Investigating Effective Parameters for Fine-tuning of Word Embeddings Using Only a Small Corpus

Kanako Komiya
Ibaraki University
4-12-1 Nakanarusawa
Hitachi Ibaraki 316-8511 Japan
kanako.komiya.nlp@vc.ibaraki.ac.jp

Hiroyuki Shinnou
Ibaraki University
4-12-1 Nakanarusawa
Hitachi Ibaraki 316-8511 Japan
hiroyuki.shinnou.0828@vc.ibaraki.ac.jp

Abstract

Fine-tuning is a popular method to achieve better performance when only a small target corpus is available. However, it requires tuning of a number of meta-parameters and thus it might carry risk of adverse effect when inappropriate meta-parameters are used. Therefore, we investigate effective parameters for fine-tuning when only a small target corpus is available. In the current study, we target at improving Japanese word embeddings created from a huge corpus. First, we demonstrate that even the word embeddings created from the huge corpus are affected by domain shift. After that, we investigate effective parameters for fine-tuning of the word embeddings using a small target corpus. We used perplexity of a language model obtained from a Long Short-Term Memory network to assess the word embeddings input into the network. The experiments revealed that fine-tuning sometimes give adverse effect when only a small target corpus is used and batch size is the most important parameter for fine-tuning. In addition, we confirmed that effect of fine-tuning is higher when size of a target corpus was larger.

1 Introduction

We investigate effective parameters for fine-tuning using nwjc2vec. Nwjc2vec is Japanese word2vec (the word embeddings proposed by (Mikolov et al., 2013)) created from NINJAL Web Japanese Corpus (NWJC) (Asahara et al., 2014) (Asahara and Teruaki, 2017). It contains 25.8 billion words as a whole. Therefore, it is believed that nwjc2vec is high-quality. In fact, some models used it showed better results (Yamaki et al., 2017) (Shinnou et al., 2017b) (Shinnou et al., 2017a). In addition, it is also believed that nwjc2vec is useful for various documents because it contains a number of documents described about various topics.

However, we show that a problem posed by domain shift occurs when nwjc2vec is used in the current study. (See Section 4)

The simplest and most effective approach to address the problem caused from domain shift of word embeddings is fine-tuning using a large target corpus. However, in practice, we often face the situation where only a small corpus of the target domain is available. It is possible to use other resources than a corpus, but they are not always available. Therefore, in the current study, we investigate effective parameters for word2vec, which is a program to create word embeddings, when we fine-tune nwjc2vec using only a small target corpus. (See Section 5)

We evaluate the word embeddings via language models obtained from a LSTM (Long Short-Term Memory) (Hochreiter and Schmidhuber, 1997) (Gers et al., 2000) (Greff et al., 2016) (See Section 3). First, we develop a language model using a LSTM. Usually, word embeddings are learned from the same corpus as a training corpus for a language model. In other words, when we have only a small target corpus, we use the word embeddings learned from the target corpus for the inputs for the LSTM that develops a language model. However, we input nwjc2vec fine-tuned using the small corpus into the LSTM instead of the word embeddings directly learned from the corpus. We evaluate the language model to assess the fine-tuned word embeddings assuming that the quality of the output language model is higher when the quality of the word embeddings used in the LSTM is higher.

Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP, pages 60–67

The experiments revealed that the batch size is the most important parameter for word2vec to fine-tune nwjc2vec using a small corpus. In addition, they also showed that fine-tuning using inappropriate parameters sometimes make performance worse. Moreover, we confirmed that size of the corpus is crucial for fine-tuning. (See Sections 6 and 7)

2 Related Work

Generally, effectiveness of word embeddings depends on tasks and target domains of the tasks. Therefore, (Schnabel et al., 2015) proposed tuning of word embeddings according to tasks and their target domains.

The simplest tuning is fine-tuning, which is an approach where learned word embeddings are used for the initial values and tuned using an additional corpus. Its effectiveness has been shown for object recognition (Agrawal et al., 2014), named entity recognition (Lee et al., 2017), and many other tasks. Usually, a large target corpus is required for fine-tuning. Some works improved the word embeddings using external knowledges such as dictionaries. (Yu and Dredze, 2014) changed the loss function to use pre-knowledges and improved the word embeddings. (Faruqui et al., 2015) proposed to use retrofitting, which is an approach where the word embeddings obtained from a huge corpus are re-learned using external knowledges.

