Semantic Shift Stability: Efficient Way to Detect Performance Degradation of Word Embeddings and Pre-trained Language Models

Shotaro Ishihara * Nikkei, Inc.
Hiromu Takahashi * Independent researcher
Hono Shirai Nikkei, Inc.
shotaro.ishihara@nex.nikkei.com
hiromu.takahashi56@gmail.com
hono.shirai@nex.nikkei.com

Abstract

Word embeddings and pre-trained language models have become essential technical elements in natural language processing. While the general practice is to use or fine-tune publicly available models, there are significant advantages in creating or pre-training unique models that match the domain. The performance of the models degrades as language changes or evolves continuously (*semantic shift*), but the high cost of model building inhibits regular re-training, especially for the language models. This study designs a methodology for observing time-series performance degradation of word embeddings and pre-trained language models using semantic shift in a corpus. We define an efficiently computable metric named Semantic Shift Stability based on the degree of semantic shift. In the experiments, we create models that vary by time series and reveal the performance degradation in two datasets, Japanese and English. Several case studies demonstrate that Semantic Shift Stability supports decision-making as to whether a model should be re-trained. The source code is available at https://github.com/Nikkei/semantic-shift-stability.

1 Introduction

The use of word embeddings and pre-trained language models has become common practice in natural language processing. Word embeddings like word2vec (Mikolov et al., 2013) are used in many applications, and pre-trained language models starting with BERT (Devlin et al., 2019) are updating state-of-the-art performance on a daily basis. Researchers and developers use or fine-tune such kinds of models to their own tasks.

While the general practice is to start from publicly available models, there are also significant advantages in creating or pre-training unique models that match the domain. In regard to pre-trained language models, for example, SciBERT (Beltagy et al., 2019), BioBERT (Lee et al., 2020), and FinBERT (Araci, 2019) are proposed. These models have performed better than other BERT models on downstream domain-specific tasks. A similar approach is traditionally used in word embeddings. There are numerous studies and applications of obtaining word embeddings in their own corpora.

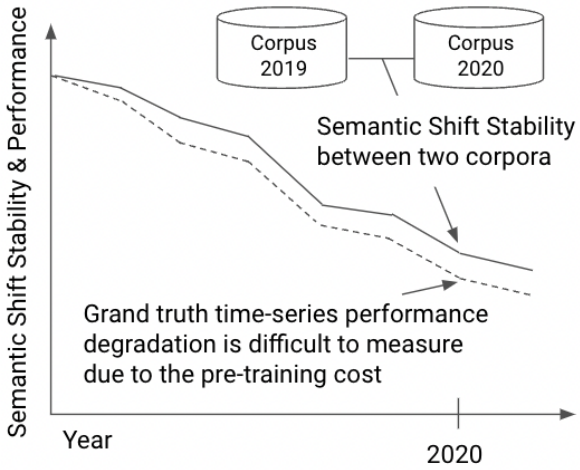

Figure 1: Methodology for observing time-series performance degradation by Semantic Shift Stability. It is difficult from a cost perspective to create a pre-train language model each time and compare the performance. Instead, by monitoring the degree of semantic shift of the corpora from period to period, we can estimate time-series performance degradation.

In creating domain-specific language models, we have to be careful of time-series changes in the characteristics of the corpus. Language changes continuously, especially when there are some socially important events. The semantic shift (Kutuzov et al., 2018) of existing words and the appearance of new words are occurring regularly. Some have reported that such time-series changes cause degradation of performance (Jaidka et al., 2018; Sato et al., 2020; Loureiro et al., 2022). Henceforth, we refer to this phenomenon as time-series

* These authors contributed equally.

Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 205–216
November 20–23, 2022.

performance degradation.

One of the solutions to tackle time-series performance degradation is re-training, but the high cost of model building is a bottleneck especially with language models. It is reported that large-scale pre-training requires large amounts of computation. For example, GPT-3 with 175B parameter consumed several thousand petaflop/s-days of compute during pre-training (Brown et al., 2020), and PaLM with 540B parameter was trained on 6144 TPU v4 chips (Chowdhery et al., 2022). This trend is accelerated by empirical scaling laws for language model performance (Kaplan et al., 2020), where the loss scales as a power-law with model size, dataset size, and the amount of compute used for training.

This study designs a methodology for observing time-series performance degradation of word embeddings and pre-trained language models using semantic shift in a corpus. The degree of semantic shift is computed by comparing two word2vec models created from corpora of different time-span. Monitoring performance leads to the decision whether the model should be re-trained (Figure 1).

The methodology has the advantage of avoiding large-scale training to measure performance. The required input is two word2vec models, which can be created much more efficiently than pre-training of language models. For word embeddings, it is also a benefit if we can infer the downstream task performance without experiments.

Our contributions are as follows.

1. We define an efficiently computable metric named Semantic Shift Stability based on the degree of semantic shift, and propose to use it for detecting time-series performance degradation of word embeddings and pre-trained language models (Section 3).
2. We create models that vary by time-series and reveal the performance degradation via the experiments on two corpora, not only English but also Japanese. In particular, we pre-train and analyze 12 RoBERTa models on a corpus of Japanese financial news at different time-span (Section 4).
3. We demonstrate case studies that the Semantic Shift Stability supports decision-making as to whether a model should be re-trained. Our experiments report that a large time-series performance degradation occurs in the years when Semantic Shift Stability is smaller (Section 5).

2 Related Work

This section describes the related work from three perspectives and highlights our study.

