A Log-Linear Weighting Approach in the Word2vec Space for Spoken Language Understanding

Janod Killian[1,2]*, Mohamed Morchid*[2]*, Richard Dufour*[2]*, Georges Linarès*[2]

[1]Orkis (France)
[2]LIA - University of Avignon (France)
[1]kjanod@orkis.com, [2]firstname.surname@univ-avignon.fr

Abstract

This paper proposes an original method which integrates contextual information of words into Word2vec neural networks that learn from words and their respective context windows. However, in this word embeddings approach, context windows are represented as bag-of-words, *i.e.* every word in the context is treated equally. A log-linear weighting approach modeling the continuous context is proposed in this article to make Word2vec neural networks take into account the relative position of words in the surrounding context. Quality improvements implied by this method are shown on the the Semantic-Syntactic Word Relationship test and on a real application framework, a theme identification task of human dialogues.

Index Terms: spoken language understanding, word2vec, words embeddings, Continuous context model

1. Introduction

The selection of the best word representation becomes crucial in many Speech and Text Processing tasks. The "bag-of-words" model [1], that represents documents as a "Term Frequency-Inverse Document Frequency" (TF-IDF) [2] vector, is one of the most used representation. This representation reveals little in way of intra- and inter-document statistical structure. The limit of this type of representation is that word order in a sequence is not taken into account, *i.e.* each word being considered independently to its position in a sentence or a document.

Recently, distributed methods based on word embeddings as well as deep neural networks emerged [3]. In these approaches, all words are represented by a small dense vector corresponding to the projection of a word in a multidimensional space. Those methods were in the first place employed in Neural Language Models [4, 5], and were then used in many Natural Language Processing tasks [6, 7, 8]. Among these methods, the Word2vec compact vector [9] becomes one of the most widespread distributed word representation. Word2vec is an efficient neural network based model that captures, as a linear structure, complex semantic and syntactic relations between words from a well-structured generative model. Its effectiveness has been demonstrated in different tasks and domains [10, 11]. The Word2vec approach employs, during the training phase, the words and their relative context windows represented as a bag-of-words. Indeed, inside these context windows, each word is treated equally, its relative position being then ignored (weight of 1 if present in the context, 0 otherwise).

The paper introduces an original words weighting approach based on the Continuous Context framework [12] allowing the neural networks to take into account the position of words in the next surrounding context. A log-linear weight is then associated to each word according to its relative position in the context. The idea behinds adapting the internal weight structure of the neural network, is that the words in the close context do have a greater impact than further ones, but which should not be completely ignored. This work differs from [9] by integrating all observed variables (words) contained in a window with respect to internal words distribution (position of the word) alongside with a well-adapted log-linear model.

The proposed weighting method is both evaluated qualitatively and quantitatively with the Semantic-Syntactic Word Relationship test [9] and a theme identification task of spoken dialogues between an agent and a customer from the French Paris transportation call center.

The paper is organized as follows: Section 2 details the two Word2vec neural network architectures while the proposed weighting function integrated into these architectures is presented in Section 3. Experiments and results are detailed in Section 4 before concluding in Section 5.

2. Word2vec Architectures

Word2vec is a method based on artificial neural networks defined in [9]. The aim of this method is to build word embeddings by maximizing the likelihood L that words are predicted from their context. This method proposes two shallow artificial neural network architectures: the Continuous Bag-Of-Words (CBOW) and the Skip-

gram (SG) shown in Figure 1. Both of these models take as input a binary vector features (1 if the word appears, 0 otherwise).

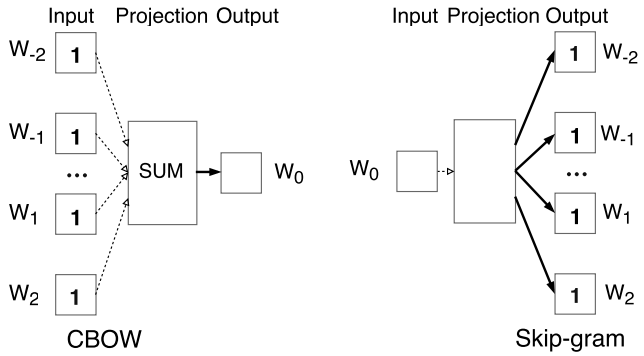

Figure 1: `Word2vec` neural network architectures.