Fine-tuning is one of the methods for transfer learning (Pan and Yang, 2009). There are also much work about multi-task learning, which is another approach often used for transfer learning for neural networks (Aguilar et al., 2017) (von Däniken and Cieliebak, 2017).

3 Evaluation Method of Word Embeddings Using a LSTM

In the current study, we used a LSTM, which is an extended version of an a RNN to evaluate the word embeddings for a certain domain as (Shinnou et al., 2017a). We developed a language model using a LSTM from a training corpus and calculated the perplexity of the language model for a test corpus. Perplexity is given by the following equation.

$$PP = 2^H$$

where H is entropy given by the following equation.

$$H = \frac{1}{|D|}\sum_{i=1}^{|D|} -P(W_i|M)log_2P(W_i|M)$$

where D denotes a size of test data, M denotes a language model, and W_i denotes i_{th} word in the test data.

We evaluate the quality of the word embeddings depending on the perplexity assuming that the quality of the output language model is higher when the quality of the word embeddings used in the LSTM is higher. Usually, word embeddings are learned from the same corpus as the training corpus for a language model. However, we used the word embeddings to be evaluated instead of the word embeddings learned together with the language model (cf. Figure1). We believe that we can evaluate the quality of the word embeddings by evaluating the perplexity of the language model when they are used in a LSTM.

4 Effect of Domain Shift for Nwjc2vec

We demonstrate that even nwjc2vec, which is a word embeddings obtained from a huge corpus, NWJC, has a problem posed by domain shift in this section.

4.1 Mai2Vec

To show this problem, we firstly created word embeddings from newspapers collected for seven years: Mainichi Shimbun newspaper articles from 1993 to 1999. We removed headlines and tables and extracted only sentences. The sentences were divided into words and the words were used for inputs into word2vec. The corpus had 6,791,403 sentences. We used MeCab-0.996 as a morphological analyzer and UniDic-2.1.2 as a dictionary. These word embeddings are referred to as mai2vec. The word2vec parameters used for mai2vec are the same as the parameters used for nwjc2vec. The final number of tokens of mai2vec we obtained was 132,509.

4.2 Language Model for Blogs and Q & A sites

First, we compared mai2vec with nwjc2vec using blogs and Q & A sites for test data. We extracted 7,330 sentences from blogs (Yahoo! blogs) and Q & A sites (Yahoo! Chiebukuro) of Balanced