2.1 Semantic Shift

Changes in human language have long been studied from a variety of perspectives (Bloomfield, 1933). There are known linguistic and cultural factors (Hamilton et al., 2016). In addition to its linguistic and sociological importance, changes in human language also attract interest from the perspective of data science, such as natural language processing and information retrieval (Kutuzov et al., 2018).

As large corpora become available, there have been accelerated efforts to capture the semantic shift using word embeddings (Traugott, 2017). For example, (Gulordava and Baroni, 2011) compared the distribution in corpora from the 1960s and 1990s and identified a cultural shift in which the word *sleep* became more negative in meaning. (Guo et al., 2021) analyzed a Twitter corpus over time and observed changes in word meaning during the COVID-19 pandemic. Furthermore, (Giulianelli et al., 2022) detected semantic shift using pre-trained language models. One of the challenges is that there is limited research on this area in non-English languages (Kutuzov et al., 2018).

2.2 Time-series Performance Degradation

Time-series performance degradation is a long-standing problem in machine learning (Quinonero-Candela et al., 2008). It is a common problem in predictive modeling that occurs when the joint distribution of inputs and outputs differs between training and test stages. Differences in distribution are often caused by the lapse of time.

This issue has also been discussed in the progress of natural language processing. (Loureiro et al., 2022) pointed out that the time variable has been largely neglected in the literature on natural language processing. They pre-trained multiple language models on a time-split Twitter corpus and investigated the differences in performance. (Mohawesh et al., 2021) reported that differences in the distribution of input and output datasets negatively affects the performance of prediction models in the detection of fake reviews. There is also a direction to incorporate time-series information into word embeddings (Rosenfeld and Erk, 2018; Hofmann et al., 2021) and pre-trained language models (Hombaiah et al., 2021).

2.3 Domain-Specific Language Models

The idea of creating embedding representations from a large dataset of unlabeled text has become an essential element in natural language processing. This trend started with simple single word embeddings such as word2vec, and has evolved into more advanced pre-trained language models such as ELMo (Peters et al., 2018), BERT, and GPT-3, etc. In the creation of word embeddings and pre-trained language models, Web domain corpora are often used. Many works use Wikipedia and other resources crawled from the Internet.

Past work has shown that using a domain-specific corpus has the potential to improve performance (Peng et al., 2019). Some conduct additional pre-training to a model that has been pre-trained on a general corpus, while others tackle the issue from scratch on a domain-specific corpus. In some cases, the latter method, which does not mix domains, leads to superior results (Gu et al., 2021).

Language is one of the domain factors, and there are several researches in non-English languages. For example, there are GPT-like models created by the corpora of Chinese (Zeng et al., 2021; Su et al., 2022) and Korean (Kim et al., 2021). Nevertheless, there are not many practical examples due to computational cost and other difficulties.

2.4 Our study highlight

Our study crosses the three research areas described in this section. Specifically, we extend the semantic shift methodology to address the problem of time-series performance degradation in domain-specific language models and word embeddings. To conclude this section, we highlight our study.

First, our effort is one of the first attempts to propose an efficient way to detect time-series performance degradation. There are some studies that recognize the existence of semantic shift and create some models incorporate time-series information. However, few studies have been designed as decision-making support application without large re-training.

Next, our experiments, especially on Japanese corpora, would become unique and valuable case studies. There is insufficient research on semantic shift and domain-specific language models for languages other than English.

Finally, when it comes to the stage of practicality, discussions of time-series performance degradation and model re-training are becoming more important. Domain-specific language models are gradually being proposed.

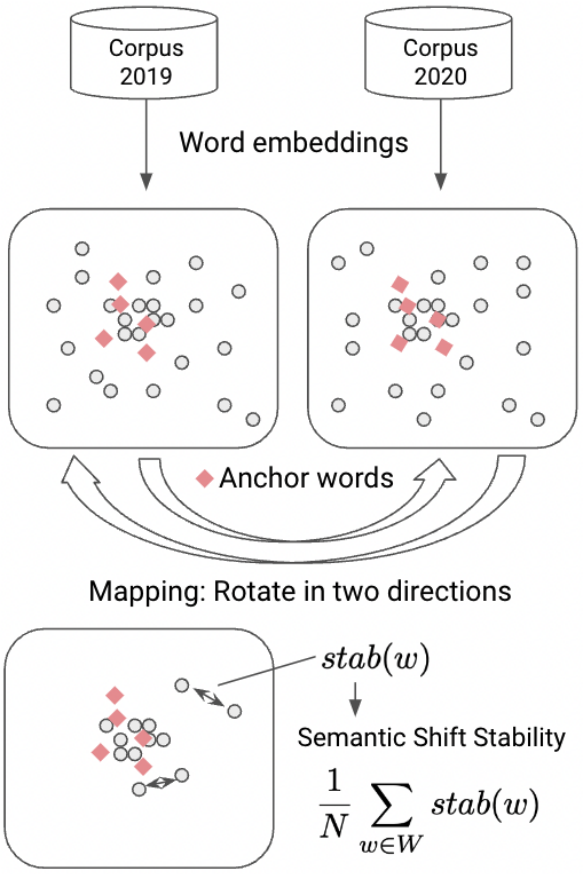

Figure 2: Procedure to calculate Semantic Shift Stability from two corpora. First, word embeddings are created. Then, we set anchor words and introduce a rotation matrix. Finally, Semantic Shift Stability is calculated by averaging the stability of each word.

3 Semantic Shift Stability

In this section, we define a metric named Semantic Shift Stability based on the degree of semantic shift of two corpora. We propose to use it for detecting time-series performance degradation of word embeddings and pre-trained language models.

Semantic Shift Stability is a metric calculated for whole word embeddings. We compute the stability of the semantic shift ($stab(w)$) on each word w and use the average of all words in the common vocabulary of two word embeddings as the overall score.