The Skip-gram (SG) model predicts a context for a given word in a vocabulary V. The input layer of the Skip-gram algorithm only contains the central word and projects it in the output layer, through the hidden layer. The prediction is corrected with each word within the context window. The Skip-gram model maximizes the likelihood:

$$L = \frac{1}{T}\sum_{t=1}^{T}\sum_{j=t-c, j\neq t}^{t+c} \log p(w_j|w_t) \tag{1}$$

where c is a hyper-parameter defining the window of the context words; T is the size of the training data, and c is the size of the context for each word. The model estimates a global matrix M of dimension $|V| \times n$, where n is the embedding dimension. Then, each embedding representation of a word w_i is mapped in a $|V|$-dimensional vector v_{w_i} to obtain the output probability $p(w_0, w_i)$ for a given word w_0 is given by the softmax function:

$$p(w_0, w_i) = \frac{e^{v_{w_0}}}{\sum_{w\in V} e^{v_{w_i}}} \tag{2}$$

The Continuous Bag-of-Words (CBOW) model attempts to find the center word w_0 for a given set of surrounding words $\{w_{-c}, \ldots, w_{-1}, w_1, \ldots, w_c\}$. Each word in the context is projected in the global matrix M. The central word of the context window is used to correct the network prediction using back-propagation algorithm. This model seeks to maximize the likelihood L defined thereafter:

$$L = \frac{1}{T}\sum_{t=1}^{T} \log p(w_t|w_{t-c}...w_{t+c}) \tag{3}$$

Moreover with the Skip-gram network, a skip mechanism is introduce . This process reduce the size of the context c by a random number every learning step. On one hand this mechanism make the learning faster, but on the other hand it will skip some rare relations.

Skip-gram (SG) as well as CBOW models do not take into account the word order to predict the output. Therefore, these methods allow us to find out close vectorial representation of a given word or context, but are not optimal to predict word sequence based on grammatical and syntactic properties of words.

3. Log-linear `Word2cec` Models

The proposed method takes advantage of words position to improve the word embedding representation with a log-linear context weighting function [12] δ to replace binary features of input (CBOW) or output (Skip-gram) vectors. The context is weighted with $\delta(w)$ for each word w:

$$\delta(w) = \frac{\alpha}{\gamma + \beta log(d(w))} \tag{4}$$

where $d(w)$ is the distance between the word in the center of the context c and the word w to weight; α, γ and β are parameters of the distance function. The log-linear function is well adapted to words weighting because this function will give high weights to words close to the center and lower weights to further words in the context. In other words, it considers that further words in the context have a lower impact than closer ones, but however should not be ignored. Figure 2 presents these models including the proposed context weighting approach.

3.1. Log-linear CBOW (LL-CBOW)

The proposed approach differs from the classical CBOW, by replacing the binary input features (1 if the word $w_i \in c$, 0 otherwise) with the value of $\delta(w)$. Thus, this model, denoted LL-CBOW, gives a different weight to the word w_i depending on its position in the context window of words c.

3.2. Log-linear Skip-gram (LL-SG)

The SG final layer uses a softmax activation function which can hardly predict more than one word at a time. To correct the network with a context window, one have to calculate the error between the prediction and each word in the context to learn one by one. This process prevents the use of the weighted sum to mirror the LL-CBOW. Instead, weights are applied on the errors generated based on words distance.

4. Experiments and Results

In this section, we propose to evaluate the effectiveness of the proposed weighting window by comparing this original approach with the initial one on the Semantic-Syntactic Word Relationship test and on a theme identification task of spoken dialogues.

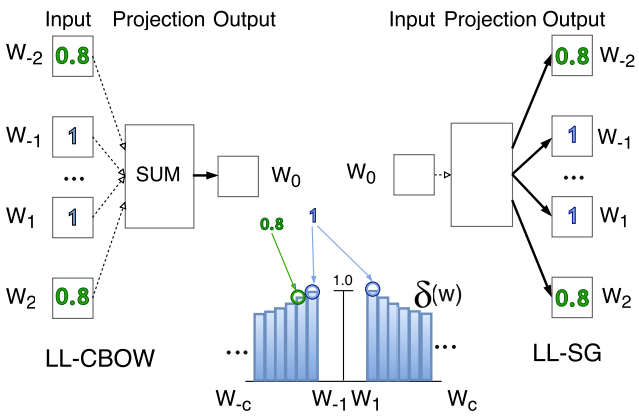

Figure 2: `Word2vec` method using the proposed log-linear context weighting approach.

