Deep Learning for Large-Scale Sentiment Analysis Using Distributed Representations

Kazuhei Katoh
Department of Computer Science
Faculty of Engineering
Ehime University
Matsuyama, Ehime, Japan
Email: katoh@ai.cs.ehime-u.ac.jp

Takashi Ninomiya
Department of Electrical and Electronic Engineering
and Computer Science
Graduate School of Science and Engineering
Ehime University
Matsuyama, Ehime, Japan
Email: ninomiya@cs.ehime-u.ac.jp

***Abstract*—This paper presents the performance evaluations of deep learning classifiers for large-scale sentiment analysis using Rakuten Data. Many NLP theories and applications use 1-of-K representations for representing a word, but 1-of-K representations are difficult to use with many deep learners because they are vectors consisting of millions of dimensions. To reduce the number of dimensions of 1-of-K representations, we used distributed representations for words by using word2vec. Two experiments were conducted: (1) sentiment analysis using a small data set, the IMDB dataset, and (2) sentiment analysis using a large-scale data set, Rakuten Data. In the experiments, we observed that multi-layer neural networks did not work well for the small data set (i.e., neural networks without hidden layers achieved the best result), but multi-layer neural networks worked well for the large-scale data set. In the experiments using Rakuten Data, we tested the neural networks with 0−6 hidden layers, and neural networks with three hidden layers achieved the best result.**

Keywords–sentiment analysis; deep learning; distributed representations.

I. INTRODUCTION

For the last decade, many kinds of social media on the Internet, such as Twitter, SNS, and blogs, have become available, and more than a billion people use them in their daily life now. As these social media grow, sentiment analysis from these media becomes more important to extract opinions about political issues, events, and some specific commercial products.

In this paper, we present performance evaluations of deep learning classifiers for a large-scale sentiment analysis using Rakuten Data. Deep learning has attracted many researchers because it has achieved significant results in the fields of speech recognition and image recognition [1][2]. The models of deep learning classifiers are defined as multi-layer neural networks having around 3 to 20 hidden layers. For a deep learner, generalized features or concepts are automatically acquired in hidden layers along with the training of whole networks. Though some deep learning tools have already been developed publicly, these tools assume dense and low dimensional vectors as inputs. This is a crucial problem for large-scale natural language processing (NLP) because many NLP theories and applications use 1-of-K representations for representing a word, which has millions of dimensions. To mitigate the problem of 1-of-K representations, we use word2vec [3] to reduce the number of dimensions of 1-of-K representations.

We conducted two experiments on sentiment analysis. In the first experiment, we used Rakuten Data [4] as a data set, and we used the IMDB dataset [5] in the second experiment. Rakuten Data is a large-scale data set, consisting of 16 million reviews, that is written in Japanese. The IMDB dataset consists of 100,000 movie reviews written in English. The IMDB dataset is smaller than Rakuten Data, and hence the deep learner can learn from the IMDB dataset using 1-of-K representations. In the experiments, we evaluated the effectiveness of word2vec by comparing it with 1-of-K representations in the IMDB dataset. Finally, we evaluated the effectiveness of our classifier for the large-scale sentiment analysis using Rakuten Data. In the experiments, we observed that multi-layer neural networks did not work well for the small data set (i.e., neural networks without hidden layers achieved the best result for the IMDB dataset), but multi-layer neural networks worked well for the large-scale data set, Rakuten Data.

Section Ⅱ introduces an overview of related work. Section Ⅲ presents large-scale sentiment analysis based on multi-layer neural networks and distributed representations for words using word2vec. Section Ⅳ describes the procedures and results of the experiments. Section V concludes the document.

II. RELATED WORK

In NLP, 1-of-K representations (or one-hot representations) are generally used for representing a word as a vector. The word vector in 1-of-K representations has the same length of vocabulary size, and each dimension corresponds to each word in a dictionary. The vector for word w in 1-of-K representations takes a form in which only one dimension corresponding to w is given as 1, and other dimensions are given as 0. For example, if we have a dictionary consisting of one hundred thousand words, the word vector takes a form in which only one dimension is given as 1 among one hundred thousand dimensions, and all the remaining dimensions are given as 0. Let n be the vocabulary size and e be the index of dimensions for word w_e. The vector for word w_e in 1-of-K representations is formally given as $(d_1, ..., d_i, ..., d_n)$, where $d_i = 1$ if $i = e$, and $d_i = 0$ otherwise. Therefore, word vectors in 1-of-K representations are large and extremely sparse.

