Enhanced Word Embeddings from a Hierarchical Neural Language Model

Xun Wang, Katsuhito Sudoh and Masaaki Nagata
NTT Communication Science Laboratories, Kyoto, 619-0237, Japan
wang.xun, sudoh.katsuhito, nagata.masaaki@lab.ntt.co.jp

ABSTRACT

This paper proposes a neural language model to capture the interaction of text units of different levels, i.e.., documents, paragraphs, sentences, words in an hierarchical structure. At each paralleled level, the model incorporates Markov property while each higher-level unit hierarchically influences its containing units. Such an architecture enables the learned word embeddings to encode both global and local information. We evaluate the learned word embeddings and experiments demonstrate the effectiveness of our model.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing – abstracting methods; I.2.7 [**Computing Methodologies**]: Artificial Intelligence – Natural Language Processing

General Terms

Algorithms; Experimentation

Keywords

neural network; hierarchical model; distributed representations; word embeddings

1. INTRODUCTION

Word embeddings represent words using real-valued, abstract and condensed vectors. There are two main families for learning embeddings for words: The first family leverages document-level word-occurrence statistics, such as LDA [Blei et al., 2003], GloVe [JeffreyPennington and Manning, 2014], or matrix factorization based approaches (e.g., LSA and SVD) given the intuition that co-occurrent words are relevant. Those global co-occurrence statistics based models neglect word order information about how local meanings are formed by neighboring words. The second category corresponds to local context window approaches (e.g.,[Bengio et al., 2006, Collobert and Weston, 2008, Mikolov et al., 2013a]). The downside of such models is that they poorly harness the global information at document, paragraph or sentence level. Some attempts try to bridge the gap between the two families: [Huang et al., 2012] proposes a document-level vector leveraged from tf-idf into local learning process and *paragraph vector* [Le and Mikolov, 2014] makes word prediction with the help of the leveraged document/paragraph/sentence level information. [Li et al., 2015b] explores hierarchical autoencoder for paragraph and document representations. Their efforts prove useful for learning sentence, paragraph and document representations. But they do not consider using the relations between different levels to improve the word embeddings which are the basis of all. Towards better word embeddings, we consider the intrinsic structure of text about how units are arranged to form meaningful context:

(1) **Horizontally**: as we look at discourse theory in early days (Mann and Thompson (1988)), in a coherent text, not only words, but clauses, sentences, and larger multi-clause groupings are tightly connected. Text units take their respective roles and interact with units at the same level (token-to-token, sentence-to-sentence and paragraph to paragraph) semantically, syntactically, and logically.

(2) **Vertically**: Words form the meanings of sentences; sentences form paragraphs, and then paragraphs form documents, which organizes the arrangements into a tree structure vertically.

The importance of tree structures in sentence, paragraph and document representation has been revealed by previous research [Socher et al., 2013, Li et al., 2014, Li et al., 2015a]. Here we show how they can be used to improve the word embeddings. The proposed model captures the two aforementioned aspects of meanings in a unified embedding learning framework which holds promise to bridge the gap between the co-occurrence based and prediction based embedding learning frameworks. Horizontally, we model each level of units based on the Markovian manner, where neighbouring unites are correlated based on the similar assumption we make in language model. Vertically, each unit (e.g., sentence) exerts its impact on its containing lower-level text units (e.g., words). Unlike [Huang et al., 2012] where document-level information is harshly incorporated, the proposed approach gently incorporates the order information at paragraph level and sentence level, and therefore preserves the semantic integrity of the contexts.

The adopted type of architecture arranges all text units in a unified structure, where influence of one unit is propagated to others (the siblings and children), naturally bridging the gap of and taking the merits from the aforementioned two learning families. To note, our approach is inspired by paragraph vector model [Le and Mikolov, 2014] where models paragraph and tokens within it in a two-level hierarchy where words are predicted given neighbours and its resided paragraph.

