IDIAP RESEARCH REPORT

WORD EMBEDDINGS THROUGH HELLINGER PCA

Rémi Lebret [a] Ronan Collobert

Idiap-RR-29-2013

AUGUST 2013

[a] Idiap

Centre du Parc, Rue Marconi 19, P.O. Box 592, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

Word Embeddings through Hellinger PCA

Rémi Lebret
Idiap Research Institute
Rue Marconi 19, CP 592
1920 Martigny, Switzerland
rlebret@idiap.ch

Ronan Collobert
Idiap Research Institute
Rue Marconi 19, CP 592
1920 Martigny, Switzerland
ronan@collobert.com

Abstract

Word embeddings resulting from neural language models have been shown to be successful for a large variety of NLP tasks. However, such architecture might be difficult to train and time-consuming. Instead, we propose to drastically simplify the word embeddings computation through a Hellinger PCA of the word co-occurence matrix. We compare those new word embeddings with the Collobert and Weston (2008) embeddings on several NLP tasks and show that we can reach similar or even better performance.

1 Introduction

Building word embeddings has always generated much interest for linguists. Popular approaches such as Brown clustering algorithm (Brown et al., 1992) has been used with success in a wide variety of NLP tasks (Schütze, 1995; Koo et al., 2008; Ratinov and Roth, 2009). Those word embeddings are often associated with a low dimensional-vector space where the dimensions can be seen as features potentially describing syntactic or semantic properties. Recently, distributed approaches based on neural network language models (NNLM) have revived the field of learning word embeddings (Collobert and Weston, 2008; Huang and Yates, 2009; Turian et al., 2010; Collobert et al., 2011; Chen et al., 2013). However, a neural network architecture can be hard to train. Finding the right parameters to tune the model is often a challenging task and it takes time to train the whole thing.

This paper aims to show that such good word embeddings can be obtained using simple linear operations. We show that similar word embeddings can be computed using the word co-occurrence statistics and a well-known dimensionality reduction operation such as Principal Component Analysis (PCA). We then compare our embeddings with the (Collobert and Weston, 2008) embeddings on several NLP tasks.

2 Related Work

As 80% of the meaning of English text comes from word choice and the remaining 20% comes from word order (Landauer, 2002), it seems quite important to preserve word order. Connectionist approaches have therefore been proposed to develop distributed representations which encode the structural relationships between words (Hinton, 1986; Pollack, 1990; Elman, 1991). Most recently, a neural network language model was proposed in Bengio et al. (2003) where word vector representations are simultaneously learned along with a statistical language model. This architecture inspired other authors: Collobert and Weston (2008) designed a neural language model which eliminates the linear dependency on vocabulary size, Mnih and Hinton (2008) proposed a hierarchical linear neural model, Mikolov et al. (2010) investigated a recurrent neural network architecture for language modeling. Such architectures being trained over large corpora of unlabeled text with the aim to predict correct scores end up learning the co-occurence statistics.

Linguists assumed long ago that words occurring in similar contexts tend to have similar meanings

(Wittgenstein, 1953). Using the word co-occurrence statistics is thus a natural choice to embed similar words into a common vector space (Turney and Pantel, 2010). Common approaches calculate the frequencies, apply some transformations (tf-idf, PPMI), reduce the dimensionality and calculate the similarities (Lowe, 2001). Considering a fixed-sized word vocabulary $\mathcal{D}$, the co-occurence matrix is then vocabulary size dependent. To reduce the dimensionality of the co-occurence matrix F by mapping F into a matrix f of size $W \times d$, where $d \ll |\mathcal{D}|$, techniques such as Singular Valued Decomposition (SVD) is widely used (e.g. LSA (Landauer and Dumais, 1997), ICA (Väyrynen and Honkela, 2004)). However, word co-occurence statistics are discrete distributions. We thus believe that information theory distance measure such as Hellinger distance should be more efficient than Euclidean distance to smooth the matrix F.