Word2vec [3] is a method for obtaining distributed representations for words by using neural networks. In word2vec,

 ISBN: 978-1-61208-420-6

neural networks are defined to solve a pseudo task of predicting a word given surrounding words. After training neural networks using huge size text, weight vectors for each word in a dictionary are retrieved from the neural networks as distributed representations for words.

Two types of neural network models are defined in word2vec: the continuous bag-of-words (CBOW) model and the Skip-gram model. The CBOW model predicts the current word given surrounding words, and the Skip-gram model predicts surrounding words given the current word. Both models generate distributed representations for words by retrieving a weight matrix from neural networks between the input layer and the hidden layer. Therefore, the number of dimensions of the weight vectors is equal to the number of nodes in the hidden layer, around 200 or 400. Thus, word2vec reduces the number of dimensions of word vectors in 1-of-K representations having hundreds of thousands of words in the input layer to hundreds of dimensions. Mikolov et al. [3] have shown that the acquired word vector represents a semantic concept for a word and relationships between words. The relationships between words can be calculated using simple addition and subtraction, e.g., the vector for 'queen' is close to the vector for 'king' minus 'man' plus 'woman.'

Deep learning involves multi-layer neural networks with efficient learning methods and generalization. In our experiments, we used Caffe [6] as a deep learning tool. Caffe is an efficient implementation, one in which neural network models can be customized in a model file. However, Caffe cannot deal efficiently with large-scale data, such as Rakuten Data, in 1-of-K representations due to memory limitation. Therefore, we reduced the number of dimensions of the dataset by using word2vec.

III. LARGE-SCALE SENTIMENT ANALYSIS BY USING DEEP LEARNING AND DISTRIBUTED REPRESENTATIONS

We present large-scale sentiment analysis based on multi-layer neural networks and distributed representations for words using word2vec. In the experiments, we tested multi-layer neural networks with $0-6$ hidden layers. In the input layer, a document vector $\mathbf{d}$ was used as an input. The document vector is a vector for a review made by adding distributed representations for all words in the review. The distributed representations for words were acquired by using word2vec trained from Rakuten Data. Formally, we have the document vector $\mathbf{d}$ as follows (1).

$$\mathbf{d} = \sum_{w \in r} word2vec(w), \tag{1}$$

where r is a review, w is a word in r, and $word2vec(w)$ is the distributed representation for word w. In the output layer, binary sentiments (positive/negative) were used as an output, and softmax functions were applied to the output from the hidden layers. Figure 1 shows the structures of neural networks that we used in the experiments. In the figure, (A), (B), and (C) draw neural networks with 0, 1, and 2 hidden layers, respectively. Neural networks with 0 hidden layers are equivalent to the logistic regression model without a prior.

TABLE I. RAKUTEN DATA

	data size (GB)	number of reviews	number of words
training set	8.45	13,133,032	656,834,594
development set	1.02	1,655,042	82,123,546
test set	1.12	1,818,107	88,549,699

TABLE II. IMDB DATASET

	data size (MB)	number of reviews	number of words
training set (labeled)	33.158	25,000	5,843,019
training set (unlabeled)	66.557	50,000	14,273,230
test set (labeled)	32.376	25,000	5,711,718

IV. EXPERIMENTS

We conducted experiments to evaluate the performance of the multi-layer neural networks for large-scale sentiment analysis using Rakuten Data.

