Linguistic Regularities in Continuous Space Word Representations

Tomas Mikolov*, **Wen-tau Yih, Geoffrey Zweig**
Microsoft Research
Redmond, WA 98052

Abstract

Continuous space language models have recently demonstrated outstanding results across a variety of tasks. In this paper, we examine the vector-space word representations that are implicitly learned by the input-layer weights. We find that these representations are surprisingly good at capturing syntactic and semantic regularities in language, and that each relationship is characterized by a relation-specific vector offset. This allows vector-oriented reasoning based on the offsets between words. For example, the male/female relationship is automatically learned, and with the induced vector representations, "King - Man + Woman" results in a vector very close to "Queen." We demonstrate that the word vectors capture syntactic regularities by means of syntactic analogy questions (provided with this paper), and are able to correctly answer almost 40% of the questions. We demonstrate that the word vectors capture semantic regularities by using the vector offset method to answer SemEval-2012 Task 2 questions. Remarkably, this method outperforms the best previous systems.

1 Introduction

A defining feature of neural network language models is their representation of words as high dimensional real valued vectors. In these models (Bengio et al., 2003; Schwenk, 2007; Mikolov et al., 2010), words are converted via a learned lookup-table into real valued vectors which are used as the inputs to a neural network. As pointed out by the original proposers, one of the main advantages of these models is that the distributed representation achieves a level of generalization that is not possible with classical n-gram language models; whereas a n-gram model works in terms of discrete units that have no inherent relationship to one another, a continuous space model works in terms of word vectors where similar words are likely to have similar vectors. Thus, when the model parameters are adjusted in response to a particular word or word-sequence, the improvements will carry over to occurrences of similar words and sequences.

By training a neural network language model, one obtains not just the model itself, but also the learned word representations, which may be used for other, potentially unrelated, tasks. This has been used to good effect, for example in (Collobert and Weston, 2008; Turian et al., 2010) where induced word representations are used with sophisticated classifiers to improve performance in many NLP tasks.

In this work, we find that the learned word representations in fact capture meaningful syntactic and semantic regularities in a very simple way. Specifically, the regularities are observed as constant vector offsets between pairs of words sharing a particular relationship. For example, if we denote the vector for word i as x_i, and focus on the singular/plural relation, we observe that $x_{apple} - x_{apples} \approx x_{car} - x_{cars}$, $x_{family} - x_{families} \approx x_{car} - x_{cars}$, and so on. Perhaps more surprisingly, we find that this is also the case for a variety of semantic relations, as measured by the SemEval 2012 task of measuring relation similarity.

*Currently at Google, Inc.

Proceedings of NAACL-HLT 2013, pages 746–751,
Atlanta, Georgia, 9–14 June 2013.

The remainder of this paper is organized as follows. In Section 2, we discuss related work; Section 3 describes the recurrent neural network language model we used to obtain word vectors; Section 4 discusses the test sets; Section 5 describes our proposed vector offset method; Section 6 summarizes our experiments, and we conclude in Section 7.

2 Related Work

Distributed word representations have a long history, with early proposals including (Hinton, 1986; Pollack, 1990; Elman, 1991; Deerwester et al., 1990). More recently, neural network language models have been proposed for the classical language modeling task of predicting a probability distribution over the "next" word, given some preceding words. These models were first studied in the context of feed-forward networks (Bengio et al., 2003; Bengio et al., 2006), and later in the context of recurrent neural network models (Mikolov et al., 2010; Mikolov et al., 2011b). This early work demonstrated outstanding performance in terms of word-prediction, but also the need for more computationally efficient models. This has been addressed by subsequent work using hierarchical prediction (Morin and Bengio, 2005; Mnih and Hinton, 2009; Le et al., 2011; Mikolov et al., 2011b; Mikolov et al., 2011a). Also of note, the use of distributed topic representations has been studied in (Hinton and Salakhutdinov, 2006; Hinton and Salakhutdinov, 2010), and (Bordes et al., 2012) presents a semantically driven method for obtaining word representations.

