530–543, April 2017.
- [28] H. Kimoto, Y. Ogawa, T. Ishikawa, Y. Masunaga, T. Fukushima, T. Tanaka, H. Nakawatase, I. Keshi, J. Toyoura, T. Miyauchi, Y. Ueda, K. Matsui, T. Kitani, S. Miike, T. Sakai, T. Tokunaga, H. Tsuruoka, and T. Agata. Construction of a Test Collection for the Evaluation of Japanese Information Retrieval Systems. *Trans. IPS of Japan*, 40(9):3537–3553, 1999.
- [29] S. Kiritchenko, X. Zhu, and S.M. Mohammad. Sentiment Analysis of Short Informal Texts. *Journal of Artificial Intelligence Research (JAIR)*, 50:723–762, 2014.
- [30] T. Kitani, Y. Ogawa, T. Ishikawa, H. Kimoto, I. Keshi, J. Toyoura, T. Fukushima, K. Matsui, Y. Ueda, T. Sakai, T. Tokunaga, H. Tsuruoka, H. Nakawatase, and T. Agata. Lessons from BMIR-J2: A Test Collection for Japanese IR Systems. In *Proc. of SIGIR*, pages 345–346, 1998.

- [31] T. Kurita, T. Kato, I. Fukuda, and A. Sakakura. Sense Retrieval on a Image Database of Full Color Paintings (in Japanes). *Trans. IPS of Japan*, 33(11):1373–1383, 1992.
- [32] T. Kurita, H. Shimogaki, and T. Kato. A Personal Interface for Similarity Retrieval on an Image Database System (in Japanese). *Trans. IPS of Japan*, 31(2):227–237, 1990.
- [33] J.H. Lau and T. Baldwin. An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation. In *Proc. of Workshop on Representation Learning for NLP*, pages 78–86, 2016.
- [34] Q.V. Le and T. Mikolov. Distributed Representations of Sentences and Documents. In *Proc. of ICML*, pages 1188–1196, 2014.
- [35] D.B. Lenat, M. Prakash, and M. Shepherd. CYC: Using Common Sense Knowledge to Overcome Brittleness and Knowledge Acquisition Bottlenecks. *AI magazine*, 6(4), 1985.
- [36] Y. Liu, Z. Liu, T. Chua, and M. Sun. Topical Word Embeddings. In *Proc. of AAAI*, pages 2418–2424, 2015.
- [37] H. Luo, Z. Liu, H. Luan, and M. Sun. Online Learning of Interpretable Word Embeddings. In *Proc. of EMNLP*, pages 1687–1692, 2015.
- [38] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3781, 2013.
- [39] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed Representations of Words and Phrases and their Compositionality. *CoRR*, abs/1310.4546, 2013.
- [40] T. Mikolov, W. Yih, and G. Zweig. Linguistic Regularities in Continuous Space Word Representations. In *Proc. of NAACL*, pages 746–751, 2013.
- [41] G.A. Miller. WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41, 1995.

- [42] S. Mohammand, S. Kiritchenko, and X. Zhu. NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. In *Proc. of SemEval-2013*, pages 321–327, 2013.
- [43] B. Murphy, P. Talukdar, and T. Mitchell. Learning Effective and Interpretable Semantic Models using Non-Negative Sparse Embedding. In *Proc. of COLING*, 2012.
- [44] P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani, and V. Stoyanov. SemEval-2016 Task 4: Sentiment Analysis in Twitter. In *Proc. of SemEval-2016*, 2016.
- [45] P. Nakov, S. Rosenthal, Z. Kozareva, V. Stoyanov, A. Ritter, and T. Wilson. SemEval-2013 Task 2: Sentiment Analysis in Twitter. In *Proc. of SemEval-2013*, pages 312–320, 2013.
- [46] J. Pennington, R. Socher, and C.D. Manning. GloVe: Global Vectors for Word Representation. In *Proc. of EMNLP*, pages 1532–1543, 2014.
- [47] L. Quanzhi, S. Sameena, L. Xiaomo, N. Armineh, and F. Rui. Tweet Topic Classification Using Distributed Language Representations. In *Proc. of IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 81–88, 2016.
- [48] X. Rong. word2vec Parameter Learning Explained. *CoRR*, abs/1411.2738, 2014.
- [49] S. Rosenthal, N. Farra, and P. Nakov. SemEval-2017 Task 4: Sentiment Analysis in Twitter. In *Proc. of SemEval-2017*, 2017.
- [50] S. Rosenthal, P. Nakov, S. Kiritchenko, S. M Mohammad, A. Ritter, and V. Stoyanov. SemEval-2015 Task 10: Sentiment Analysis in Twitter. In *Proc. of SemEval-2015*, pages 451–463, 2015.
- [51] S. Rosenthal, A. Ritter, P. Nakov, and V. Stoyanov. SemEval-2014 Task 9: Sentiment Analysis in Twitter. In *Proc. of SemEval-2014*, pages 73–80, 2014.
- [52] D. E. Rumelhart and P. M. Todd. *Learning and Connectionist Representations*, pages 3–30. MIT Press, 1993.