The procedure to calculate $stab(w)$ and Semantic Shift Stability from two corpora is described in Figure 2. There are four steps followed in the method proposed by (Guo et al., 2021): 1. Create word embeddings, 2. Set anchor words, 3. Intro-

duce the rotation matrix, and 4. Calculate $stab(w)$. Our new point in this study is that we define a metric that averages $stab(w)$, to quantify the semantic shift of two corpora.

3.1 Create word embeddings

The first step is to create word embeddings from each of the two corpora for comparison. For word embeddings, word2vec is used.

3.2 Set anchor words

The second step is to set *anchor words*, which are the starting points for comparing two word embeddings in the next step. We assume that the meaning of frequently appearing words does not change over time and that the local structure is preserved. It is based on the idea that the rate of semantic shift follows a negative power of word frequency (Hamilton et al., 2016). Under this assumption, the top 1000 frequent words are set as anchor words.

3.3 Introduce rotation matrix

The third step is to introduce a rotation matrix by taking two trained word embeddings. Specifically, the matrices of anchor words are taken from the two word embeddings, aligned and optimized while preserving cosine similarity (Schönemann, 1966). This optimization problem is solved by applying singular value decomposition to obtain the optimal rotation matrices between the two embedding spaces. We call this step mapping.

3.4 Calculate $stab(w)$

The fourth step is to calculate $stab(w)$, where the degree of semantic shift of the word can be observed by computing the cosine similarity of the word embedding in each model. However, since the average similarity is low for one-way mapping (Azarbonyad et al., 2017), the same process are applied in the reverse direction. The definition of $stab(w)$ that compares word embeddings i and j is as follows.

$$stab(w) = \frac{sim_{ij}(w) + sim_{ji}(w)}{2}$$

$$sim_{ij}(w) = \cos(R^{ji}R^{ij}V_w^i, V_w^i)$$

The smaller $stab(w)$ is, the larger the difference between the two word embeddings, and the more the word is considered to have changed its meaning. Here, cos is the cosine similarity, R^{ji} is the rotation matrix used for mapping from model j to i, and V_w^i is the embedding of the word w in model i.

3.5 Semantic Shift Stability

We define a metric to calculate the degree of semantic shift of the entire model using the average $stab(w)$. The smaller this value is, the greater the degree of change of the entire model. Here, W is a vocabulary commonly included in the word2vec model i and j, and N is the number of W.

$$\text{Semantic Shift Stability} = \frac{1}{N}\sum_{w \in W} stab(w)$$

3.6 Enumerate words with small $stab(w)$

We can infer the reason for the semantic shift by enumerating words with small $stab(w)$. This is one of the advantages of using the methodology to analyze the difference.

4 Preliminary Experiments: Time-series Performance Degradation

In this section, we create models that vary by time-series and analyze them to reveal the performance degradation. The purpose of this preliminary experiments was to quantify the performance degradation that should be detected in the next section. The rest of this section describes the dataset, model creation, and their time-series performance degradation. We used RoBERTa (Liu et al., 2019) for pre-trained language models and word2vec for word embeddings. RoBERTa is a optimized version of BERT, and word2vec is a well-known word embeddings.

4.1 Dataset

We prepared the following two corpora:

Nikkei Japanese financial news corpus from the Nikkei Online Edition [1] from March 23, 2010, when the service was launched, to December 31, 2021. It contains several genres such as business, lifestyle, international, sports, market, economy, society, and politics.

NOW English news corpus from News on the Web (NOW) (Davies, 2017). The period is from 2010 to April 2022. It contains articles from various news media such as *TechCrunch*, *ESPN*, *Ars Technica*, *Salon*, *CNET*, and *Politico*.

Table 1: Training time and loss for the pre-trained RoBERTa models. Starting in 2010, the training corpus was increased year by year. As the size of the corpus increased, there was a trend of increasing training time and decreasing losses.

Corpus	Time (sec)	Loss	Corpus size
2010	8387	6.81	151 MB
2010-2011	20791	5.42	391 MB
2010-2012	34007	4.26	636 MB
2010-2013	46764	3.79	874 MB
2010-2014	58510	3.27	1.09 GB
2010-2015	69279	3.13	1.30 GB
2010-2016	82267	2.99	1.54 GB
2010-2017	96455	2.71	1.79 GB
2010-2018	111204	2.82	2.06 GB
2010-2019	125481	2.67	2.33 GB
2010-2020	142336	2.69	2.62 GB
2010-2021	140196	2.82	2.80 GB

4.2 Pre-train RoBERTa models

We pre-trained multiple RoBERTa models with different time-span of the Nikkei corpus. The architecture was RoBERTa base with 125M parameters including 12 layer, 768 hidden, and 12 heads. The corpus was prepared for 12 patterns; the years 2010, 2010-2011, ... , and 2010-2021 as listed in Table 1. As the size of the corpus increased, there was a trend of increasing training time and decreasing losses.

Pre-training language models required large computational cost. For example, the RoBERTa 2010-2021 took appropriately 140 thousand seconds (39 hours) and $ 1278 to pre-train. We used Amazon EC2 P4 Instances for computational resource. This instance provides eight A100 GPUs and its on-demand price per hour is $ 32.77.

We used Transformers (Wolf et al., 2020) for the implementation. Training epochs were set at 50 for all models and the hyperparameters were set as follows according to the instruction [2]: max sequence length: 128, batch size: 32, learning rate: 0.0003, and weight decay (Loshchilov and Hutter, 2017): 0.001. The optimizer was Adafactor (Shazeer and Stern, 2018).