The proposed architecture is ultimately grounded on the lowest level of the hierarchical structure, words, by predicting the current token, where the embeddings for neighboring tokens, and higher levels text units are simultaneously updated. The proposed algo-

CIKM'15, October 19–23, 2015, Melbourne, Australia.
© 2015 ACM. ISBN 978-1-4503-3794-6/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2806416.2806637.

rithm is a general one and can be adjusted to currently prevailed frameworks, e.g., skip-gram models, CBOW, recurrent neural models [Mikolov et al., 2010]. The system can be optimized by standard strategies taken in embedding learning literature, either through standard softmax, or others like hierarchical softmax, NCE or negative sampling.

Note that the proposed algorithm ends up with distributed representations for documents, sentences and words, which could be used as input for different applications for different levels of units. But here we just keep the word embeddings for the proposed model has not been optimized regarding to upper level text units. For these large text units, it is still not clear how their meanings are made up from their containing words. Though several composition based neural network models [Tai et al., 2015, Zaremba and Sutskever, 2014, Socher et al., 2013] have been proposed and proved useful in a range of tasks, none of them manage to achieve the expected level of performance as word embedding models do.

We evaluate the learning frameworks on word analogy and word similarity tasks, two basic tasks for word embedding evaluation. Experimental results demonstrate that by harnessing the hierarchical structure of documents, we obtain better performances.

2. MODEL

Document D is comprised of a sequence of paragraphs $D = \{P_1, P_2, .., P_{N_D}\}$, paragraph P is comprised of a sequence of sentences $P = \{S_1, S_2, ...S_{N_P}\}$ and sentence S is comprised of a sequence of words $S = \{w_1, w_2, ..., w_{N_S}\}$, where N_D, N_P and N_S respectively denote the number of correspondent children in the document, paragraph and sentence. Each level text unit D, P, S, w is associated with a K dimensional embedding e_D, e_P, e_S and e_w. All text units are therefore arranged in into a tree hierarchy with $L = 4$ levels. Let η denote any node in the tree, where η could be document, paragraph, sentence or word with embedding e_η. $parent(\eta)$, $sibling(\eta)$ and $kid(\eta)$ respectively denote the parent, siblings and kids of η.

2.1 Revisit Distributed Neural Language Models

We first revisit the general neural learning framework for word embedding widely adopted in existing research.

Consider a sequence of word tokens $\{w_1, w_2, ..., w_N\}$. The conditional probability of current word is given by:

$$\begin{aligned} &p(w_n|w_{n-1}, ..., w_{n-k}, w_{n+1}, .., w_{n+k}) = \\ &f(e_n|g(e_{n-1}, ..., e_{n-k}, e_{n+1}, .., e_{n+k}), \Theta) \end{aligned} \tag{1}$$

where Θ denotes the parameter space involved in the probability function $f()$. $g()$ denotes the operation performed on neighboring vectors. Many different types of $g()$ have been explored such as averaging neighboring embeddings (CBOW) [Mikolov et al., 2013a], getting the dot product between w_n and each of its neighbors (skip-grams) [Mikolov et al., 2013a], concatenating neighboring vectors and projecting the concatenating into a low-dimensional space (e.g., [Collobert et al., 2011, Vaswani et al., 2013]), or convolving the preceding words using a recurrent network [Mikolov et al., 2011].

Commonly used forms of $f()$ include predicting current word using a softmax function or contrastive sampling. Many alternatives have been proposed for easy training, such as hierarchical softmax [Mikolov et al., 2013b] and Noise-contrastive Estimation [Vaswani et al., 2013, Mnih and Teh, 2012].

2.2 Joint Embedding Training from Hierarchical Structure

Our model take advantages of the hierarchitecture of text.

- Horizontally, we incorporate Markov property at each level of the tree structure.
- Vertically, kid embeddings are influenced by their parent nodes.