3 Word Representations Model

A NNLM learns which words among the vocabulary appears more likely after a given context sequence of words. More formally, it learns the next word probability distribution. Instead, simply counting words on a large corpus of unlabeled text can be perform to retrieve those word distributions and to represent words (Turney and Pantel, 2010).

3.1 Word co-occurence statistics

"You shall know a word by the company it keeps" (Firth, 1957). It is a natural choice to use the word co-occurence statistics to acquire representations of word meanings. Raw word co-occurence frequencies are computed by counting the number of times each word $w \in \mathcal{D}$ occurs after a context sequence of words T:

$$p(w|T) = \frac{p(w,T)}{p(T)} = \frac{n(w,T)}{\sum_c n(w,T)} \qquad (1)$$

where $n(w,T)$ is the number of times each context word w occurs after the context T. The next word probability distribution p for each word or sequence of words is thus obtained. It is a multinomial distribution of $|\mathcal{D}|$ classes (words). A co-occurence matrix of size $N \times |\mathcal{D}|$ is thus obtained by computing those frequencies over all the N possible sequences of words.

3.2 Hellinger distance

Similarities between words can be derived by computing a distance between their corresponding word distributions. Several distances (or metric) over discrete distributions exist, such as the Bhattacharyya distance, the Hellinger distance or Kullback-Leibler divergence. We chose here the Hellinger distance for its simplicity and symmetry property (as it is a true distance). Considering two discrete probability distributions $P = (p_1, \ldots, p_k)$ and $Q = (q_1, \ldots, q_k)$, the Hellinger distance is formally defined as:

$$H(P,Q) = -\frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{k} (\sqrt{p_i} - \sqrt{q_i})^2} \qquad (2)$$

which is directly related to the Euclidean norm of the difference of the square root vectors:

$$H(P,Q) = \frac{1}{\sqrt{2}} \|\sqrt{P} - \sqrt{Q}\|_2 \qquad (3)$$

Note that it makes more sense to take the Hellinger distance rather than the Euclidean distance for comparing discrete distributions, as P and Q are unit vectors according to the Hellinger distance ($\sqrt{P}$ and $\sqrt{Q}$ are units vector according to the ℓ_2 norm).

3.3 Dimensionality Reduction

As discrete distributions are vocabulary size dependent, using directly the distribution as a word embedding is not really tractable for large vocabulary. We propose to perform a principal component analysis (PCA) of the word co-occurence probabilities square root matrix to represent words in a lower dimensional space while minimizing the reconstruction error according to the Hellinger distance.

4 Experimental Setup

We evaluate the quality of our embeddings obtained on a large corpora of unlabeled text by comparing their performance of several NLP tasks against performance obtained on these tasks with the Collobert and Weston (2008) embeddings.

4.1 Building Word Representation over Large Corpora

Our English corpus is composed of the entire English Wikipedia[1] (where all MediaWiki markups have been removed), the Reuters corpus and the Wall Street Journal (WSJ) corpus. We chose to consider lower case words to limit the number of words in the vocabulary. Additionally, all occurrences of sequences of numbers within a word are replaced with the string "NUMBER". The resulting text was tokenized using the Stanford tokenizer[2]. The data set contains about 1,652 million words. As vocabulary we considered all the words within the supervised evaluation tasks data sets described in section 4.2. This results in a 48,004 words vocabulary. To build the co-occurence matrix we use two different sets of context words:

- the most common words in our corpus split into two sets: the 10,000 and the 50,000 most common. The resulting embeddings after PCA are respectively called $H_1^{(10000)}$ and $H_1^{(50000)}$ in section 4.3. We also built the co-occurence probabilities matrix without the square root to highlight the importance of this aspect with $N = 10000$. The resulting embeddings are called $E_1^{(10000)}$.
- the vocabulary itself and a subset version by keeping the 10,000 most frequent words. The resulting embeddings are respectively called $H_2^{(10000)}$ and $H_2^{(48004)}$ in section 4.3.

Each word is represented in a 50-dimensional vector as the Collobert and Weston (2008) (CW) embeddings.