We used SentencePiece (Kudo and Richardson, 2018) as a tokenizer in the setting of unigram language model (Kudo, 2018). SentencePiece does not require prior segmentation and can directly generate vocabulary from the raw text. This feature is useful for languages such as Chinese and Japanese, where there are no explicit spaces between words. Figure 3 shows the difference in tokenizer work between the corpora used for training. The tokenizer trained on the new corpus was able to process the newly introduced words appropriately.

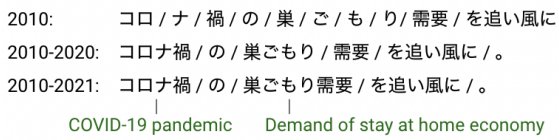

Figure 3: Tokenizers trained from the Nikkei corpora with different time-span. The tokenizer, which is trained to include the post-2020 corpus, is able to properly separate words that are new in COVID-19. The tokenizer trained only on the 2010 corpus break them up into smaller pieces.

4.3 Degradation of RoBERTa models

We measured RoBERTa time-series performance degradation using the Pseudo-perplexity (PPPL) (Salazar et al., 2020) following a previous study (Loureiro et al., 2022). The PPPL is computed on the basis of the idea of iteratively replacing each token in a sequence with a mask and summing the corresponding conditional log probabilities. This approach is especially suited to masked language models such as RoBERTa. To see the change in time-series performance, the PPPL is computed for combinations of the RoBERTa models and the corpora.

Table 2 showed that, as expected, the performance of the model degraded with each time-series. The PPPL is a metric in which a smaller value is better. The overall trend is that the numbers worsen as one moves to the right side of the table and improve as one moves to the bottom. For example, the model for RoBERTa 2010 shows 800.57 PPPL for the Nikkei corpus 2010. The newer the corpus for evaluation, the worse the PPPL. RoBERTa 2010 model shows 1076.00 PPPL against the Nikkei corpus 2020, but performance improves as RoBERTa is trained on newer corpora.

4.4 Create word2vec models

We created multiple word2vec models with different time-span of the Nikkei and the NOW corpora. Each corpus was prepared for 12 patterns by year; the years 2010, 2011, ... , and 2021.

Building word2vec is much more efficient than pre-trained language models reported in Section

[1] https://aws.amazon.com/marketplace/seller-profile?id=c8d5bf8a-8f54-4b64-af39-dbc4aca94384

[2] https://github.com/huggingface/transformers/tree/main/examples/flax/language-modeling

Table 2: Pseudo-perplexity (PPPL) results computed for combinations of the different RoBERTa models and time-span corpora. The PPPL is a metric in which the smaller value is better. The overall trend is that the worse the performance as one moves to the right side (the evaluation corpora become newer) and the better the performance as one moves to the bottom (the newer corpora used for RoBERTa pre-training).

	Evaluation										
RoBERTa	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
2010	**800.57**	883.31	913.05	930.00	924.94	933.43	962.32	992.69	1,011.57	1,012.89	**1,076.00**
2010-2011		192.98	222.05	235.83	237.15	240.40	260.57	269.68	278.32	282.36	300.60
2010-2012			63.13	70.25	73.17	74.54	82.86	86.41	90.58	92.51	96.80
2010-2013				38.62	41.93	43.81	49.08	51.17	53.76	55.46	58.17
2010-2014					23.04	24.88	28.19	29.42	31.33	32.58	34.24
2010-2015						17.33	20.16	21.19	22.77	23.59	24.79
2010-2016							16.73	18.21	19.73	20.37	21.86
2010-2017								12.26	13.72	14.38	15.42
2010-2018									15.44	16.58	18.15
2010-2019										11.74	13.16
2010-2020										10.73	11.15
2010-2021										18.04	18.21

4.2. Training with the Nikkei corpus for one year (around 200 MB) took about 20 minutes on a laptop (MacBook Pro, 2.4 GHz 8 core Intel Core i9).

We used gensim (Řehůřek and Sojka, 2010) to build the word2vec models. For the Nikkei corpus, we performed an additional process to handle Japanese texts. HTML tags and URLs were removed as text preprocessing. We used MeCab (Kudo, 2005) for text splitting and mecab-ipadic-NEologd (Sato et al., 2017) for the dictionary.

We confirmed that the training of word2vec was sufficient by comparing the performance with other Japanese models. The word2vec model created using the Nikkei corpus showed competitive performance to other models. For comparison, we used WikiEntVec (Suzuki et al., 2018), Shiroyagi [3] and chiVe [4]. Appendix A describes the details of this evaluation.

4.5 Degradation of word2vec models

We measured word2vec time-series performance degradation using a classification task, following a previous study (Kutuzov et al., 2018). The aim was to see how well word2vec trained on a previous corpus performs against a newer corpus (the corpus 2021). As input, we used the keywords of the article in the Nikkei corpus and the words of the article texts in the NOW corpus. The average of the word embeddings for each word was treated as feature (Shen et al., 2018) and LightGBM (Ke et al., 2017) was used as a classifier. The classification objective was the genres of the article. The eight genres for the Nikkei corpus are described in Section 4.1. For the NOW corpus, we regarded the six news media as genres written in Section 4.1.

Table 3: The transition of the word2vec performance on the corpus 2021. The results showed that models trained on newer corpus performed better.