3 Recurrent Neural Network Model

The word representations we study are learned by a recurrent neural network language model (Mikolov et al., 2010), as illustrated in Figure 1. This architecture consists of an input layer, a hidden layer with recurrent connections, plus the corresponding weight matrices. The input vector $\mathbf{w}(t)$ represents input word at time t encoded using 1-of-N coding, and the output layer $\mathbf{y}(t)$ produces a probability distribution over words. The hidden layer $\mathbf{s}(t)$ maintains a representation of the sentence history. The input vector $\mathbf{w}(t)$ and the output vector $\mathbf{y}(t)$ have dimensionality of the vocabulary. The values in the hidden and output layers are computed as follows:

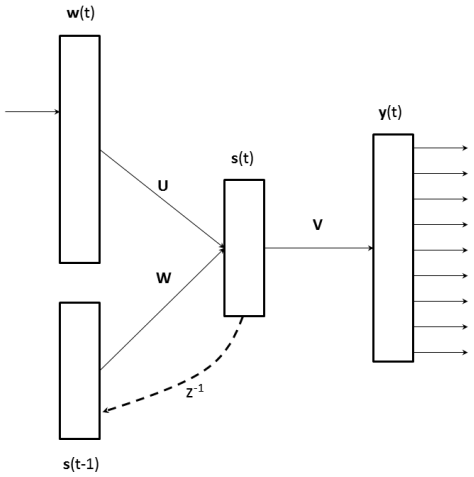

Figure 1: Recurrent Neural Network Language Model.

$$\mathbf{s}(t) = f\left(\mathbf{U}\mathbf{w}(t) + \mathbf{W}\mathbf{s}(t-1)\right) \quad (1)$$

$$\mathbf{y}(t) = g\left(\mathbf{V}\mathbf{s}(t)\right), \quad (2)$$

where

$$f(z) = \frac{1}{1+e^{-z}}, \quad g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}}. \quad (3)$$

In this framework, the word representations are found in the columns of $\mathbf{U}$, with each column representing a word. The RNN is trained with backpropagation to maximize the data log-likelihood under the model. The model itself has no knowledge of syntax or morphology or semantics. Remarkably, training such a purely lexical model to maximize likelihood will induce word representations with striking syntactic and semantic properties.

4 Measuring Linguistic Regularity

4.1 A Syntactic Test Set

To understand better the syntactic regularities which are inherent in the learned representation, we created a test set of analogy questions of the form "*a* is to *b* as *c* is to ___" testing base/comparative/superlative forms of adjectives; singular/plural forms of common nouns; possessive/non-possessive forms of common nouns; and base, past and 3rd person present tense forms of verbs. More precisely, we tagged 267M words of newspaper text with Penn

Category	Relation	Patterns Tested	# Questions	Example
Adjectives	Base/Comparative	JJ/JJR, JJR/JJ	1000	good:better rough:___
Adjectives	Base/Superlative	JJ/JJS, JJS/JJ	1000	good:best rough:___
Adjectives	Comparative/ Superlative	JJS/JJR, JJR/JJS	1000	better:best rougher:___
Nouns	Singular/Plural	NN/NNS, NNS/NN	1000	year:years law:___
Nouns	Non-possessive/ Possessive	NN/NN_POS, NN_POS/NN	1000	city:city's bank:___
Verbs	Base/Past	VB/VBD, VBD/VB	1000	see:saw return:___
Verbs	Base/3rd Person Singular Present	VB/VBZ, VBZ/VB	1000	see:sees return:___
Verbs	Past/3rd Person Singular Present	VBD/VBZ, VBZ/VBD	1000	saw:sees returned:___

Table 1: Test set patterns. For a given pattern and word-pair, both orderings occur in the test set. For example, if "see:saw return:___" occurs, so will "saw:see returned:___".

Treebank POS tags (Marcus et al., 1993). We then selected 100 of the most frequent comparative adjectives (words labeled JJR); 100 of the most frequent plural nouns (NNS); 100 of the most frequent possessive nouns (NN_POS); and 100 of the most frequent base form verbs (VB). We then systematically generated analogy questions by randomly matching each of the 100 words with 5 other words from the same category, and creating variants as indicated in Table 1. The total test set size is 8000. The test set is available online. [1]

4.2 A Semantic Test Set

In addition to syntactic analogy questions, we used the SemEval-2012 Task 2, *Measuring Relation Similarity* (Jurgens et al., 2012), to estimate the extent to which RNNLM word vectors contain semantic information. The dataset contains 79 fine-grained word relations, where 10 are used for training and 69 testing. Each relation is exemplified by 3 or 4 gold word pairs. Given a group of word pairs that supposedly have the same relation, the task is to order the target pairs according to the *degree* to which this relation holds. This can be viewed as another analogy problem. For example, take the *Class-Inclusion:Singular_Collective* relation with the prototypical word pair *clothing:shirt*. To measure the degree that a target word pair *dish:bowl* has the same relation, we form the analogy "*clothing* is to *shirt* as *dish* is to *bowl*," and ask how valid it is.