- [53] T. Sakai, T. Kitani, Y. Ogawa, T. Ishikawa, H. Kimoto, I. Keshi, J. Toyoura, T. Fukushima, K. Matsui, Y. Ueda, T. Tokunaga, H. Tsuruoka, H. Nakawatase, T. Agata, and N. Kando. BMIR-J2: A Test Collection for Evaluation of Japanese Information Retrieval Systems. *SIGIR Forum*, 33(1):13–17, 1999.
- [54] Y. Sakaizawa and M. Komachi. Construction of a Japanese Word Similarity Dataset. *CoRR*, abs/1703.05916, 2017.
- [55] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian Optimization of Machine Learning Algorithms. *Advances in Neural Information Processing Systems*, pages 2951–2959, 2012.
- [56] R. Speer, J. Chin, and C. Havasi. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *Proc. of AAAI*, pages 4444–4451, 2017.
- [57] R. Speer and C. Havasi. Representing General Relational Knowledge in ConceptNet 5. In *Proc. of LREC*, 2012.
- [58] F. Sun, J. Guo, Y. Lan, J. Xu, and X. Cheng. Sparse Word Embeddings Using  $\ell_1$  Regularized Online Learning. In *Proc. of IJCAI*, pages 2915–2921, 2016.
- [59] S. Suzuki. Probabilistic Word Vector and Similarity based on Dictionaries. In *Proc. of CICLing*, pages 564–574, 2003.
- [60] S. Vosoughi, P. Vijayaraghavan, and D. Roy. Tweet2Vec: Learning Tweet Embeddings Using Character-level CNN-LSTM Encoder-Decoder. In *Proc. of SIGIR*, pages 1041–1044, 2016.
- [61] D.L. Waltz and J.B. Pollack. *Connectionist Models and Their Implications: Readings from Cognitive Science*, chapter Massively Parallel Parsing: A Strongly Interactive Model of Natural Language Interpretation. Ablex Publishing Corp., 1988.
- [62] C. Xu, Y. Bai, J. Bian, B. Gao, G. Wang, X. Liu, and T. Liu. RC-NET: A General Framework for Incorporating Knowledge into Word Representations. In *Proc. of CIKM*, pages 1219–1228, 2014.



# Publication List

## Journal papers

1. I. Keshi, Y. Suzuki, K. Yoshino, and S. Nakamura.  
Semantically Readable Distributed Representation Learning and Its Expandability Using a Word Semantic Vector Dictionary.  
IEICE Trans. on Information and Systems, 2018.  
(under review, corresponds to Chapter 6, Section 4.3.2)
2. I. Keshi, Y. Suzuki, K. Yoshino, G. Neubig, K. Ohara, T. Mukai, and S. Nakamura.  
Reputation Information Extraction from Twitter Using a Word Semantic Vector Dictionary (in Japanese).  
IEICE Trans. on Information and Systems (Japanese Edition), Vol.J100-D, No.4, pp.530-543, April 2017.  
(corresponds to Chapter 5, Section 4.3.1)
3. I. Keshi, H. Ikeuchi, and K. Kuromusha.  
Associative Image Retrieval Using Knowledge in Encyclopedia Text (in Japanese).  
IEICE Trans. on Information and Systems (Japanese Edition), Vol.J79-D-II, No.4, pp.484-491, April 1996.  
(corresponds to Chapter 3)
4. H. Kimoto, Y. Ogawa, T. Ishikawa, Y. Masunaga, T. Fukushima, T. Tanaka, H. Nakawatase, I. Keshi, J. Toyoura, T. Miyauchi, Y. Ueda, K. Matsui, T. Kitani, S. Miike, T. Sakai, T. Tokunaga, H. Tsuruoka, and T. Agata.  
Construction of a Test Collection for the Evaluation of Japanese Information Retrieval Systems (in Japanese).  
Trans. IPS of Japan, Vol.40, No.9, pp.3537-3553, Sep. 1999.  
(corresponds to Section 4.2)