Corpus	Nikkei	Nikkei	NOW	NOW
Train	w2v	w2v, lgbm	w2v	w2v, lgbm
2011	0.8036	0.1886	0.9056	0.7562
2012	0.8060	0.1102	0.9084	0.7324
2013	0.8090	0.3768	0.9070	0.7759
2014	0.8087	0.3989	0.9064	0.7850
2015	0.8113	0.2234	0.9078	0.7831
2016	0.8157	0.4092	0.9108	0.7330
2017	0.8180	0.2610	0.9094	0.7088
2018	0.8193	0.3946	0.9081	0.7376
2019	0.8233	0.4684	0.9093	0.7758
2020	**0.8284**	**0.5412**	**0.9182**	**0.8621**

As shown in Table 3, the performance generally degraded as the training corpus moved into the past. There were two experimental settings for each corpus. The first setting was that only the word2vec model was trained on the corpus of a specific year. LightGBM was trained on the corpus 2021. The second setting was that both the word2vec model and LightGBM were trained. In both experimental settings of the two corpora, the corpus 2020 showed the highest performance.

5 Experiments

In this section, we calculated Semantic Shift Stability and analyzed the relationship to the time-series performance degradation shown in Section 4.

5.1 Semantic Shift Stability

We calculated Semantic Shift Stability for the two corpora, shifting the window width by one year. There were two corpora of reference year and the

[3] https://github.com/shiroyagicorp/japanese-word2vec-model-builder

[4] https://github.com/WorksApplications/chiVe

Table 4: Semantic Shift Stability. Note that there are two corpora of reference year and the year. The smaller the value, the greater the difference between the two comparisons. It was smaller in 2016 and 2020 for both corpora. In the Nikkei corpus, it was also smaller in 2012.

Reference year	Year	Nikkei	NOW
2011	**2012**	**0.9770**	0.9840
2012	2013	0.9815	0.9855
2013	2014	0.9825	0.9850
2014	2015	0.9860	0.9805
2015	**2016**	**0.9800**	**0.9610**
2016	2017	0.9860	0.9830
2017	2018	0.9840	0.9875
2018	2019	0.9850	0.9710
2019	**2020**	**0.9710**	**0.9610**
2020	2021	0.9835	0.9835

year. The flow was to compare the corpus 2011 and 2012, then the corpus 2012 and 2013, etc. All results are listed in Table 4. Note that the smaller Semantic Shift Stability value, the greater the difference between the two comparisons.

Nikkei Semantic Shift Stability was smaller in the 2012, 2016, and 2020. The first change, inferred from social events, was probably due to the Great East Japan Earthquake in 2011. The United States presidential election 2016 can be raised as a possible reason for the second change. The third change could be because of the arrival of the COVID-19 pandemic. Although these are only analogies of social events, the methods described in Section 3.6 can help in the discussion. For example, when we analyzed the third change per word, the words enumerated were as follows: infection spread, new coronavirus, infection etc.

NOW Semantic Shift Stability was smaller in the corpora 2016, and 2020. The reasons for the changes are considered to be the same as for the Nikkei corpus. When we analyzed the change of 2016 per word, the words enumerated were: donald, trump etc. This implied that the change was because of Donald Trump, who won the United States presidential election 2016.

5.2 Case study on RoBERTa

This case study demonstrates that large time-series performance degradation occurred in the years when Semantic Shift Stability was smaller. We analyzed the relationship between time-series performance degradation of RoBERTa models calculated in Section 4.3 and Semantic Shift Stability introduced in Section 5.1. As preparation, the raw data of PPPL results in Table 2 were converted to year-to-year performance differences (Table 5).

The objective of converting the table is to clarify the impact on performance per year. First, for each RoBERTa model, we calculated the percentage of performance degradation compared to the newest year included in the training corpus. Temporary table is shown in Appendix B. Then, the difference from the previous year was calculated for each RoBERTa model.

We focus on three years (2012, 2016, and 2020) for Table 5 because Semantic Shift Stability was smaller. At the corpus 2012 column, there was the highest value in the whole table. Note that the discussion for the corpus 2012 was a bit difficult because there were not enough previous periods. Looking at the corpus 2016 column, almost all RoBERTa models showed significant performance degradation. The corpus 2016 caused the most performance degradation for almost all models trained before 2016. After 2016, the highest values appeared in the 2020 column. Performance degradation in 2020 was greater than in 2019 for all RoBERTa models.

5.3 Case study on word2vec

This case study demonstrates that large time-series performance degradation occurred in the years when Semantic Shift Stability was smaller. We analyzed the relationship between time-series performance degradation of word2vec models calculated in Section 4.5 and Semantic Shift Stability introduced in Section 5.1. There were two experimental settings, and we investigated the relationship to Semantic Shift Stability for each setting.

We found that in years when Semantic Shift Stability was smaller, using that year's corpus for training improved the performance compared to the previous year. Figures 4 and 5 show the visualization of the first setting, in which we only trained word2vec. The red wavy line shows the performance against the evaluation corpus (the corpus 2021), as a difference compared to the previous year. Semantic Shift Stability, the blue line, was smaller in 2012, 2016, and 2020. In both figures, there was a significant performance improvement in 2016 and 2020. The correlation coefficient is -0.4855 and -0.8861, respectively.

On the contrary, the second setting in which we

Table 5: Converted Pseudo-perplexity results for clarifying the impact on performance from year to year. First, for each model, we calculated the percentage of performance degradation compared to the newest year included in the training corpus. Then, we calculated the difference from the previous year, respectively. Looking at the corpus 2016 column, almost all RoBERTa models showed significant performance degradation. Coefficient means the correlation coefficient with Semantic Shift Stability.