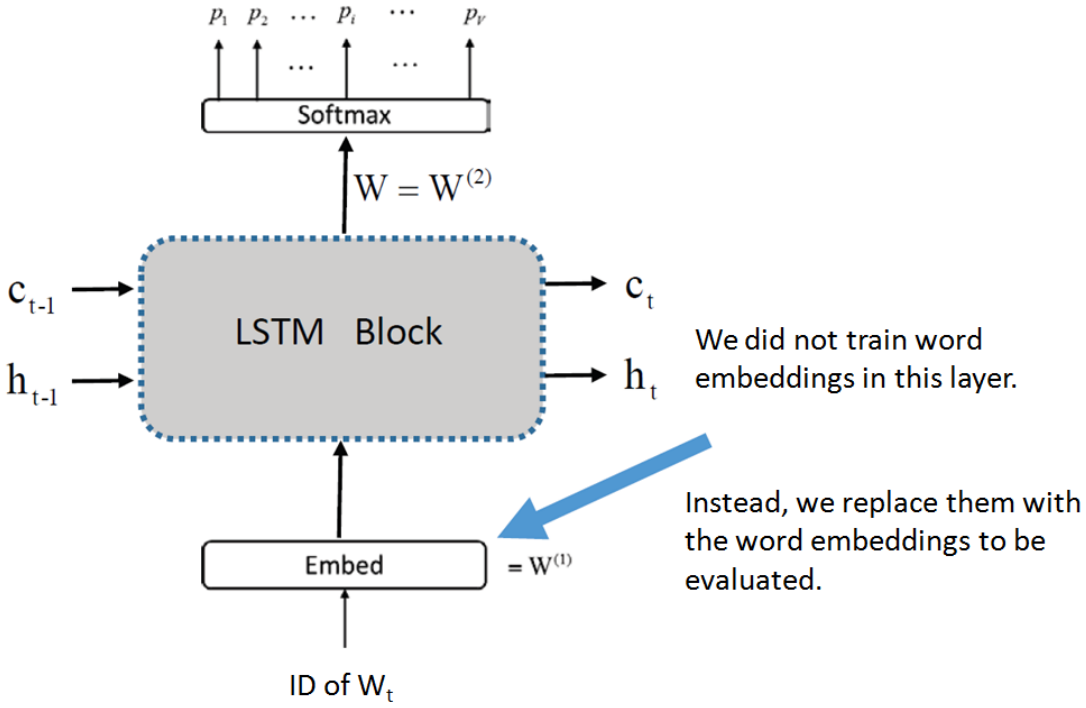

Figure 1: Evaluation Method of Word Embeddings Using LSTM

Corpora of Contemporary Witten Japanese (BCCWJ) (Maekawa et al., 2014) and used them for the language model. We used 7,226 sentences for the training and 104 sentences for the test. The language model that used nwjc2vec in the LSTM was referred to as nwjc2vec-lm-1 and the language model that used mai2vec in the LSTM was referred to as mai2vec-lm-1. Base-lm-1, which was a language model that used the word embeddings learned together with the language model in the LSTM, was also evaluated for reference. Table 1 shows the corpora used for this experiment.

Perplexity was used for the evaluation of the language models. We conducted learning 15 epochs, saved the models, and calculated their perplexities for each epoch. After that, we evaluated the lowest perplexity for each model[1].

Table 2 shows the results. According to the table, the perplexity of nwjc2vec-lm-1 is the lowest, which indicates that the quality of nwjc2vec is higher than that of mai2vec.

However, the domains of the training and test corpora for the language model, blogs and Q &A site, were different from that of mai2vec, Mainichi Shimbun Newspaper. Therefore, nwjc2vec perhaps had an advantage.

[1]The perplexity was the lowest at the fourth or fifth epoch for all the models.

4.3 Language Model for Newspaper

Next, we evaluate the word embeddings using the training and test corpora from newspapers, whose domain is the same as that of mai2vec. We used 100,000 sentences extracted from Mainichi Shimbun Newspaper in 2007 for the training of the language models. Ten thousand sentences extracted from Mainichi Shimbun Newspaper in 2008 were used for the test. Nwjc2vec-lm-2 and mai2vec-lm-2, which were the language models that used nwjc2vec and mai2vec respectively, were developed again. Base-lm-2, which was a language model that uses the word embeddings learned together with the language model in a LSTM, was also evaluated for reference. Note that the training corpora of word2vec for base-lm-1 and base-lm-2 are different. Table 3 shows the corpora used for this experiment.

Table 4 and Figure 2 show the results. They show that the perplexity of mai2vec-lm is the lowest, which indicates that the quality of mai2vec is higher than that of nwjc2vec.

The better method was shifted from nwjc2vec-lm into mai2vec-lm when the domain of the training and test corpora were the same as that of mai2vec. This fact indicates that even nwjc2vec has a problem posed by domain shift.

Table 1: Corpora Used for Word2vec and Training and Test Data for Language Model for Blogs and Q & A Sites

Name of model	Word2Vec corpus	Training data	Test data
mai2vec-lm-1	Newspaper in from 1993 to 1999	Blogs And Q & A sites	
nwjc2vec-lm-1	NWJC (Web pages)		
base-lm-1	Blogs and Q & A sites		

Table 2: Evaluation of Language Models Obtained from Each Word Embeddings 1

base-lm-1	mai2vec-lm-1	nwjc2vec-lm-1
130.35	124.72	**118.68**

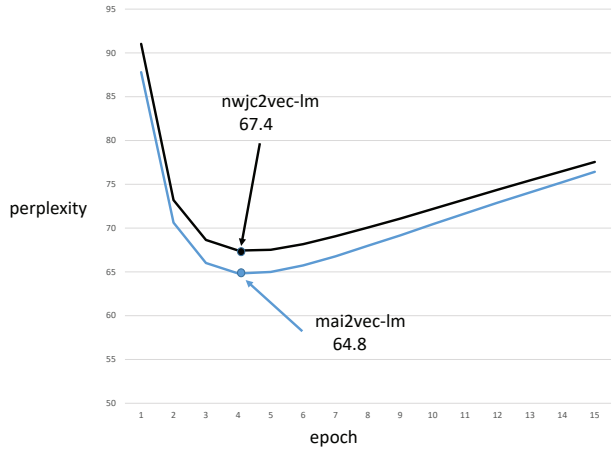

Figure 2: Perplexities of Language Models Developed Using Each Word Embeddings

Finally, Table 5 summarizes the number of sentences of each corpus. Please note that the corpora used for word2vec for base-lm-1 and base-lm-2 are the same as the corpora used for training the language models, respectively. The word embeddings were learned together with the LSTMs.