## Journal papers (letters, technical reports etc.)

1. T. Sakai, T. Kitani, Y. Ogawa, T. Ishikawa, H. Kimoto, I. Keshi, J. Toyoura, T. Fukushima, K. Matsui, Y. Ueda, T. Tokunaga, H. Tsuruoka, H. Nakawatase, T. Agata, and N. Kando.  
BMIR-J2: a Test Collection for Evaluation of Japanese Information Retrieval Systems.  
ACM SIGIR Forum, Vol.33, Issue 1, pp.13-17, Fall 1999.  
(corresponds to Section 4.2)

## International Conferences

1. I. Keshi, Y. Suzuki, K. Yoshino, and S. Nakamura.  
Semantically Readable Distributed Representation Learning for Social Media Mining.  
In Proc. of IEEE/WIC/ACM Web Intelligence (WI), pp.716-722, Aug. 2017.  
(corresponds to Chapter 6)
2. T. Kitani, Y. Ogawa, T. Ishikawa, H. Kimoto, I. Keshi, J. Toyoura, T. Fukushima, K. Matsui, Y. Ueda, T. Sakai, T. Tokunaga, H. Tsuruoka, H. Nakawatase, and T. Agata.  
Lessons from BMIR-J2: a Test Collection for Japanese IR Systems.  
In Proc. of SIGIR, pp.345-346, Aug. 1998.  
(corresponds to Section 4.2)

## Domestic Conferences

1. 芥子育雄, 鈴木 優, 吉野 幸一郎, 大原 一人, 向井 理朗, 中村 哲.  
分散的意味表現学習のための単語意味ベクトル辞書 Ver.2 と日本語 Twitter 極性分析ベンチマークについて.  
情処研報, Vol.2017-NL-231, Vol.2017-SLP-116, No.8, pp.1-7, 2017 年 5 月.  
(corresponds to Section 4.3.2, Chapter6)
2. 芥子育雄, 鈴木 優, 吉野 幸一郎, ニュービグ グラム, 大原 一人, 向井 理朗, 中村 哲.

可読性のある分散的意味表現学習に向けて.  
関西データベースワークショップ 2016, 2016 年 9 月.  
(corresponds to Chapter 6)

3. 芥子育雄, 鈴木優, 吉野幸一郎, 大原一人, 向井理朗, 中村哲.  
単語・パラグラフの分散表現を用いた Twitter からの日本語評判情報抽出.  
第 8 回 データ工学と情報マネジメントに関するフォーラム論文集 (A1-3),  
2016 年 2 月.  
(corresponds to Chapter 5, Section 4.3.1)
4. 河村晃好, 黒武者健一, 佐藤亮一, 芥子育雄.  
グループ嗜好モデルと視聴履歴を利用したコンテンツ検索サーバの試作.  
情処研報, 2001-DBS-125, pp.177-184, 2001 年 7 月.  
(corresponds to Chapter 3)
5. 芥子育雄, 佐藤亮一, 宮川晴光, 黒武者健一, 清水仁.  
機能意味検索と操作自動実行に基づくナビゲーションソフト.  
情処研報, HCI Vol.99, No.35, pp.43-48, 1999 年 5 月.  
(corresponds to Chapter 3)
6. 芥子育雄, 他.  
パソコン用オン・デマンド・ユーザインタフェースの開発.  
創造的ソフトウェア育成事業 最終成果発表会論文集, 情報処理振興事業協  
会, pp.251-258, 1998 年.  
(corresponds to Chapter 3)
7. 木谷強, 小川泰嗣, 石川徹也, 木本晴夫, 中渡瀬秀一, 芥子育雄, 豊浦潤, 福島  
俊一, 松井くにお, 上田良寛, 酒井哲也, 徳永健伸, 鶴岡弘, 安形輝.  
日本語情報検索システム評価用テストコレクション BMIR-J2.  
情処研報, DBS Vol.98, No.2, pp.15-22, 1998 年 1 月.  
(corresponds to Section 4.2)
8. 黒武者健一, 芥子育雄.  
連想検索技術を利用した文書の要約.  
Proc. Advanced Database System Symposium ' 97, pp.135-142, 1997 年 12 月.  
(corresponds to Chapter 3)