	Evaluation											Coefficient
RoBERTa	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	
2010	0.00	10.33	3.72	2.12	-0.63	1.06	3.61	3.79	2.36	0.17	7.88	-0.7775
2010-2011		0.00	15.06	7.14	0.68	1.68	10.45	4.72	4.47	2.09	9.46	-0.7010
2010-2012			0.00	11.28	4.63	2.17	13.17	5.62	6.60	3.07	6.79	-0.3776
2010-2013				0.00	8.59	4.86	13.65	5.43	6.70	4.41	7.00	-0.3271
2010-2014					0.00	7.96	14.36	5.36	8.26	5.47	7.17	-0.1952
2010-2015						0.00	16.28	5.96	9.13	4.73	6.95	-0.1340
2010-2016							0.00	8.87	9.07	3.86	8.87	-0.3122
2010-2017								0.00	11.94	5.35	8.53	-0.0364
2010-2018									0.00	7.41	10.15	-
2010-2019										0.00	12.11	-
2010-2020										0.00	3.92	-
2010-2021										0.00	0.89	-

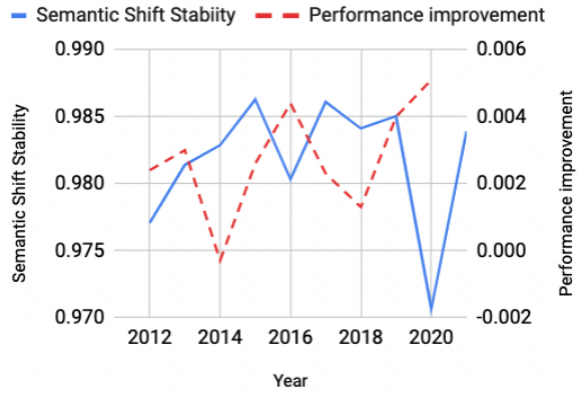

Figure 4: Relationship between Semantic Shift Stability and performance improvement difference of word2vec trained on the Nikkei corpus #. We found that in years when Semantic Shift Stability was small, using that year's corpus for training improved the performance compared to the previous year.

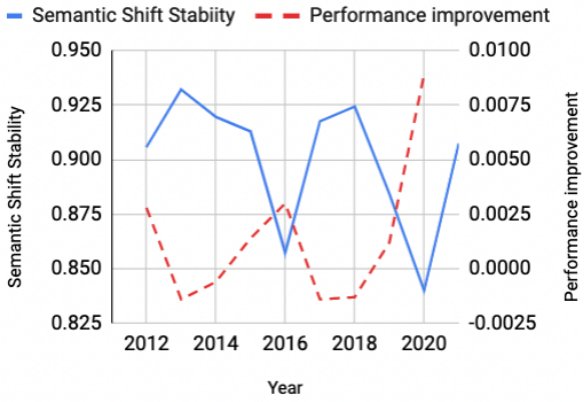

Figure 5: Relationship between Semantic Shift Stability and performance improvement difference of trained on the NOW corpus #. We found that in years when Semantic Shift Stability was small, using that year's corpus for training improved the performance compared to the previous year.

trained word2vec and LightGBM showed a relatively undistinguished trend. The visualization of the second setting is shown in Appendix C. This may be because LightGBM was also trained on a corpus, making it difficult to see the effect of word2vec.

6 Conclusion and Future Work

This study designs a methodology for observing time-series performance degradation of word embeddings and pre-trained language models by Semantic Shift Stability. It is a metric that can be calculated more efficiently than pre-training language models, which requires large computational cost. Monitoring performance via Semantic Shift Stability supports decision-making as to whether a model should be re-trained. We created word embeddings and pre-trained language models that vary by time-series. In particular, we pre-trained and analyze 12 RoBERTa models on a corpus of Japanese financial news at different time-span. We quantified the time-series performance degradation in experiments on two corpora, Japanese and English. The experiments confirmed that a large time-series performance degradation occurred in the years when Semantic Shift Stability was smaller.

Our effort is one of the first attempts to propose an efficient way to detect time-series performance degradation, designed as a decision-making support application without large re-training. In future work, we plan to conduct further experiments with more diverse corpora and models. In the

present study, the relationship between Semantic Shift Stability and time-series performance degradation was discussed qualitatively based on the calculated quantitative information. Additional research should lead us to explore ways to formulate this discussion in a more persuasive manner.

Acknowledgements

We thank anonymous reviewers for their careful reading of our manuscript and for their many insightful comments and suggestions. In addition, we appreciate AWS Japan for prototyping support in pre-training language models. We are also grateful to Tony Bucher for proofreading. Finally, we thank all members of Nikkei, Inc. for their devoted support and discussion.

References

Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.

Hosein Azarbonyad, Mostafa Dehghani, Kaspar Beelen, et al. 2017. Words are malleable: Computing semantic shifts in political and media discourse. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1509–1518, Singapore, Singapore. Association for Computing Machinery.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

L. Bloomfield. 1933. *Language*. Holt, Rinehart and Winston, New York.

Tom Brown, Benjamin Mann, Nick Ryder, et al. 2020. Language models are Few-Shot learners. *Adv. Neural Inf. Process. Syst.*, 33:1877–1901.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, et al. 2022. PaLM: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Davies. 2017. The new 4.3 billion word NOW corpus, with 4–5 million words of data added every day. *The 9th International Corpus Linguistics Conference*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, et al. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mario Giulianelli, Andrey Kutuzov, and Lidia Pivovarova. 2022. Do not fire the linguist: Grammatical profiles help language models detect semantic change. In *Proceedings of the 3rd Workshop on Computational Approaches to Historical Language Change*, pages 54–67, Dublin, Ireland. Association for Computational Linguistics.

Yu Gu, Robert Tinn, Hao Cheng, et al. 2021. Domain-Specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare*, 3(1):1–23.