A. Dataset and Tools

We used two datasets, Rakuten Data [4] and IMDB dataset [5]. Rakuten Data is a large-scale data set consisting of around 16 million reviews written in Japanese[1]. Each review in Rakuten Data is labeled with $0-5$ grade labels: 0 is the most negative, and 5 is the most positive. We converted the $0-5$ grade sentiments into binary sentiments by regarding $0-3$ grades as negative and $4-5$ grades as positive. In the experiments, we evaluated the binary classification task. Table I shows the specifications of Rakuten Data. The IMDB dataset consists of 100,000 movie reviews written in English. In the IMDB dataset, 50,000 reviews are labeled with $1-10$ grade labels: 1 is the most negative, and 10 is the most positive. We converted the $1-10$ grade sentiments into binary sentiments by regarding $1-5$ grades as negative and $6-10$ grades as positive. Table II shows the specifications of the IMDB dataset. The IMDB dataset is smaller than Rakuten Data, and hence the deep learner can learn from the IMDB dataset using 1-of-K representations. We evaluated the effectiveness of the deep learning classifier for the large-scale sentiment analysis using Rakuten Data. We also evaluated the effectiveness of word2vec by comparing it with 1-of-K representations in the IMDB dataset, where we have the document vector $\mathbf{d}$ for 1-of-K representations as follows (2).

$$\mathbf{d} = \sum_{w \in r} \text{1-of-}K(w), \tag{2}$$

where r is a review, w is a word in r, and 1-of-$K(w)$ is the 1-of-K representation for word w.

The number of dimensions for the distributed representations was determined using the development set. We tested 100, 200, 400, and 800 dimensions for Rakuten Data, and 100, 200, 400, 800, and 1600 dimensions for the IMDB dataset. Table III shows the details of other hyper parameters that were determined by using the development data set.

[1]Currently, Rakuten Data 2014 consists of around 64 million reviews for 150 million products. We used Rakuten Data 2010 in the experiments, and it consists of around 16 million reviews.

 ISBN: 978-1-61208-420-6

TABLE III. HYPERPARAMETERS OF WORD2VEC.

hyperparameters	setting
Model	CBOW
Window Size	8
Negative Samples	25
Hierarchical Softmax	none
Iteration	15
Subsampling of Frequent Words	1e-3

TABLE IV. HYPERPARAMETERS OF CAFFE.

hyperparameters	setting
Hidden layer	0−6
The number of nodes	500
Test interval	1,000
Max iteration	100,000

We used Caffe as a deep learning tool. However, Caffe cannot efficiently learn from Rakuten Data using 1-of-K representations because it is a large data set and because Caffe does not support sparse vectors. We first trained word2vec using 13,133,032 reviews (656,834,594 words) in Rakuten Data, and then we trained Caffe using 2,684,354 reviews (137,456,326 words) in Rakuten Data. Table IV shows the hyper parameters for Caffe. The batch size was 200 for Rakuten Data and 1000 for the IMDB dataset. The base learning rate was 0.01 for Rakuten Data and 0.005 for the IMDB dataset. Figure 1 shows the structures of the multi-layer neural networks.

We also compared the performance of deep learning with L2-regularized logistic regression. We used Liblinear [7] for evaluating L2-regularized logistic regression. The hyperparameters were tuned by using the development data.

We used Mecab [8] for tokenizing Rakuten data and used Stepp Tagger [9] for tokenizing the IMDB dataset.

B. Results

In the experiments, "LR(1-of-K)" means the result of L2-regularized logistic regression using 1-of-K representations. "LR(w2v)" means the result of L2-regularized logistic regression using distributed representations for words. "NN-Li(1-of-K)" means multi-layer neural networks with i hidden layers using 1-of-K representations. "NN-Li(w2v)" means multi-layer neural networks with i hidden layers using distributed representations for words.

Table V shows the results of the experiments for the test set of Rakuten Data. In the table, neural networks with three hidden layers (NN-L3(w2v)) achieved the best result for Rakuten Data. We can also see that NN-L3(w2v) achieved a better result than that of logistic regression, LR(w2v). In the table, we can observe that the accuracy increased when we used more hidden layers such that the number of hidden layers was less than four, and the accuracy decreased when we used more than four hidden layers.

Table VIII shows the results of the experiments for the test set of the IMDB dataset. In the table, we can see the difference in the neural networks using 1-of-K representations and those using distributed representations. The best result was achieved by the neural networks without hidden layers using the distributed representations. Contrary to our expectation, the multi-layer neural networks using 1-of-K representations were worse than those using the distributed representations.

TABLE V. ACCURACY FOR TEST DATASET OF RAKUTEN DATA.