9. 芥子育雄, 他.  
パソコン用オン・デマンド・ユーザインタフェースの開発.  
創造的ソフトウェア育成事業 中間成果発表会論文集, 情報処理振興事業協会, pp.333-339, 1997 年.  
(corresponds to Chapter 3)
10. 黒武者健一, 池内洋, 勘座浩幸, 芥子育雄.  
意味ベクトルによる WWW 情報自動収集と情報発信支援.  
Proc. Advanced Database System Symposium '96, 1996 年 12 月.  
(corresponds to Chapter 3)
11. 芥子育雄, 木本晴夫, 田中智博, 石川徹也, 増永良文, 小川泰嗣, 豊浦潤, 福島俊一, 宮内忠信, 三池誠司, 松井くにお, 木谷強.  
情報検索システム評価用ベンチマーク Ver.1.0(BMIR-J)1 について.  
情処研報, DBS Vol.96, No.11, pp.139-146, 1996 年 1 月.  
(corresponds to Section 4.2)
12. 小川泰嗣, 木本晴夫, 田中智博, 石川徹也, 増永良文, 芥子育雄, 豊浦潤, 福島俊一, 宮内忠信. 日本語情報検索システムのためのベンチマークの構築.  
情処研報, DBS Vol.94, No.86, pp.145-152, 1994 年 10 月.  
(corresponds to Section 4.2)
13. 池内洋, 芥子育雄.  
意味ベクトルによる画像検索の試み.  
日本ソフトウェア科学会第 11 回大会論文集, pp.413-416, 1994 年 11 月.  
(corresponds to Chapter 3)
14. 芥子育雄, 池内洋, 田中理恵子.  
意味ベクトルによる画像検索.  
画像電子学会研究会, pp.21-24, 1994 年 4 月.  
(corresponds to Chapter 3)
15. 芥子育雄, 池内洋, 小淵保司.  
意味ベクトルによる百科事典テキストデータベースの構築.  
Proc. Advanced Database System Symposium '93, pp.227-234, 1993 年 12 月.  
(corresponds to Chapter 3)

16. 木本晴夫, 田中智博, 石川徹也, 増永良文, 小川泰嗣, 芥子育雄, 福島俊一, 豊浦潤.  
情報検索システム評価用データベースの構築の提案.  
情処研報, FI Vol.93, No.98, pp.1-8, 1993 年 11 月.  
(corresponds to Section 4.2)
17. 芥子育雄, 乾隆夫, 奥西稔幸.  
意味ベクトルによる自己組織型百科事典データベース構築の試み.  
人工知能学会全国大会論文集, Vol.7, pp.317-320, 1993 年 7 月.  
(corresponds to Chapter 3)
18. 芥子育雄, 乾隆夫, 石鞍謙一郎.  
大規模文書データベースから連想検索.  
信学技報, AI Vol.92, No.99, pp.73-80, 1993 年 1 月.  
(corresponds to Chapter 3)

## Others

1. I. Keshi, K. Kuromusha, R. Sato, A. Kawamura, H. Shimizu, H. Miyakawa, A. Ito, A. Matsuoka, H. Takezawa, and M. Konya.  
Interface Agent for Digital Information Appliances (in Japanese).  
Sharp Technical Journal, Vol.77, No.9, pp.15-20, Aug. 2000.  
(corresponds to Chapter 3)
2. I. Keshi, H. Ikeuchi, and K. Kuromusha.  
Associative Image Retrieval Using Knowledge in Encyclopedia Text.  
Systems and Computers in Japan, Vol.27, No.12, pp.53-62, 1996.  
(corresponds to Chapter 3)
3. I. Keshi, H. Ikeuchi, R. Tanaka, and M. Osaki.  
Multimedia Information Retrieval Using Knowledge in Encyclopedia Texts (in Japanese).  
Sharp Technical Journal, Vol.60, pp.31-36, Dec. 1994.  
(corresponds to Chapter 3)

4. R. Tanaka, I. Keshi, H. Ikeuchi.  
Image retrieval using semantic vectors (in Japanese).  
Journal of Information Processing and Management, Vol.37, No.7, pp.579-585,  
Oct. 1994.  
(corresponds to Chapter 3)

## Awards

1. 第8回データ工学と情報マネジメントに関するフォーラム, 学生プレゼンテーション賞, 2016.
2. 「連想検索写真データベース」, (財) 国際AI財団「優秀AI製品賞」, 1994.