Kristina Gulordava and Marco Baroni. 2011. A distributional similarity approach to the detection of semantic change in the Google Books ngram corpus. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 67–71, Edinburgh, UK. Association for Computational Linguistics.

Yanzhu Guo, Christos Xypolopoulos, and Michalis Vazirgiannis. 2021. How COVID-19 is changing our language : Detecting semantic shift in twitter word embeddings. *arXiv preprint arXiv:2102.07836*.

William L Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1489–1501, Berlin, Germany. Association for Computational Linguistics.

Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2021. Dynamic contextualized word embeddings. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6970–6984, Online. Association for Computational Linguistics.

Spurthi Amba Hombaiah, Tao Chen, Mingyang Zhang, et al. 2021. Dynamic language models for continuously evolving content. *arXiv preprint arXiv:2106.06297*.

Keisuke Inohara and Akira Utsumi. 2021. JWSAN: Japanese word similarity and association norm. *Language Resources and Evaluation*, pages 1–29.

Kokil Jaidka, Niyati Chhaya, and Lyle Ungar. 2018. Diachronic degradation of language models: Insights from social media. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 195–200, Melbourne, Australia. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, et al. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Guolin Ke, Qi Meng, Thomas Finley, et al. 2017. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Boseop Kim, Hyoungseok Kim, Sang-Woo Lee, et al. 2021. What changes can large-scale language models bring? intensive study on HyperCLOVA: Billions-scale Korean generative pretrained transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3405–3424, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Takumitsu Kudo. 2005. Mecab : Yet another part-of-speech and morphological analyzer.

Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, et al. 2018. Diachronic word embeddings and semantic shifts: a survey. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1384–1397, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, et al. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Yinhan Liu, Myle Ott, Naman Goyal, et al. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Daniel Loureiro, Francesco Barbieri, Leonardo Neves, et al. 2022. TimeLMs: Diachronic language models from Twitter. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 251–260, Dublin, Ireland. Association for Computational Linguistics.

Tomás Mikolov, Kai Chen, Greg Corrado, et al. 2013. Efficient estimation of word representations in vector space. In *Workshop Track Proceedings of 1st International Conference on Learning Representations*, Scottsdale, Arizona, USA.

Rami Mohawesh, Son Tran, Robert Ollington, et al. 2021. Analysis of concept drift in fake reviews detection. *Expert Systems with Applications*, 169:114318.

Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 58–65, Florence, Italy. Association for Computational Linguistics.

Matthew E Peters, Mark Neumann, Mohit Iyyer, et al. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. 2008. *Dataset Shift in Machine Learning*. MIT Press.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.

Alex Rosenfeld and Katrin Erk. 2018. Deep neural models of semantic shift. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 474–484, New Orleans, Louisiana. Association for Computational Linguistics.

Yuya Sakaizawa and Mamoru Komachi. 2018. Construction of a Japanese word similarity dataset. In *Proceedings of the 11th International Conference on Language Resources and Evaluation*, Miyazaki, Japan. European Language Resources Association.

Julian Salazar, Davis Liang, Toan Q Nguyen, et al. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

Shoetsu Sato, Jin Sakuma, Naoki Yoshinaga, et al. 2020. Vocabulary adaptation for domain adaptation in neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4269–4279, Online. Association for Computational Linguistics.

Toshinori Sato, Taiichi Hashimoto, and Manabu Okumura. 2017. Implementation of a word segmentation dictionary called mecab-ipadic-neologd and study on how to use it effectively for information retrieval. In *Proceedings of the 23rd Annual Meeting of the Association for Natural Language Processing*. The Association for Natural Language Processing.

Peter H Schönemann. 1966. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604. PMLR.

Dinghan Shen, Guoyin Wang, Wenlin Wang, et al. 2018. Baseline needs more love: On simple Word-Embedding-Based models and associated pooling mechanisms. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 440–450, Melbourne, Australia. Association for Computational Linguistics.

Hui Su, Xiao Zhou, Houjing Yu, et al. 2022. WeLM: A Well-Read pre-trained language model for Chinese. *arXiv preprint arXiv:2209.10372*.

Masatoshi Suzuki, Koji Matsuda, Satoshi Sekine, et al. 2018. A joint neural model for Fine-Grained named entity classification of wikipedia articles. *IEICE Transactions on Information and Systems*, E101.D(1):73–81.

Elizabeth Closs Traugott. 2017. Semantic change. In *Oxford Research Encyclopedia of Linguistics*.

Thomas Wolf, Lysandre Debut, Victor Sanh, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wei Zeng, Xiaozhe Ren, Teng Su, et al. 2021. PanGu-α: Large-scale autoregressive pretrained Chinese language models with auto-parallel computation. *arXiv preprint arXiv:2104.12369*.

A Evaluation of Created word2vec

We confirmed that the training of word2vec was sufficient by comparing the performance with other Japanese models. The word2vec model created using the Nikkei corpus showed competitive performance as shown in Table 6. As a representative of our word2vec models, a word2vec model was created with the Nikkei corpus from March 23, 2010 to October 31, 2019. For comparison, we used WikiEntVec, Shiroyagi and chiVe. WikiEntVec and Shiroyagi were trained in Japanese Wikipedia, and chiVe was trained in Japanese Web corpus.

Table 6: Comparison of Japanese word2vec models. The word2vec model created using the Nikkei corpus showed competitive performance to other models.