5 Fine-tuning Using a Small Corpus

Fine-tuning of nwjc2vec using a target corpus is the simplest way to address the problem caused from domain shift. It is preferable that the large target corpus is used for the fine-tuning but sometimes only a small target corpus is available. In these cases, it is not clear yet if the fine-tuning improves nwjc2vec.

Therefore, we tested various parameters of word2vec, which was a program to develop word embeddings, and found out if they were effective or not.

First, we set standard parameters of word2vec and fine-tuned nwjc2vec through them using an additional corpus (the small target corpus). Next, only a windows size parameter was changed from the standard one and fine-tuned nwjc2vec through them using the same additional corpus. We changed the batch size and epoch number parameters and fine-tuned nwjc2vec in the same way.

Table 6 lists the standard parameters of word2vec for the fine-tuning.

The procedures to investigate the parameters are described as follows. First, we fine-tuned nwjc2vec using the standard parameters listed in Table 6 and developed word embeddings. The word embeddins developed in this setting are referred to as base_emb. Next, we changed the window size parameter into 8 and fine-tuned nwjc2vec. The word embeddins developed in this setting are referred to as win_emb. After that, we changed only the batch size parameter into 20 remaining the other parameters as the standard ones and fine-tuned nwjc2vec. The word embeddins developed in this setting are referred to as batch20_emb. We also evaluated batch100_emb, which were word embeddings fine-tuned using the standard parameters except the batch size, which had been changed into 100. Finally, we evaluated epch_emb, which were word embeddings fine-tuned using the standard parameters except the epoch number, which had been changed into 20. Table 7 lists the parameters of word2vec we tried for the fine-tuning. We tested the five settings of fine-tuning including the setting in Table 6. We used 100,000 sentences randomly extracted from Mainichi Shimbun in from 1993 to 1999 as an additional corpus for the fine-tuning.

6 Experiments

We developed the language models through the LSTMs. We used the five fine-tuned word embeddings described above, base_emb, win_emb, batch20_emb, batch100_emb, and epch_emb, and used 100,000 sentences randomly extracted from Mainichi Shimbun Newspaper in from 1993 to

Table 3: Corpora Used for Word2vec and Training and Test Data for Language Model for Newspapers

Name of model	Word2Vec corpora	Training data	Test data
mai2vec-lm-2	Newspaper in from 1993 to 1999	Newspaper	Newspaper
nwjc2vec-lm-2	NWJC (Web pages)	In	In
base-lm-2	Newspaper in 2007	2007	2008

Table 4: Evaluation of Language Models Obtained from Each Word Embeddings 2

base-lm-2	mai2vec-lm-2	nwjc2vec-lm-2
81.52	**64.81**	67.47

1999 to train the LSTMs. We calculated perplexities of the language models obtained from the LSTMs at each epoch using the test data, which was 10,000 sentences from the same corpus as the training data. These is no overlap among the data for the fine-tuning, the training, and the testing. Table 8 summarizes the number of sentences of each corpus.

Table 9 and Figure 3 show the results. They include the perplexities of the language model obtained from the LSTMs when original nwjc2vec was used without the fine-tuning. The asterisks in the table mean that the language model using the fine-tuned word embeddings was better than that using nwjc2vec.

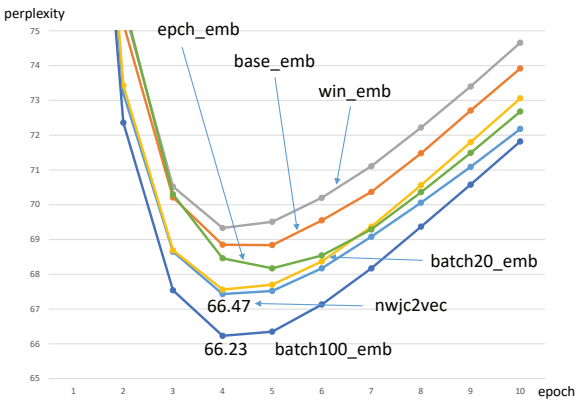

Figure 3: Changes of Perplexities According to Various Settings

These results show that the perplexities of the language model decrease only when batch100_emb is used. It indicates that fine-tuning is only effective when the batch size parameter is changed into 100. Other parameter changes made the results worse. The experiments revealed that fine-tuning has an opposite effect when unsuitable parameters are used in the case where small corpora are used.