4.1. Semantic-Syntactic Word Relationship test

The Semantic-syntactic Word relationship test [9] is made of 19,000 questions. Its main objective is to verify if a distributed representation of words captures complex syntactic and semantic relations between words. A question is made of two pairs of words sharing the same relation:

$$\text{Paris} - \text{France} = \text{London} - \text{England} \quad (5)$$

Performance is measured in terms of percentage of correctly retrieved relations. For this experiment, different `Word2vec` Skip-gram and CBOW configurations with various context and hidden layer sizes are evaluated as shown in Tables 2 and 3. The weighting function is defined for these experiments as follows:

$$\delta(w) = \frac{1 + log(2)}{1 + log(d(w))} \quad (6)$$

The English corpus used for training the different word embeddings is composed of:

Word Language Modeling Benchmark : a corpus made for language modeling containing 31 million documents (700 million words).

Wikipedia : an English dump from Wikipedia containing 124 303 documents (124 million words).

Gigaword : the English Gigaword from 1994 to 2011 containing 190 million documents (3,771 million words).

the Brown corpus : general text corpus in the field of corpus linguistics containing 57,341 documents (1 million words).

The final training corpus contains around 4 billion words for a vocabulary size of 1 million words. Table 1 presents neighborhood words extracted with CBOW and LL-CBOW trained on this corpus, one could notice that models integrating log-linear weighting tend to thematically gather related words.

Table 1: Examples of neighborhood words extracted from models trained without (CBOW baseline) and with contextual information (LL-CBOW proposed approach).

Holidays		Meat	
LL-CBOW	CBOW	LL-CBOW	CBOW
holiday	vacations	chicken	pork
vacation	**thanksgiving**	beef	**not-pasteurized**
festivities	vacation	pork	mutton
thanksgiving	**christmas**	**milk**	**eggs**
easter	celebration	**eggs**	cattle
christmas	**easter**	**seafood**	chicken

Tables 2 and 3 show that models trained using the proposed continuous context weighting approach globally achieve better results. Best improvement is obtained using the whole document as the context (100 words), with a gain of 7% for the LL-CBOW and of 7.7% for the LL-SG. The negative impact of the context window size is reduced and almost neglected using LL-SG, conversely to both Skip-gram and CBOW where the performance falls when augmenting the context size. Moreover the LL-CBOW is improved by the using bigger weighted context. Furthermore, Table 3 points out that a smaller hidden layer tends to have a smaller gain by using the log-linear weighting method. This could be explained by the fact that having a smaller hidden layer makes the network capable of memorizing less information, making it more difficult for it to capture the contextual information.

Table 2: Accuracies (%) on the Semantic-Syntactic Word Relationship test depending on the context size (c) with a hidden layer of size 300.

	Skip-gram			CBOW		
context	10	15	100	10	15	100
standard	50.0	50.9	43.7	39	38.9	36.9
log-linear	**55.0**	53.7	51.4	39.9	39.6	**43.9**

Table 3: Accuracies (%) on the Semantic-Syntactic Word Relationship test without and with weighting distance using different hidden layer sizes and a context window of size 10.

	Skip-gram		CBOW	
hidden layer's size	120	300	120	300
standard	43.9	50.0	29.0	39.0
log-linear	45.1	**55.0**	30.3	**39.9**

Table 4: Description of the DECODA dataset.

Class label	*Number of samples*		
	training	**dev.**	**text**
problems of itinerary	145	44	67
lost and found	143	33	63
time schedules	47	7	18
transportation cards	106	24	47
state of the traffic	202	45	90
fares	19	9	11
infractions	47	4	18
special offers	31	9	13
Total	**740**	**175**	**327**

4.2. Classification of Telephone Conversations

The second experiment evaluates the proposed weighting approach in a classification task using the DECODA project corpus [13] which aims at identifying conversation themes. This corpus is composed of 1,242 telephone conversations split in training, dev. and test validation sets, each conversation being manually annotated with one of the 8 themes as shown in Table 4.

The LIA-Speeral automatic speech recognition (ASR) system [14] with 230,000 Gaussians in the triphone acoustic models has been used for the experiments. Model parameters were estimated with maximum *a posteriori* probability (MAP) adaptation from 150 hours of speech in telephone condition. The vocabulary contains 5,782 words, a 3-gram language model (LM) was obtained by adapting a basic LM with the transcriptions of the DECODA training set. The ASR system obtains Word Error Rates (WER) of 33.8% on the training, 45.2% on the development, and 49.5% on the test set. These high WERs are mainly due to speech disfluencies and to adverse acoustic environments for some dialogues. The word embedding models are trained on a French corpus composed of:

GigaWord : The French version containing 17 million documents (500 million words).