Model	Accuracy
NN-L0(w2v)	89.130 %
NN-L1(w2v)	90.220 %
NN-L2(w2v)	90.703 %
NN-L3(w2v)	91.015 %
NN-L4(w2v)	91.001 %
NN-L5(w2v)	90.795 %
NN-L6(w2v)	90.727 %
LR(1-of-K)	90.956 %
LR(w2v)	90.124 %

Table VI and Figure 2 show the analyses for the development set of Rakuten Data, and Table VII and Figure 3 show the analyses for the development set of the IMDB dataset.

C. Discussion

We can see from Table VIII that neural networks with a higher number of hidden layers did not work well for the IMDB dataset, especially in the case of 1-of-K representations. We think that the multi-layer neural networks with 1-of-K representations failed to learn the concepts of words in their hidden layers. This may be because the size of the IMDB dataset was too small to learn them. In the case of training with Rakuten Data, the neural networks with three hidden layers achieved better results than those of the neural networks without hidden layers. We think that these results partially support our hypothesis that extremely large datasets, such as Rakuten Data, enable neural networks to learn their hidden layers well in the task of sentiment analysis.

With these experimental results, we think that in tasks of natural language processing, unlike image recognition or speech recognition, extremely large datasets are needed to learn the concepts of words or phrases in the hidden layers of multi-layer neural networks. In the experiments of the IMDB dataset, we used around 20 million words (75,000 reviews) for word2vec training. But, the data size of the IMDB dataset was much smaller than that of Rakuten Data, which consists of around 650 million words (around 13 million reviews). We think that pre-training of neural networks using an extremely large dataset is a good solution for simultaneously learning the word concepts and the tasks of NLP, such as multi-task learning [10] or a stacked auto-encoder [11].

From Table V and VIII, the accuracy of LR(1-of-K) was better than that of LR(w2v) in both experiments. However, the accuracy of NN-Li(1-of-K) was worse than that of NN-Li(w2v) in the experiment of the LDBM dataset. We think that this also means that neural networks with 1-of-K representations fail to learn the hidden layers. The tendency of how hidden layers are learned from 1-of-K representations can be seen more clearly if we could conduct experiments on the multi-layer neural networks with 1-of-K representations for Rakuten Data. We leave this for future work.

V. CONCLUSION

In this paper, we presented performance evaluations of deep learning classifiers for large-scale sentiment analysis using Rakuten Data. Many NLP theories and applications use 1-of-K

 ISBN: 978-1-61208-420-6

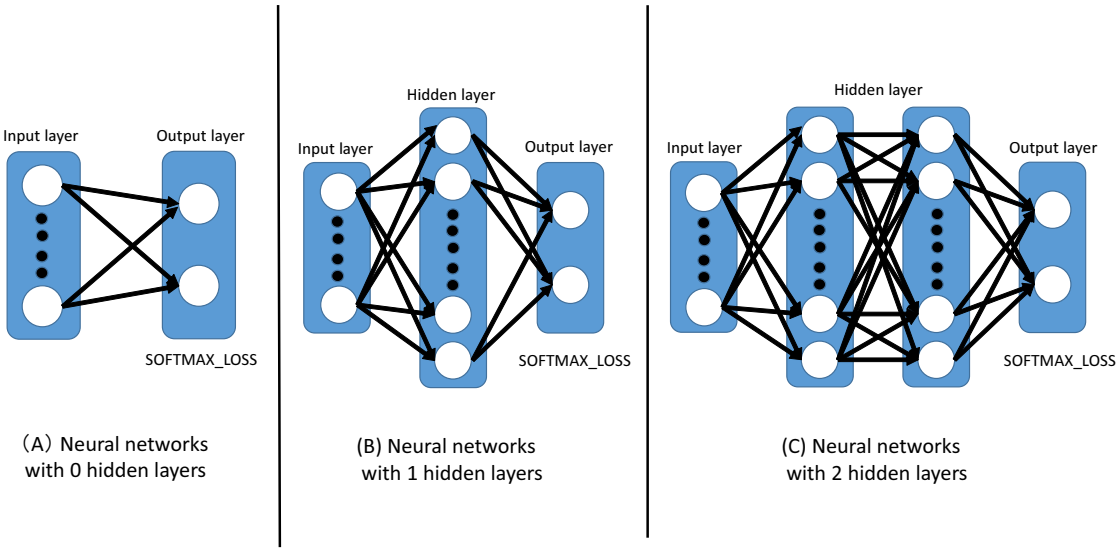

Figure 1. Structure of neural network

TABLE VI. ACCURACY FOR DEVELOPMENT DATASET OF RAKUTEN DATA.