7 Discussion

We think that although we might obtain better performance if we changed parameters other than batch size, the best results would be around the performance of batch100_emb because the batch size affected much more than the window size and the epoch number according to Table 9 and Figure 3.

In addition, we believe that the most important factor for the effective fine-tuning of nwjc2vec is the size of the additional corpus. To confirm this point, we tried some variation of the additional corpus size, 200,000 and 300,000 sentences in addition to the original setting, 100,000 sentences.

Table 10 and Figure 4 list the results of these experiments. These results indicate that the effect of fine-tuning is higher when the size of the additional corpus is larger.

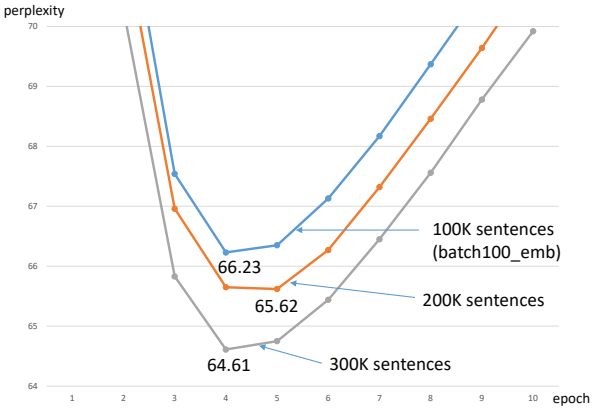

Figure 4: Changes of Perplexities According to Various Sizes of Additional Corpora

The fine-tuning approach we employed is the simplest way to tune word embeddings. Fine tuning of nwjc2vec requires large-sized additional corpus. Instead of the additional corpus, the external resources such as dictionaries would be useful. We plan to improve nwjc2vec using such external resources in the future.

Table 5: Corpus Data for Domain Shift Experiments

Corpus	Type	Aim	Genre	Number of Sentences
NWJC	Training	Nwjc2vec	Web pages	1,463,142,939
Mainichi Shimbun 1993-1999	Training	Mai2vec	Newspaper	6,791,403
BCCWJ	Training	Word2vec of base-lm-1	Blogs	7,226
		Language model	And	
BCCWJ	Test	For blogs and Q &A sites	Q & A sites	104
Mainichi Shimbun 2007	Training	Word2vec of base-lm-2		100,000
		Language model	Newspaper	
Mainichi Shimbun 2008	Test	For newspaper		10,000

Table 6: Standard Parameters for Word2vec

Model Name	base_emb
Number of Units	200
Window Size	5
Batch Size	10
Epoch Number	10
Used Model	skip-gram

8 Conclusion

We showed the problem occurred by domain shift when nwjc2vec was used and investigated the effective parameters of word2vec to fine-tune nwjc2vec using a small corpus.

The experiments revealed that it is possible to obtain better results using fine-tuning of nwjc2vec if we properly adjust parameters. We showed that the most effective parameter of the fine-tuning is the batch size and fine-tuning using improper parameters make the results worse. Finally, we demonstrated that the size of the additional corpus is crucial for fine-tuning of nwjc2vec. We plan to use external resources instead of the large-sized corpus in the future.

References

Pulkit Agrawal, Ross Girshick, and Jitendra Malik. 2014. Analyzing the Performance of Multilayer Neural Networks for Object Recognition. In *Proceedings of ECCV-2014*, page arXiv:1407.1610.

Gustavo Aguilar, Suraj Maharjan, A. Pastor Lopez-Monroy, and Thamar Solorio. 2017. A Multi-task Approach for Named Entity Recognition in Social Media Data. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 148–153.

Masayuki Asahara, Kikuo Maekawa, Mizuho Imada, Sachi Kato, and Hikari Konishi. 2014. Archiving and Analysing Techniques of the Ultra-large-scale Web-based Corpus Project of NINJAL, Japan. *Alexandria: The Journal of National and International Library and Information Issues*, 25(1-2):129–148.