[1] http://research.microsoft.com/en-us/projects/rnn/default.aspx

5 The Vector Offset Method

As we have seen, both the syntactic and semantic tasks have been formulated as analogy questions. We have found that a simple vector offset method based on cosine distance is remarkably effective in solving these questions. In this method, we assume relationships are present as vector offsets, so that in the embedding space, all pairs of words sharing a particular relation are related by the same constant offset. This is illustrated in Figure 2.

In this model, to answer the analogy question *a:b c:d* where d is unknown, we find the embedding vectors x_a, x_b, x_c (all normalized to unit norm), and compute $y = x_b - x_a + x_c$. y is the continuous space representation of the word we expect to be the best answer. Of course, no word might exist at that exact position, so we then search for the word whose embedding vector has the greatest cosine similarity to y and output it:

$$w^* = argmax_w \frac{x_w y}{\|x_w\|\|y\|}$$

When d is given, as in our semantic test set, we simply use $\cos(x_b - x_a + x_c, x_d)$ for the words

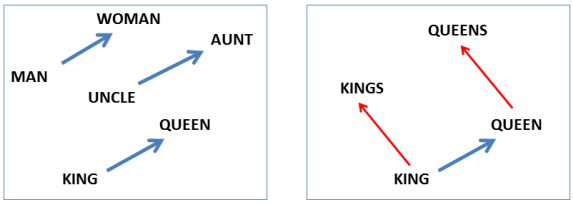

Figure 2: Left panel shows vector offsets for three word pairs illustrating the gender relation. Right panel shows a different projection, and the singular/plural relation for two words. In high-dimensional space, multiple relations can be embedded for a single word.

provided. We have explored several related methods and found that the proposed method performs well for both syntactic and semantic relations. We note that this measure is qualitatively similar to relational similarity model of (Turney, 2012), which predicts similarity between members of the word pairs $(x_b, x_d), (x_c, x_d)$ and dis-similarity for (x_a, x_d).

6 Experimental Results

To evaluate the vector offset method, we used vectors generated by the RNN toolkit of Mikolov (2012). Vectors of dimensionality 80, 320, and 640 were generated, along with a composite of several systems, with total dimensionality 1600. The systems were trained with 320M words of Broadcast News data as described in (Mikolov et al., 2011a), and had an 82k vocabulary. Table 2 shows results for both RNNLM and LSA vectors on the syntactic task. LSA was trained on the same data as the RNN. We see that the RNN vectors capture significantly more syntactic regularity than the LSA vectors, and do remarkably well in an absolute sense, answering more than one in three questions correctly. [2]

In Table 3 we compare the RNN vectors with those based on the methods of Collobert and Weston (2008) and Mnih and Hinton (2009), as implemented by (Turian et al., 2010) and available online [3] Since different words are present in these datasets, we computed the intersection of the vocabularies of the RNN vectors and the new vectors, and restricted the test set and word vectors to those. This resulted in a 36k word vocabulary, and a test set with 6632

[2]Guessing gets a small fraction of a percent.

[3]http://metaoptimize.com/projects/wordreprs/

Method	Adjectives	Nouns	Verbs	All
LSA-80	9.2	11.1	17.4	12.8
LSA-320	11.3	18.1	20.7	16.5
LSA-640	9.6	10.1	13.8	11.3
RNN-80	9.3	5.2	30.4	16.2
RNN-320	18.2	19.0	45.0	28.5
RNN-640	21.0	25.2	54.8	34.7
RNN-1600	**23.9**	**29.2**	**62.2**	**39.6**

Table 2: Results for identifying syntactic regularities for different word representations. Percent correct.

Method	Adjectives	Nouns	Verbs	All
RNN-80	**10.1**	8.1	**30.4**	**19.0**
CW-50	1.1	2.4	8.1	4.5
CW-100	1.3	4.1	8.6	5.0
HLBL-50	4.4	5.4	23.1	13.0
HLBL-100	7.6	**13.2**	30.2	18.7

Table 3: Comparison of RNN vectors with Turian's Collobert and Weston based vectors and the Hierarchical Log-Bilinear model of Mnih and Hinton. Percent correct.

questions. Turian's Collobert and Weston based vectors do poorly on this task, whereas the Hierarchical Log-Bilinear Model vectors of (Mnih and Hinton, 2009) do essentially as well as the RNN vectors. These representations were trained on 37M words of data and this may indicate a greater robustness of the HLBL method.

We conducted similar experiments with the semantic test set. For each target word pair in a relation category, the model measures its relational similarity to each of the prototypical word pairs, and then uses the average as the final score. The results are evaluated using the two standard metrics defined in the task, Spearman's rank correlation coefficient ρ and MaxDiff accuracy. In both cases, larger values are better. To compare to previous systems, we report the average over all 69 relations in the test set.