The model extends standard embedding learning framework by subsequently predicting embedding of every node η along the tree structure given its parent and siblings:

$$\begin{aligned} &p(\eta|parent(\eta), sibling(\eta)) \\ &\quad = f(e_\eta|g(e_{parent(\eta)}, \{e_{\eta', \eta' \in Sibling(\eta)}\}, \Theta) \end{aligned} \tag{2}$$

Thus, the probability of the whole document is given by:

$$\begin{aligned} &p(D|\Theta, e_D, \{e_P\}, \{e_S\}, \{e_w\}) \\ &= \prod_{\eta \in Tree} f(e_\eta|g(e_{parent(\eta)}, \{e_{\eta', \eta' \in Sibling(\eta)}\}, \Theta) \end{aligned} \tag{3}$$

As can be seen, for two words that do not reside in the same sentence, they will still distantly interact with each other as the influence is propagated up to the containing sentence embedding, paragraph embedding and document embedding, and then down to the other word. Therefore, the proposed model can to some extent capture global level statistics without losing the advantages of local neural composition.

On the other hand, based on the Markov property along each level of the trees, the meanings of adjacent text units interact with each other and preserves the integrity of meanings at each level, potentially leading to better representations at lower levels. Eventually these merits will be further propagated to word level prediction, leading to better word level embeddings.

For illustration purpose, we assume $g()$ takes the form the concatenation of sibling embeddings and parent embedding. $f(\cdot)$ takes the form of sigmoid function at sentence/paragraph level and softmax at word level. Let P denote the the paragraph that sentence S_i resides in, and S denotes the sentence that word w_i resides in, we have:

$$\begin{aligned} &p(e_{S_i}|\cdot) = \sigma(e_{S_i} \cdot g(e_P, e_{S_{i-1}}, ..., e_{S_{i-N}})] \\ &p(e_{w_i}|\cdot) = \frac{\exp(e_{w_i} \cdot g(e_S, e_{w_{i-1}}, ..., e_{w_{i-N}}))}{\sum_w \exp(e_w \cdot g(e_S, e_{w_{i-1}}, ..., e_{w_{i-N}}))} \end{aligned} \tag{4}$$

where $\sigma(\cdot)$ denotes sigmoid function.

Parameters Θ and embeddings are estimated by making MLE estimation:

$$\begin{aligned} &[\Theta, e_D, \{e_P\}, \{e_S\}, \{e_w\}] = \\ &\underset{\Theta', e'_D, e'_P, e'_S, e'_w}{\text{argmax}} \prod_D p(D|\Theta', e'_D, \{e'_P\}, \{e'_S\}, \{e'_w\}) \end{aligned} \tag{5}$$

2.3 Details of Implementation

Parameters Θ and word embeddings are to be estimated from the training corpus. Meanwhile we also estimate embeddings of document, paragraph and sentence given containing words and the correspondent embeddings. MLE estimation is implemented as is the same with previous work. A similar strategy can be found in [Le and Mikolov, 2014]. The estimated embeddings can be used as feature for downstream applications.

Model	WS-353	RG	MC	SCWS	RW
Skp-Gram	68.7	78.1	71.5	58.1	37.2
Paragraph Vector	69.2	77.8	72.9	58.0	39.6
Joint Learning	71.2	78.6	73.8	57.9	41.7
CBOW	61.7	77.8	64.5	57.2	33.8
Paragraph Vector	62.4	79.1	65.8	56.9	34.2
Joint Learning	64.2	79.2	66.4	57.2	37.1
Concatenation	70.1	76.0	72.3	54.6	35.2
Paragraph Vector	70.0	77.1	72.5	57.2	37.9
Joint Learning	71.7	77.5	74.8	57.0	39.4
GloVe	68.6	77.5	77.2	52.7	39.2

Table 1: Spearman correlation results on word similarity tasks. Dimensionality of vectors are set to 300. All reported results are based on embeddings trained from the same Wiki2014 dataset. For each subset, Paragraph Vector and Joint Learning use the same $f(\cdot)$ and $g(\cdot)$ as the model at the top.