Representation	Number of dimensions	NN-L0	NN-L1	NN-L2	NN-L3	NN-L4	NN-L5	NN-L6	LR
w2v	100	87.978 %	89.363 %	89.962 %	90.119 %	90.135 %	90.149 %	90.088 %	88.949 %
	200	88.111 %	89.947 %	90.478 %	90.551 %	90.595 %	90.538 %	90.541 %	89.392 %
	400	88.689 %	90.257 %	90.746 %	90.840 %	90.860 %	90.805 %	90.789 %	89.633 %
	800	88.110 %	90.434 %	90.847 %	90.947 %	90.982 %	90.957 %	90.900 %	89.878 %
1-of-K	81,420	-	-	-	-	-	-	-	90.671 %

TABLE VII. ACCURACY FOR DEVELOPMENT DATASET OF IMDB DATASET.

Representation	Number of dimensions	NN-L0	NN-L1	NN-L2	NN-L3	NN-L4	NN-L5	NN-L6	LR
w2v	100	87.180 %	50.200 %	52.780 %	86.041 %	86.460 %	86.820 %	87.201 %	87.380 %
	200	87.761 %	86.520 %	57.820 %	86.920 %	86.780 %	87.080 %	87.840 %	88.100 %
	400	88.140 %	53.720 %	54.800 %	86.940 %	87.760 %	88.081 %	88.200 %	88.620 %
	800	88.780 %	50.580 %	52.200 %	88.260 %	87.520 %	88.240 %	88.300 %	89.000 %
	1600	88.599 %	60.460 %	50.000 %	87.620 %	87.600 %	87.920 %	88.300 %	89.040 %
1-of-K	35,309	88.780 %	86.520 %	57.820 %	88.260 %	87.760 %	88.240 %	88.300 %	89.040 %

representations for representing a word, but 1-of-K representations are difficult to use with many deep learners because they are vectors consisting of millions of dimensions. To reduce the number of dimensions of 1-of-K representations, we used distributed representations for words by using word2vec.

We conducted two experiments: (1) sentiment analysis using a small data set, the IMDB dataset, and (2) sentiment analysis using a large-scale data set, Rakuten Data. In the experiments, we observed that multi-layer neural networks did not work well for the small data set (i.e., neural networks without hidden layers achieved the best result), but multi-layer neural networks worked well for the large-scale data set. In the experiments using Rakuten Data, we tested the neural networks with $0-6$ hidden layers, and neural networks with three hidden layers achieved the best result. We think that these results partially support our hypothesis that extremely large datasets, such as Rakuten Data, enable neural networks to learn their hidden layers well in the task of sentiment analysis.

In the experiments for the IMDB dataset, we also compared 1-of-K representations with distributed representations. In the experiments, neural networks using distributed representations achieved better results than those using 1-of-K representations. We think that this may be because the size of the IMDB dataset was too small to learn the concepts of words in the neural networks. The tendency of how multi-layer neural networks are learned from 1-of-K representations can be seen more clearly if we could conduct experiments on the multi-layer neural networks with 1-of-K representations for Rakuten Data. We leave this for future work.

ACKNOWLEDGMENT

This work was supported by a JSPS KAKENHI Grant-in-Aid for Scientific Research (B) Grant Number 25280084.

 ISBN: 978-1-61208-420-6

TABLE VIII. ACCURACY FOR TEST DATASET OF IMDB DATASET.