4.2 Supervised Evaluation Tasks

CW embeddings[3] proved that they can improve the generalization performance on several NLP tasks (Turian et al., 2010; Collobert et al., 2011; Chen et al., 2013). Using our word embeddings, we thus trained the sentence-level log-likelihood architecture described in Collobert et al. (2011) with the exact same parameters on three NLP tasks.

Part-Of-Speech (POS) it aims at labeling each word with a unique tag that indicate its syntactic role. Sections 0-18 of Wall Street Journal (WSJ) data are used for training, while sections 19-21 are for validation and sections 22-24 for testing.

Chunking (CHUNK) it aims at labeling segments of a sentence with syntactic constituents. Chunking is often evaluated using the CoNLL 2000 shared task[4]. Sections 15-18 of WSJ data are used for training and section 20 for testing. Validation is achieved by splitting the training set.

Named Entity Recognition (NER) it labels atomic elements in the sentence into categories such as "PERSON" or "LOCATION". The CoNLL 2003 setup[5] is a NER benchmark data set based on Reuters data. The contest provides training, validation and testing sets.

These evaluation tasks are mainly syntactic. We wish to evaluate whether our word embeddings can also capture semantic.

Semantically Related Words Word embeddings have been recently benchmarked in Chen et al. (2013). They proposed several evaluation tasks to test the syntactic and semantic properties of different embeddings. To evaluate the semantic characteristics we selected the *Synonyms and Antonyms* task which attempt to classify synomyns word pairs against antonyms word pairs extracted from Wordnet. We followed the exact setup described in the paper. We ended up with 3,900 different word pairs equally distributed among synonyms and antonyms.

4.3 Results

For the three NLP tasks, networks are fed with two raw features: lower case words, and a capital letter feature. The "caps" feature tells if each word was in lowercase, was all uppercase, had first letter capital, or had at least one non-initial capital letter. No other feature has been used to tune the models. Results summarized in Table 1 reveal that performance on NLP tasks can be as good with word embeddings from a word co-occurence matrix decomposition as with a neural network language model trained for weeks. Using a 10,000 words vocabulary built over

[1] Available at http://download.wikimedia.org. We took the May 2012 version.

[2] Available at http://nlp.stanford.edu/software/tokenizer.shtml

[3] Available in SENNA at http://ml.nec-labs.com/senna/.

[4] See http://www.cnts.ua.ac.be/conll2000/chunking

[5] http://www.cnts.ua.ac.be/conll2003/ner/

Approach	POS (PWA)	CHUNK (F1)	NER (F1)
CW (LM1)	97.10	93.65	87.58
CW (LM2)	97.20	93.63	88.67
$H_1^{(10000)}$	97.11	93.38	88.29
$E_1^{(10000)}$	97.02	92.46	86.90
$H_1^{(50000)}$	97.02	93.05	87.79
$H_2^{(10000)}$	97.05	93.02	87.19
$H_2^{(48004)}$	97.20	93.75	88.15

Table 1: Comparison in generalization performance on POS, chunking and NER tasks. We report results from (Collobert et al., 2011) with neural networks trained with CW embeddings (LM1: from a 100,000 words vocabulary; LM2: from a 130,000 words vocabulary) and with our own word embeddings (H_1, E_1 and H_2). Generalization performance is reported in per-word-accuracy (PWA) for POS and F1 score for other tasks.

a large corpus yields similar performance than a neural language model with a 130,000 words vocabulary. The Hellinger distance demonstrates its value compared to the Euclidean distance. We can also remark that enlarging the vocabulary size does not improve the results when the words come from large corpora. However, performance increases when the co-occurence distributions are computed with words from the evaluation data sets.

For the synonyms and antonyms classification task, we used a SVM with RBF-kernel as linear classifier as in (Chen et al., 2013). We obtained a similar accuracy between CW embeddings and the $H_2^{(48004)}$ embeddings. We again notice the Hellinger distance outperform the Euclidean one. As for the evaluation on POS, chunking and NER, $H_1^{(10000)}$ embeddings outperform $H_1^{(50000)}$ in term of accuracy. Results on this task are summarized in table 2.