## Related Publications

### Journal papers

1. I. Keshi, N. Fukuda, and Y. Fujimoto.  
A Knowledge-based Framework in an Intelligent Assistant System for Making Documents.  
Future Generation Computer Systems Vol.5, No.1, 1989.
2. R. Mizoguchi, I. Keshi, Y. Isomoto, and O. Kakusho.  
An Implementation of Intelligent Man - Machine Interface Using DBMS -On the Speech Database SPEECH - DB- (in Japanese).  
Trans. IPS of Japan, Vol.25, No.3, pp.404-412, May 1984.
3. R. Mizoguchi, N. Maeda, M. Hamaguchi, I. Keshi, M. Yanagida, and O. Kakusho.  
Speech Data Base with an Intelligent Access Mechanism -SPEECH- DB (in Japanese).  
Trans. IPS of Japan, Vol.24, No.3, pp.271-280, May 1983.
4. K. Ohara, I. Keshi, and T. Onoye.  
Load Adaptive Decoder Controlling Method for Simultaneous Browsing of Video

Contents (in Japanese).

Annual Report of Image Electronics Engineering, Vol.39, No.6, pp.1095-1103, 2010.

5. M. Ise, Y. Ogasahara, K. Watanabe, M. Hatanaka, T. Onoye, H. Niwamoto, I. Keshi, and I. Shirakawa.

Design and Implementation of Home Network Protocol for Appliance Control Based on IEEE 802.15.4.

International Journal of Computer Science and Network Security, Vol.7, No.7, pp.20-30, July 2007.

6. K. Watanabe, M. Ise, T. Onoye, H. Niwamoto, and I. Keshi.

An Energy-efficient Architecture of Wireless Home Network Based on MAC Broadcast and Transmission Power Control.

IEEE Trans. Consumer Electronics, Vol.53, No.1, pp.124-130, Feb. 2007.

## International Conferences

1. I. Keshi, S. Yamaguchi, T. Mukai, H. Yamamura, H. Hinode, H. Inoue, M. Nakazawa, S. Ota, A. Azuma, and Y. Matsumoto.

3D Revolving User Interface Optimized for Tablets and a Cloud-Based MediaService.

In Proc. of International Display Workshops (IDW'11), Nov. 2011.

2. I. Keshi, Y. Shiraishi, H. Niwamoto, M. Okada, and H. Yamamoto.

Is Home Network Application Acceptable or Not?

In Proc. of IEEE ISCAS, Vol.5, pp.5337-5340, May 2005.

3. I. Keshi and B. Katz.

Speech-Act Based Message Conversation System.

In Proc. of International Workshop on CSCW, Berlin, Vol.4, No.91, pp.59-73, April 1991.

4. I. Keshi, N. Fukuda and Y. Fujimoto.

A Knowledge-based Framework in an Intelligent Assistant System for Making Documents.

In Proc. of the International Conference on Artificial Intelligence (AI'87 Japan)  
Abstracts pp.286-294, 1987.

5. Y. Kawamura, Y. Manabe, T. Onoye, K. Ohara, H. Okada, and I. Keshi.  
Implementation of Simultaneous Video Decoding on Multicore Processor.  
In Proc. of International Symposium on Communications, Control and Signal  
Processing, March 2010.
6. K. Watanabe, M. Ise, T. Onoye, H. Niwamoto, and I. Keshi.  
An Energy-efficient Architecture of Wireless Home Network Based on MAC  
Broadcast and Transmission Power Control.  
In Proc. of International Conference on Consumer Electronics, Digest of Tech-  
nical Papers, pp.1-20, Jan. 2007.