Masayuki Asahara and Oka Teruaki. 2017. nwjc2vec: word embedding data based on NINJAL Japanese Web Corpus (in Japanese). In *ANLP-2017*, pages 94–97.

Pius von Däniken and Mark Cieliebak. 2017. Transfer Learning and Sentence Level Features for Named Entity Recognition on Tweets. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 166–171.

Manaal Faruqui, Jesse Dodge, Sujay K. Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting Word Vectors to Semantic Lexicons. In *Proceedings of NAACL*.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 2000. Learning to forget: Continual prediction with LSTM. *Neural computation*, 12(10):2451–2471.

Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. 2016. LSTM: A Search Space Odyssey. *IEEE transactions on neural networks and learning systems*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Ji Young Lee, Franck Dernoncourt, and Peter Szolovits. 2017. Transfer Learning for Named-Entity Recognition with Neural Networks. In *arXiv*, page arXiv:1705.06273.

Kikuo Maekawa, Makoto Yamazaki, Toshinobu Ogiso, Takehiko Maruyama, Hideki Ogura, Wakako Kashino, Hanae Koiso, Masaya Yamaguchi, Makiro Tanaka, and Yasuharu Den. 2014. Balanced corpus of contemporary written japanese. *Language Resources and Evaluation*, 48(2):345–371.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Table 7: Standard Parameters for Word2vec

Model Names	win_emb	batch20_emb	batch100_emb	epoch_emb
Number of Units	200			
Window Size	8	5	5	5
Batch Size	10	20	100	10
Epoch Number	10	10	10	20
Used Model	skip-gram			

Table 8: Corpus Data for Fine-tuning Experiments

Corpus	Type	Aim	Genre	Number of Sentences
Mainichi Shimbun 1993-1999	Fine-tuning	Word2vec	Newspaper	100,000
	Training	Language Model		100,000
	Test			10,000

Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 22(10):1345 – 1359.

Tobias Schnabel, Igor Labutov, David M Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *EMNLP 2015*, pages 298–307.

Hiroyuki Shinnou, Masayuki Asahara, Kanako Komiya, and Minoru Sasaki. 2017a. nwjc2vec: Word Embedding Data Constructed from NINJAL Web Japanese Corpus. *Journal of Natural Language Processing*, 24(5):705–720.

Hiroyuki Shinnou, Kanako Komiya, and Minoru Sasaki. 2017b. Supervised Word Sense Disambiguation using Forward Multilayered LSTM and Word Embeddings (in Japanese). In *IPSJ Natural Language Processing Report*, pages NL–232–4.

Shoma Yamaki, Hiroyuki Shinnou, Kanako Komiya, and Minoru Sasaki. 2017. Construction of word embeddings using labeled data (in Japanese). In *ANLP-2017*, pages 78–81.

Mo Yu and Mark Dredze. 2014. Improving Lexical Embeddings with Semantic Knowledge. In *ACL (2)*, pages 545–550.

Table 9: Perplexities of Various Settings

epoch	nwjc2vec	base_emb	win_emb	batch20_emb	batch100_emb	epch_emb
1	91.03	93.70	95.36	91.51	89.69	95.06
2	73.20	75.21	75.71	73.43	72.36	75.89
3	68.65	70.21	70.52	68.69	67.54	70.30
4	**67.43**	68.85	**69.33**	**67.56**	**66.23***	68.46
5	67.52	**68.84**	69.51	67.70	66.35*	**68.17**
6	68.17	69.55	70.20	68.37	67.13*	68.54
7	69.08	70.37	71.11	69.37	68.17	69.29
8	70.06	71.48	72.22	70.56	69.37	70.36
9	71.09	72.71	73.40	71.80	70.58	71.49
10	72.18	73.92	74.66	73.06	71.82	72.68

Table 10: Perplexities of Sizes of Additional Corpora

epoch	100 thousand sentences (batch100_emb)	200 thousand sentences	300 thousand sentences
1	89.69	89.55	87.94
2	72.36	71.50	70.28
3	67.54	66.96	65.83
4	**66.23**	65.65	**64.61**
5	66.35	**65.62**	64.75
6	67.13	66.27	65.44
7	68.17	67.32	66.45
8	69.37	68.46	67.56
9	70.58	69.64	68.78
10	71.82	70.89	69.92