Embeddings	Accuracy (%)
CW (LM2)	76.95
$H_1^{(10000)}$	71.51
$E_1^{(10000)}$	68.05
$H_1^{(50000)}$	70.26
$H_2^{(10000)}$	70.10
$H_2^{(48004)}$	76.00

Table 2: Accuracy on Synonyms and Antonyms tasks with different word embeddings.

5 Conclusion

We have demonstrated that appealing word embeddings can be obtained by computing a *Hellinger PCA* of the word co-occurence matrix. While a neural network language model can be painful and long to train, we can get a word co-occurence matrix by simply counting words over a large corpus. The resulting embeddings give similar results on NLP tasks even from a $N \times 10,000$ word co-occurence matrix. We have shown that adding more common words to the word co-occurence matrix does not help to increase performance. However by adding more specific words (like for $H_2^{(48004)}$ which contains all words within the evaluation data sets even the rare ones) performance can be improved. Having a significant but not too large set of common words seems sufficient for capturing most of the syntactic and semantic characteristics of words. To refine the embeddings, it is crucial to also add the most discriminative words. Future work should explore more context words to build the co-occurence matrix. Bullinaria and Levy (2007) demonstrated that using the left word context can improve general performance. We could also apply this method to embed sequences of words in a common space.

Acknowledgments

We gratefully acknowledge the financial support of the Hasler Foundation.

References

Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March.

P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

J. A. Bullinaria and J. P. Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, pages 510–526.

Y. Chen, B. Perozzi, R. Al-Rfou', and S. Skiena. 2013. The expressive power of word embeddings. *CoRR*, abs/1301.3226.

R. Collobert and J. Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *International Conference on Machine Learning, ICML*.

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

J. L. Elman. 1991. Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, 7:195–225.

J. R. Firth. 1957. A synopsis of linguistic theory 1930-55. 1952-59:1–32.

G. E. Hinton. 1986. Learning distributed representations of concepts. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, pages 1–12. Hillsdale, NJ: Erlbaum.

F. Huang and A. Yates. 2009. Distributional representations for handling sparsity in supervised sequence-labeling. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 495–503. Association for Computational Linguistics.

T. Koo, X. Carreras, and M. Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 595–603.

T. K. Landauer and S. T. Dumais. 1997. A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104:211–240.

T. K. Landauer. 2002. On the computational basis of learning and cognition: Arguments from lsa. In N. Ross, editor, *The psychology of learning and motivation*, volume 41, pages 43–84. Academic Press, San Francisco, CA.

W. Lowe, 2001. *Towards a theory of semantic space*, pages 576–581.

T. Mikolov, M. Karafit, L. Burget, J. ernock, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010)*, volume 2010, pages 1045–1048. International Speech Communication Association.

A. Mnih and G. Hinton. 2008. A Scalable Hierarchical Distributed Language Model. In *Advances in Neural Information Processing Systems*, volume 21.

J. B. Pollack. 1990. Recursive distributed representations. *Artificial Intelligence*, 46:77–105.

L. Ratinov and D. Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL)*, pages 147–155. Association for Computational Linguistics.

H. Schütze. 1995. Distributional part-of-speech tagging. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 141–148. Morgan Kaufmann Publishers Inc.

J. Turian, L. Ratinov, and Y. Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *ACL*.

P. D. Turney and P. Pantel. 2010. From frequency to meaning: Vector space models of semantics. *CoRR*, abs/1003.1141.

J. J. Väyrynen and T. Honkela. 2004. Word category maps based on emergent features created by ICA. In Heikki Hyötyniemi, Pekka Ala-Siuru, and Jouko Seppänen, editors, *Proceedings of the STeP'2004 Cognition + Cybernetics Symposium*, number 19 in Publications of the Finnish Artificial Intelligence Society, pages 173–185. Finnish Artificial Intelligence Society.

L. Wittgenstein. 1953. *Philosophical Investigations*. Blackwell.