We employ three forms of operational functions.
(1) Skip-gram model [Mikolov et al., 2013a]:

$$\begin{aligned} & f(e_\eta | g(e_{parent(\eta)}, \{e_{\eta',\eta' \in Sibling(\eta)}\}, \Theta)) \\ & = \sigma(e_\eta \cdot e_{parent(\eta)}) \prod_{\eta' \in Sibling(\eta)} \sigma(e_\eta \cdot e_{\eta'}) \end{aligned} \tag{6}$$

(2) CBOW like model [Mikolov et al., 2013a] which first averages the embeddings of parent and siblings and dot products with current node embedding:

$$\begin{aligned} & f(e_\eta | g(e_{parent(\eta)}, \{e_{\eta',\eta' \in Sibling(\eta)}\}, \Theta)) \\ & = \sigma(e_\eta \cdot g(e_{parent(\eta)}, \{e_{\eta',\eta' \in Sibling(\eta)}\})) \\ & g(e_{parent(\eta)}, \{e_{\eta',\eta' \in Sibling(\eta)}\}) = \\ & \frac{1}{1 + |Sibling(\eta)|}(e_{parent(\eta)} + \sum_{\eta' \in Sibling(\eta)} e_{\eta'}) \end{aligned} \tag{7}$$

(3) Concatenation model which can takes sequence order information by first concatenating embeddings of parent and siblings and then projects the concatenated vector sharing same dimensionality with current node embedding:

$$\begin{aligned} & f(e_\eta | g(e_{parent(\eta)}, \{e_{\eta',\eta' \in Sibling(\eta)}\}, \Theta)) \\ & = \sigma(e_\eta \cdot g(e_{parent(\eta)}, \{e_{\eta',\eta' \in Sibling(\eta)}\})) \\ & g(e_{parent(\eta)}, \{e_{\eta',\eta' \in Sibling(\eta)}\}) = \\ & \quad tanh(W \cdot [e_S, e_{w_{i-1}}, ..., e_{w_{i-N}}]) \end{aligned} \tag{8}$$

where $[\cdot]$ denotes the concatenation of its containing vectors and W denotes the $(1+N) \times K$ dimensional convolutional matrix. For concatenation approach, we use a dropout [Hinton et al., 2012, Srivastava, 2013] rate of 0.5. The initialization of embeddings for sentences, paragraphs and sentences are conducted by averaging embeddings for its containing tokens using tf-idf, similar as in [Huang et al., 2012].

3. EXPERIMENTAL RESULTS

Word Similarity Evalution.

Word embeddings are evaluated in terms of standard word similarity measures to see whether taking account text hierarchy can improve those measures. We train our models using Wikipedia2014 dataset. We adopt a hierarchical softmax function for word prediction. The window size is set to 11.

Model	Accuracy
Skp-Gram	0.691
Paragraph-Vector	0.690
Joint Learning	0.714
CBOW	0.657
Paragraph-Vector	0.662
Joint Learning	0.678
Concatenation	0.702
Paragraph-Vector	0.706
Joint Learning	0.718
GloVe	0.716

Table 2: Results on word analogy task. Models are trained on the same Wiki2014 corpus. Skip-Gram and CBOW are trained on Word2Vec.

We employ standard ontology evaluation metrics include Tofel-353 [Finkelstein et al., 2001], MC [Miller and Charles, 1991], RG [Rubenstein and Goodenough, 1965], SCWS [Huang et al., 2012], and RW [Luong et al., 2013]. Each dataset is comprised of pairs of words with gold-standard human annotations, indicating the similarity score between the pair of words. For example, "book, paper, 7.46" denotes the similarity score for word pair (book, paper) is 7.46. Standardly, we adopt cosine similarity. Spearman's rank correlation coefficient is then obtained between this score and human judgement. Baselines include Skip-Gram, CBOW, Concatenation paragraph vector [Le and Mikolov, 2014].

Word Analogy Task.