Model	Nikkei	WikiEntVec	Shiroyagi	chiVe
Dimension	300	200	50	300
Vocabulary	493,531	1,015,474	335,476	3,644,628
JWSD-adv	**0.281**	0.182	0.155	0.255
JWSD-verb	0.251	0.149	0.223	**0.260**
JWSD-noun	0.274	0.250	0.203	**0.310**
JWSD-adj	0.287	0.158	0.257	**0.404**
JWSAN-2145	0.627	0.642	0.580	**0.701**
JWSAN-1400	0.499	0.499	0.416	**0.541**
NIKKEI	**0.934**	0.896	0.896	0.925

Each model was evaluated using the Japanese Word Similarity Dataset (JWSD) (Sakaizawa and Komachi, 2018), the Japanese Word Similarity and Relatedness Dataset (JWSAN) (Inohara and Utsumi, 2021), and the Nikkei corpus. JWSD is a dataset that assigns similarity values from 0 to 10 to words, and has four parts of speech: adjectives (JWSD-adv), verbs (JWSD-verb), nouns (JWSD-noun), and adverbs (JWSD-adj). JWSAN is a dataset of similarity and relatedness of nouns, verbs, and adjectives, with similarity and relatedness assigned values from 1 to 7, respectively. There are two datasets: one with all 2145 word pairs (JWSAN-2145) and the other with 1400 word pairs (JWSAN-1400) carefully selected for distributed representation. Spearman's rank correlation coefficient [5] was used as the evaluation metric.

In the task of NIKKEI, using the Nikkei corpus, genres were predicted from the keywords contained in the articles. Keywords are manually assigned by the editors, mainly nouns extracted from the article texts. The average of the word embeddings of each keyword was used as input. The genres were the same as described in Section 4.1. Accuracy was used as the evaluation metric. The Nikkei corpus from January 1, 2020 to November 30, 2021 was used for validation. In particular, the NIKKEI task showed the highest accuracy among the four models, suggesting that the created word2vec model was useful for the analysis of the Nikkei corpus.

B Temporary Table During Converting

Table 7 shows the temporary table during the conversion of the RoBERTa performance. We calculated the percentage of performance degradation by comparing to the newest year included in the training corpus.

C Visualization of the Relationship

Figures 6 and 7 show the visualization of the setting in which we train both word2vec and LightGBM. This setting showed a relatively undistinguished

[5] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.spearmanr.html

Table 7: Temporary table during converting in RoBERTa performance. The percentage of performance degradation is calculated by compared to the newest year included in the training corpus.

	Evaluation										
RoBERTa	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
2010	0.00 %	10.33 %	14.05 %	16.17 %	15.54 %	16.60 %	20.20 %	24.00 %	26.36 %	26.52 %	34.40 %
2010-2011		0.00 %	15.06%	22.21 %	22.89 %	24.58 %	35.03 %	39.75 %	44.22 %	46.32 %	55.77 %
2010-2012			0.00 %	11.28 %	15.91 %	18.08 %	31.26 %	36.88 %	43.48 %	46.55 %	53.35 %
2010-2013				0.00 %	8.59 %	13.44 %	27.09 %	32.52 %	39.21 %	43.63 %	50.62 %
2010-2014					0.00 %	7.96 %	22.32 %	27.67 %	35.94 %	41.40 %	48.57 %
2010-2015						0.00 %	16.28 %	22.24 %	31.36 %	36.09 %	43.04 %
2010-2016							0.00 %	8.87 %	17.94 %	21.80 %	30.67 %
2010-2017								0.00 %	11.94 %	17.29 %	25.82 %
2010-2018									0.00 %	7.41 %	17.56 %
2010-2019										0.00 %	12.11 %
2010-2020										0.00 %	3.92 %
2010-2021										0.00 %	0.89 %

trend compared to when only word2vec was trained. This may be because LightGBM was also trained on a corpus from a different time period, making it difficult to see the effect of word2vec. The correlation coefficient is -0.2611 and -0.1738, respectively.

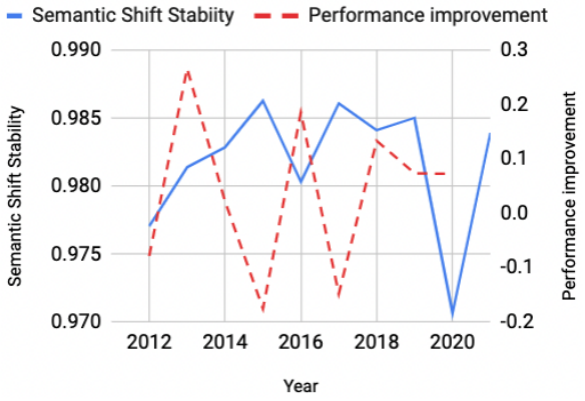

Figure 6: Relationship between Semantic Shift Stability and performance improvement difference of word2vec and LightGBM trained on the Nikkei corpus #. This setting showed a relatively undistinguished trend compared to when only word2vec was trained.

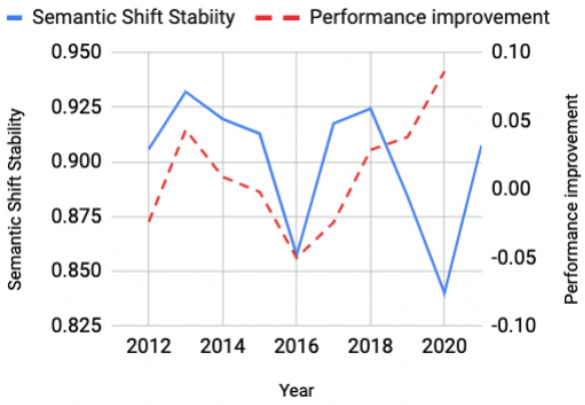

Figure 7: Relationship between Semantic Shift Stability and performance improvement difference of word2vec and LightGBM trained on the NOW corpus #. This setting showed a relatively undistinguished trend compared to when only word2vec was trained.