Wikipedia : A dump of the French Wikipedia composed of 16 million documents (400 million words).

Newspapers : Various French newspapers such as the *AFP*, *Le Monde* and *Le Soir* containing 56 million documents (737 million words).

Documents crawled : Documents crawled from the Internet representing 4 million documents (108 million words).

Manual transcriptions : Various manual transcription from recent French evaluation campaigns such as ESTER, EPAC, ETAPE and REPERE, containing 411,000 documents (379 million words).

This corpus contains 2 billion words for a vocabulary of 3 million words. The train set is used to compose a subset of discriminative words selected with the TF-IDF-Gini method [15]. For each theme, a set of 100 specific words is identified to form a vocabulary of 707 words. All the selected words are then merged without repetition (note that a same word may appear in the vocabulary of more than one theme). Each discriminant word is used as a landmark in the multidimensional space. Then each dialogue is assigned with a vector of scores representing the distance between the sum of words in the dialogue and each landmark. Finally, those features are given to a classifier. To evaluate the impact of the continuous context weighting function, 4 models were trained: a CBOW, a LL-CBOW, a Skip-gram and a LL-SG. Both LL-CBOW and LL-SG methods use the proposed weighting function defined in Section 4.1, while baseline approaches (CBOW and Skip-gram) do not integrate any weighting function.

For this task, two different types of classifiers are used: a Gradient Tree Boosting (GTB) [16, 17] and a Multilayer Perceptron (MLP) neural network [18, 19]. GTB algorithm is a generalization of the boosting algorithm using a loss function. This classifier is used as the baseline classifier for its off-the-shelf performance. The Multilayer Perceptron (MLP) neural network [18, 19] is made of 3 layers: 707, 32 and 8 neurons for input, hidden and output respectively with "sigmoid" then "softmax" activation layers as well as dropout regularization. Each classifier is trained with the features made with the 4 models.

The accuracies observed on both development and test sets using Skip-gram/LL-SG and CBOW/LL-CBOW architectures, are presented in Table 5. This table points out that all the tested classification approaches improve their performance by using the proposed contextual information. Moreover, the weighting function allows the Skip-gram based models (LL-SG) to better identify the theme in the dialogues with a gain of 10 and 20 points with the GTB and MLP classification methods respectively. This phenomenon is also observed for the CBOW-based models (LL-CBOW) with a gain of 33 and 34 points for the GTB and MLP classification methods respectively. For the MLP, results are measured every 10 epochs on the dev. and reported in Figure 3. Models trained with the log-linear weighting approach are more accurate and converge faster than their counterpart.

These experiments show that inserting a log-linear weight in a `Word2vec` model allows us to capture more information: the bigger the context is, the more important information the continuous context contains. In these experiments, neural networks need a large enough hidden layer to capture the additional information. This difference leads to slightly better models, as shown by the gain on the Semantic-Syntactic Word Relationship test (see Section 4.1). Results on the DECODA task show that

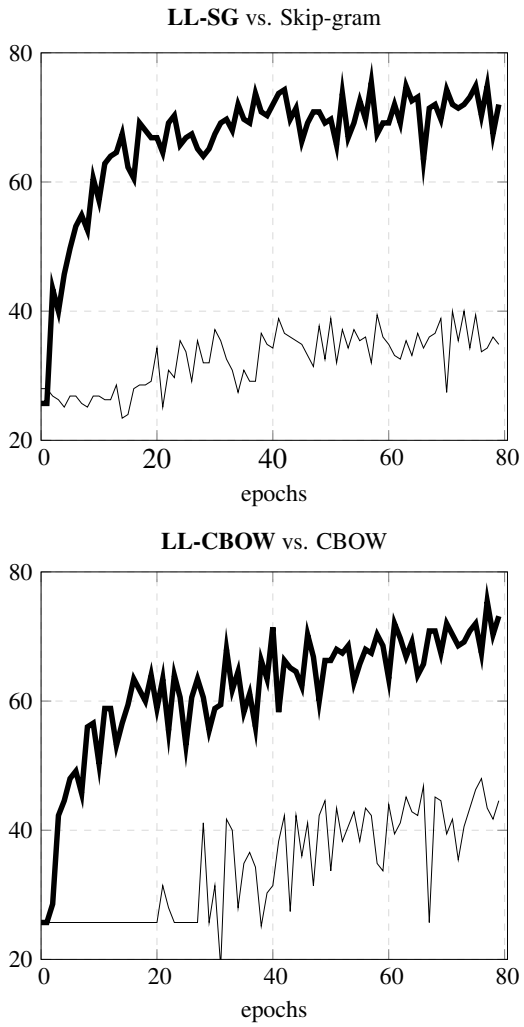

Figure 3: DECODA theme identification accuracies (y axis in %) obtained with the Skip-gram and CBOW models as well as the proposed **LL-SG** and **LL-CBOW** models in **bold** for different epochs.