Word analogy evaluation aims at answering questions like "a is to b as c is to what". Question types include semantic ones like 'Beijing is to China like London to what" (captial) or syntactic ones like "dance to dancing as fly to what" (tense). The dataset is introduced in [Mikolov et al., 2013a] and contains 8,869 semantic questions and 10,675 syntactic questions. We follow the protocols described in [Mikolov et al., 2013c, JeffreyPennington and Manning, 2014] that to answer questions "a is to b as c is to what", we do the simple math by computing $E_b - E_a + E_c$, where E denotes the embedding for current word, and find the word d with closest representation based on cosine similarity.

Performances regarding different models are illustrated in Table 2. Similar phenomenon are observed as word similarity tasks where better performances are observed when text structure are considered. The proposed model gives better performances than previous models for word embeddings. Comparing with previous models, we consider both the local and global information.

The sole aim of this paper is to use text structure to improve word embeddings. We do generate embeddings for sentences, paragraphs and documents. But we cannot expect them to produce satisfying performances for a range of tasks without further improvements, which is our future work.

4. CONCLUSION

We present a hierarchical neural network model for word embeddings learning. Experiments verify the effectiveness of the learned word embeddings. As stated above, the learning of large text unit embeddings remains a problem. In the future we will explore new architectures to learn powerful embeddings for sentences, paragraphs and documents.

5. REFERENCES

[Bengio et al., 2006] Bengio, Y., Schwenk, H., Senécal, J.-S., Morin, F., and Gauvain, J.-L. (2006). Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer.

[Blei et al., 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

[Collobert and Weston, 2008] Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

[Collobert et al., 2011] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

[Finkelstein et al., 2001] Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. (2001). Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414. ACM.

[Hinton et al., 2012] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

[Huang et al., 2012] Huang, E. H., Socher, R., Manning, C. D., and Ng, A. Y. (2012). Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics.

[JeffreyPennington and Manning, 2014] JeffreyPennington, R. and Manning, C. (2014). Glove: Global vectors for word representation.

[Le and Mikolov, 2014] Le, Q. V. and Mikolov, T. (2014). Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*.

[Li et al., 2015a] Li, J., Jurafsky, D., and Hovy, E. (2015a). When are tree structures necessary for deep learning of representations? *arXiv preprint arXiv:1503.00185*.

[Li et al., 2014] Li, J., Li, R., and Hovy, E. (2014). Recursive deep models for discourse parsing.

[Li et al., 2015b] Li, J., Luong, M.-T., and Jurafsky, D. (2015b). A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057*.

[Luong et al., 2013] Luong, M.-T., Socher, R., and Manning, C. (2013). Better word representations with recursive neural networks for morphology. *CoNLL-2013*, 104.

[Mikolov et al., 2013a] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[Mikolov et al., 2010] Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *INTERSPEECH*, pages 1045–1048.

[Mikolov et al., 2011] Mikolov, T., Kombrink, S., Deoras, A., Burget, L., and Cernocky, J. (2011). Rnnlm-recurrent neural network language modeling toolkit. In *Proc. of the 2011 ASRU Workshop*, pages 196–201.

[Mikolov et al., 2013b] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

[Mikolov et al., 2013c] Mikolov, T., Yih, W.-t., and Zweig, G. (2013c). Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751. Citeseer.

[Miller and Charles, 1991] Miller, G. A. and Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28.

[Mnih and Teh, 2012] Mnih, A. and Teh, Y. W. (2012). A fast and simple algorithm for training neural probabilistic language models. *arXiv preprint arXiv:1206.6426*.

[Rubenstein and Goodenough, 1965] Rubenstein, H. and Goodenough, J. B. (1965). Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.

[Socher et al., 2013] Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*.

[Srivastava, 2013] Srivastava, N. (2013). *Improving neural networks with dropout*. PhD thesis, University of Toronto.

[Tai et al., 2015] Tai, K. S., Socher, R., and Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

[Vaswani et al., 2013] Vaswani, A., Zhao, Y., Fossum, V., and Chiang, D. (2013). Decoding with large-scale neural language models improves translation. In *EMNLP*, pages 1387–1392. Citeseer.

[Zaremba and Sutskever, 2014] Zaremba, W. and Sutskever, I. (2014). Learning to execute. *arXiv preprint arXiv:1410.4615*.