From Table 4, we see that as with the syntactic regularity study, the RNN-based representations perform best. In this case, however, Turian's CW vectors are comparable in performance to the HLBL vectors. With the RNN vectors, the performance improves as the number of dimensions increases. Surprisingly, we found that even though the RNN vec-

Method	Spearman's ρ	MaxDiff Acc.
LSA-640	0.149	0.364
RNN-80	0.211	0.389
RNN-320	0.259	0.408
RNN-640	0.270	0.416
RNN-1600	**0.275**	**0.418**
CW-50	0.159	0.363
CW-100	0.154	0.363
HLBL-50	0.149	0.363
HLBL-100	0.146	0.362
UTD-NB	0.230	0.395

Table 4: Results in measuring relation similarity

tors are not trained or tuned specifically for this task, the model achieves better results (RNN-320, RNN-640 & RNN-1600) than the previously best performing system, UTD-NB (Rink and Harabagiu, 2012).

7 Conclusion

We have presented a generally applicable vector offset method for identifying linguistic regularities in continuous space word representations. We have shown that the word representations learned by a RNNLM do an especially good job in capturing these regularities. We present a new dataset for measuring syntactic performance, and achieve almost 40% correct. We also evaluate semantic generalization on the SemEval 2012 task, and outperform the previous state-of-the-art. Surprisingly, both results are the byproducts of an unsupervised maximum likelihood training criterion that simply operates on a large amount of text data.

References

Y. Bengio, R. Ducharme, Vincent, P., and C. Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Reseach*, 3(6).

Y. Bengio, H. Schwenk, J.S. Senécal, F. Morin, and J.L. Gauvain. 2006. Neural probabilistic language models. *Innovations in Machine Learning*, pages 137–186.

A. Bordes, X. Glorot, J. Weston, and Y. Bengio. 2012. Joint learning of words and meaning representations for open-text semantic parsing. In *Proceedings of 15th International Conference on Artificial Intelligence and Statistics*.

R. Collobert and J. Weston. 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.

S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(96).

J.L. Elman. 1991. Distributed representations, simple recurrent networks, and grammatical structure. *Machine learning*, 7(2):195–225.

G.E. Hinton and R.R. Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

G. Hinton and R. Salakhutdinov. 2010. Discovering binary codes for documents by learning deep generative models. *Topics in Cognitive Science*, 3(1):74–91.

G.E. Hinton. 1986. Learning distributed representations of concepts. In *Proceedings of the eighth annual conference of the cognitive science society*, pages 1–12. Amherst, MA.

David Jurgens, Saif Mohammad, Peter Turney, and Keith Holyoak. 2012. Semeval-2012 task 2: Measuring degrees of relational similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics (SemEval 2012)*, pages 356–364. Association for Computational Linguistics.

Hai-Son Le, I. Oparin, A. Allauzen, J.-L. Gauvain, and F. Yvon. 2011. Structured output layer neural network language model. In *Proceedings of ICASSP 2011*.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19(2):313–330.

Tomas Mikolov, Martin Karafiat, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of Interspeech 2010*.

Tomas Mikolov, Anoop Deoras, Daniel Povey, Lukas Burget, and Jan Cernocky. 2011a. Strategies for Training Large Scale Neural Network Language Models. In *Proceedings of ASRU 2011*.

Tomas Mikolov, Stefan Kombrink, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2011b. Extensions of recurrent neural network based language model. In *Proceedings of ICASSP 2011*.

Tomas Mikolov. 2012. RNN toolkit.

A. Mnih and G.E. Hinton. 2009. A scalable hierarchical distributed language model. *Advances in neural information processing systems*, 21:1081–1088.

F. Morin and Y. Bengio. 2005. Hierarchical probabilistic neural network language model. In *Proceedings of the*

international workshop on artificial intelligence and statistics, pages 246–252.

J.B. Pollack. 1990. Recursive distributed representations. *Artificial Intelligence*, 46(1):77–105.

Bryan Rink and Sanda Harabagiu. 2012. UTD: Determining relational similarity using lexical patterns. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics (SemEval 2012)*, pages 413–418. Association for Computational Linguistics.

Holger Schwenk. 2007. Continuous space language models. *Computer Speech and Language*, 21(3):492 – 518.

J. Turian, L. Ratinov, and Y. Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of Association for Computational Linguistics (ACL 2010)*.

P.D. Turney. 2012. Domain and function: A dual-space model of semantic relations and compositions. *Journal of Artificial Intelligence Research*, 44:533–585.