Model	Accuracy
NN-L0(1-of-K)	85.040 %
NN-L1(1-of-K)	83.540 %
NN-L2(1-of-K)	76.840 %
NN-L3(1-of-K)	50.720 %
NN-L4(1-of-K)	50.000 %
NN-L5(1-of-K)	50.000 %
NN-L6(1-of-K)	50.000 %
NN-L0(w2v)	88.476 %
NN-L1(w2v)	50.920 %
NN-L2(w2v)	54.908 %
NN-L3(w2v)	86.504 %
NN-L4(w2v)	86.940 %
NN-L5(w2v)	86.984 %
NN-L6(w2v)	87.344 %
LR(1-of-K)	86.848 %
LR(w2v)	78.936 %

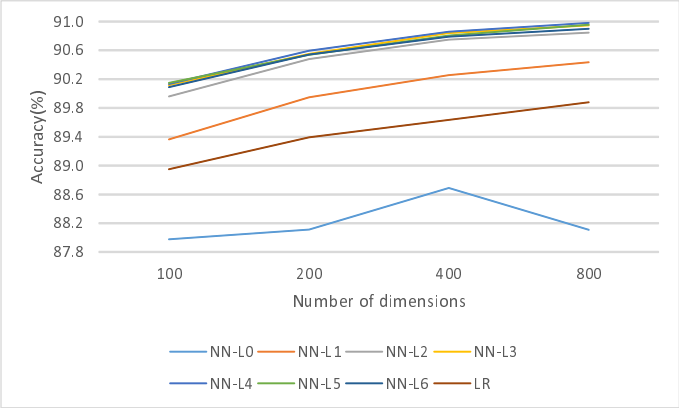

Figure 2. Accuracy of each model for development dataset of Rakuten Data

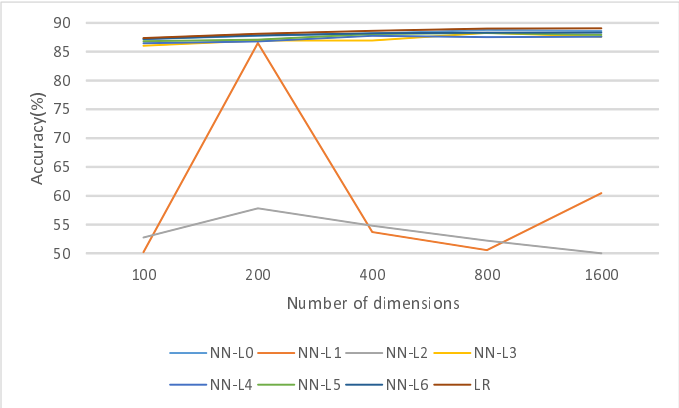

Figure 3. Accuracy of each model for development dataset of IMDB dataset

REFERENCES

[1] H. Kaiming, Z. Xiangyu, R. Shaoqing, and S. Jian, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," 2015. [Online]. Available: http://arxiv.org/abs/1502.01852 retrived:05, 2015

[2] S. Christian, W. Liu, J. Yangqing, S. Pierre, R. Scott, A. Dragomir, E. Dumitru, V. Vincent, and R. Andrew, "Going deeper with convolutions," 2014. [Online]. Available: http://arxiv.org/abs/1409.4842 retrived:05, 2015

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013. [Online]. Available: http://arxiv.org/abs/1301.3781 retrived:05, 2015

[4] "Rakuten dataset," http://rit.rakuten.co.jp/opendataj.html retrived:05, 2015.

[5] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 142–150. [Online]. Available: http://www.aclweb.org/anthology/P11-1015 retrived:05, 2015

[6] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," Proceedings of the ACM International Conference on Multimedia, 2014, pp. 675–678.

[7] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," Journal of Machine Learning Research Volume 9, 2008, pp. 1871–1874.

[8] T. Kudo, K. Yamamoto, and Y. Matsumoto, "Applying conditional random fields to japanese morphological analysis, proceedings of the 2004 conference on empirical methods in natural language processing," EMNLP-2004, 2004, pp. 230–237.

[9] Y. Tsuruoka, Y. Tateishi, J.-D. Kim, T. Ohta, J. McNaught, S. Ananiadou, and J. Tsujii, "Developing a robust part-of-speech tagger for biomedical text," in Advances in Informatics, ser. Lecture Notes in Computer Science, P. Bozanis and E. Houstis, Eds. Springer Berlin Heidelberg, 2005, vol. 3746, pp. 382–392. [Online]. Available: http://link.springer.com/chapter/10.1007retrived:05, 2015

[10] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," ICML '08 Proceedings of the 25th international conference on Machine learning, 2008, pp. 160–167.

[11] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," Journal of Machine Learning Research Volume 11, 2010, pp. 3371–3408.

 ISBN: 978-1-61208-420-6