Table 5: Classification accuracies (%) on the DECODA task using two classification approaches and features from word embeddings models without (standard) and with log-linear weighting.

	Skip-gram		CBOW	
	Dev.	Test	Dev.	Test
GTB(standard)	39	42	28	27
GTB(log-linear)	56	52	66	60
MLP(standard)	50	50	41	37
MLP(log-linear)	75	**70**	74	**71**

`Word2vec` embeddings learnt with the contextual information project words in a space where a thematic classification on textual data coming from spoken dialogues is made easier.

5. Conclusion

The `Word2vec` context window uses words as a bag-of-words representation and randomly ignores distant words and thus. A bag-of-words representation treats each word equally in a given context. Both qualitative and quantitative experiments show that the *Word2vec* compact representation without an adapted weighting strategy obtains lower results compared to the performance obtained with the proposed log-linear weighting approach. This paper proposes an alternative word weighting method that reinforces the contextual information and preserves distant relationship in distributed representations of words. Our experiments made on a word similarity test and a classification task of noisy spoken dialogues, show that accuracies were improved to 7% on the similarity task, and more than 20% on the classification task. These experiments also demonstrate that the use of our method is relevant to a thematic classification task based on word embedding features. We plan to extend this work by evaluating the impact of different types of weighting functions and of the same information on different distributed word representations.

6. References

[1] G. Salton, "Automatic text processing: the transformation," *Analysis and Retrieval of Information by Computer*, 1989.

[2] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[3] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[4] S. Bengio and G. Heigold, "Word embeddings for speech recognition," in *Proceedings of the 15th Conference of the International Speech Communication Association, Interspeech*, 2014.

[5] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural Language Processing (almost) from Scratch," *The Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[6] Q.-K. Do, A. Allauzen, and F. Yvon, "Modèles de langue neuronaux: une comparaison de plusieurs stratégies dapprentissage," in *TALN 2014*, 2014.

[7] A. Vaswani, Y. Zhao, V. Fossum, and D. Chiang, "Decoding with large-scale neural language models improves translation." in *EMNLP*. Citeseer, 2013, pp. 1387–1392.

[8] G. Mesnil, T. Mikolov, M. Ranzato, and Y. Bengio, "Ensemble of Generative and Discriminative Techniques for Sentiment Analysis of Movie Reviews," 2015.

[9] T. Mikolov, G. Corrado, K. Chen, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *Proceedings of the International Conference on Learning Representations (ICLR 2013)*, pp. 1–12, 2013.

[10] M. Iyyer, P. Enns, J. Boyd-Graber, and P. Resnik, "Political ideology detection using recursive neural networks," in *Association for Computational Linguistics*, 2014.

[11] Y. Kim, "Convolutional neural networks for sentence classification," 2014.

[12] B. Bigot, G. Senay, G. Linares, C. Fredouille, and R. Dufour, "Combining Acoustic Name Spotting and Continuous Context Models to improve Spoken Person Name Recognition in Speech," *Interspeech*, pp. 2539–2543, 2013.

[13] F. Bechet, B. Maza, N. Bigouroux, T. Bazillon, M. El-Beze, R. De Mori, and E. Arbillot, "Decoda: a call-centre human-human spoken conversation corpus." in *LREC*, 2012, pp. 1343–1347.

[14] G. Linares, P. Nocéra, D. Massonie, and D. Matrouf, "The lia speech recognition system: from 10xrt to 1xrt," in *Text, Speech and Dialogue*. Springer, 2007, pp. 302–308.

[15] S. R. Singh, H. A. Murthy, and T. A. Gonsalves, "Feature selection for text classification based on gini coefficient of inequality." *FSDM*, vol. 10, pp. 76–85, 2010.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[17] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics Springer, Berlin, 2001, vol. 1.

[18] D. W. Ruck, S. K. Rogers, M. Kabrisky, M. E. Oxley, and B. W. Suter, "The multilayer perceptron as an approximation to a bayes optimal discriminant function," *Neural Networks, IEEE Transactions on*, vol. 1, no. 4, pp. 296–298, 1990.

[19] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio, "Theano: new features and speed improvements," Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.