## Domestic Conferences

1. 富士谷康, 堀井遼太, 芥子育雄, 西尾信彦.  
テレビ番組視聴履歴を用いた電子書籍推薦.  
ユビキタス・ウェアラブルワークショップ 2014 論文集, PP.3, 2014 年 12 月.
2. 倉持淳子, 芥子育雄.  
家電メーカーにおけるビッグデータ応用事例.  
人間工学, Vol.50, PP.84-85, 2014 年.
3. 松下裕丈, 河村侑輝, 尾上孝雄, 大原一人, 芥子育雄.  
携帯機器における動画像ストリーム高速簡略復号の一手法.  
信学技報, SIS Vol.109, No.78, PP.19-24, 2009 年 6 月 4 日.
4. 河村侑輝, 真鍋安武, 尾上孝雄, 大原一人, 岡田浩行, 芥子育雄.  
動画像並列復号のマルチコアプロセッサへの実装.  
信学技報, SIS Vol.108, No.86, pp.51-56, 2008 年.
5. 佐藤文代, 芥子育雄, 新井宏之.  
加速度センサと携帯端末を用いたセルフケア支援システムの試作.  
電子情報通信学会ソサイエティ大会講演論文集 Vol.2006, pp.202, 2006 年 9  
月 7 日.



6. 渡邊賢治, 伊勢正尚, 藤田玄, 畠中理英, 尾上孝雄, 庭本浩明, 芥子育雄, 白川功.  
無線ホームネットワークにおける消費電力および即時性の改善手法.  
信学技報, CS Vol.105, No.637, pp.25-30, 2006 年 3 月.
7. 伊勢正尚, 小笠原泰弘, 渡邊賢治, 畠中理英, 尾上孝雄, 庭本浩明, 芥子育雄, 白川功.  
IEEE 802.15.4 を用いたホームネットワーク向け無線ネットワークプロトコル.  
信学技報, CS Vol.105, No.633, pp.19-24, 2006 年 3 月.
8. 鮎澤篤, 余梯榕, 漆原育子, 山内規義, 木村真, 佐藤光, 庭本浩明, 川尻百恵, 芥子育雄.  
超小型・双方向無線センサモジュール Ni3 を用いた実験用センサネットワークシステム.  
電子情報通信学会総合大会講演論文集, pp.29-30, 2005 年 3 月.
9. 白石裕美, 庭本浩明, 芥子育雄, 岡田実, 山本平一.  
次世代ホームネットワーク環境におけるアプリケーション受容性の検証.  
情処研報, 2004-HI-112, pp.9-16, 2005 年 1 月.
10. 盧承烈, 小笠原泰弘, 伊勢正尚, 畠中理英, 尾上孝雄, 庭本浩明, 芥子育雄, 白川功.  
ユニバーサルプラグアンドプレイ技術を用いたホームネットワーク一構成方式.  
信学技報, CAS Vol.104, No.556, pp.7-12, 2005 年 1 月.
11. 芥子育雄.  
ベクトル空間モデルに基づくフルテキストサーチシステム.  
人工知能学会全国大会論文集, Vol.6, pp.343-346, 1992 年 6 月.
12. Ikuo Keshi.  
Speech-Act Based Message Conversation System.  
6th Symposium on Human Interface, pp.541-544, 1990.
13. 水谷直樹, 芥子育雄, 藤本好司.  
ビジネスレター作成支援システム：意味内容による文書の登録・検索支援.

情報処理学会全国大会講演論文集 37th, pp.1016-1017, 1988 年 9 月.

14. 芥子育雄, 福田尚行, 藤本好司.  
オブジェクト指向の概念に基づく文書要約システムについて.  
電気関係学会関西支部連合大会講演論文集, S55, 1987 年.
15. 芥子育雄, 福田尚行, 藤本好司.  
文書要約・検索システムの一構成法.  
情報処理学会全国大会講演論文集 34th, pp.1461-1462, 1987 年.
16. 福田尚行, 芥子育雄, 藤本好司.  
ビジネスレター作成支援システム —知識ベース作成支援環境—.  
情報処理学会全国大会講演論文集 34th, pp.1459-1460, 1987 年.
17. 芥子育雄, 福田尚行, 藤本好司.  
ビジネスレター作成支援システム.  
情報処理学会全国大会講演論文集 32th, pp.1055-1056, 1986 年.

## Others

1. R. Horii, K. Fujitani, M. Hori, T. Inui, H. Yamamura, I. Keshi.  
Retaining Consumer Attention Using Recommendation (in Japanese).  
Sharp Technical Journal, Vol.109, PP.13-16, July 2015.
2. I. Keshi, N. Fukuda, and Y. Fujimoto  
Intelligent Assistant System for Making Business Letters (in Japanese).  
Sharp Technical Journal, Vol.39, pp.27-30, 1989.