TOWARDS DECOMPOSED LINGUISTIC REPRESENTATION WITH HOLOGRAPHIC REDUCED REPRESENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

The vast majority of neural models in Natural Language Processing adopt a form of structureless distributed representations. While these models are powerful at making predictions, the representational form is rather crude and does not provide insights into linguistic structures. In this paper we introduce novel language models with representations informed by the framework of Holographic Reduced Representation (HRR). This allows us to inject structures directly into our word-level and chunk-level representations. Our analyses show that by using HRR as a structured compositional representation, our models are able to discover crude linguistic roles, which roughly resembles a classic division between syntax and semantics.

1 INTRODUCTION

Recent advances in representation learning have been unequivocally led by the long strides of progress in deep learning and its distributed representations. In many tasks of Natural Language Processing (NLP), researchers have convincingly shown that distributed representations are capable of encoding the complex structure of textual inputs (for example Mikolov et al. (2010; 2013); Józefowicz et al. (2016); Sutskever et al. (2014)). The dominant approach for many NLP tasks is the encoder-decoder paradigm that uses neural networks to learn the transformations from many smaller comprising units to one complex embedding, and vice versa. The underlying structure, in a rather crude fashion, is assumed to be represented by this complex embedding. In many cases, such crude way of representing the structure is unsatisfactory, due to a lack of transparency, interpretability or transferability. On account of shortcomings, much previous work has been devoted to inducing disentangled representations.

We attempt to address these issues by utilizing a more principled framework to encode complex symbolic structures using distributed representations. Specifically, we employ Holographic Reduced Representation (HRR) to represent and manipulate structures. As a member of the Vector Symbolic Architecture (VSA) family (Gayler, 2003; Smolensky, 1990; Plate, 1995; Kanerva, 2009), HRR builds upon the notions of roles and fillers (i.e., values for the roles). For instance, with semantic roles, the sentence *John loves his mom* can be represented by three role-filler pairs, namely (`agent`, *John*), (`predicate`, *loves*), and (`patient`, *his mom*). Each role and filler is represented by a high-dimensional vector, and HRR provides a mathematical framework to encode role-filler pairs, compose complex embeddings, and retrieve fillers given corresponding roles. A disentangled representation, using HRR terminology, is synonymous with decomposing a complex embedding into many role-filler pairs.

In this paper, we investigate the effectiveness of HRR at inducing disentangled representations on the task of language modeling (LM). We applied HRR to language modeling because it requires minimal supervision, and has been proven hugely beneficial for many other NLP tasks. The versatility of language modeling demonstrates that some linguistic regularities much be present, and the training signal is sufficient for them to arise. We carefully design a language model with HRR that explicitly encodes the underlying structure as role-filler pairs on both word-level and chunk-level, and show that HRR provides an inductive bias towards the learning of decomposed representations. We demonstrate that on both Penn Treebank (PTB) and a subset of One-Billion-Word LM data set

(1B), our model can effectively separate certain aspects of word or chunk representation, which roughly corresponds to a division between syntax and semantics. We perform various analyses on the learned embeddings, and validate that they indeed capture distinct linguistic regularities.

Our papers is structured as follows. Section 2 gives a background overview of VSA and HRR; Section 3 details our proposed models; Experimental results are shown in Section 4, followed by related work in Section 5 and a conclusion in Section 6.

2 BACKGROUND

Vector Symbolic Architecture (VSA) is a family of models that enable connectionist models to perform symbolic processing, while encoding complex structures in distributed representations. A set of algebraic operations defined by these approaches allow them to compose, decompose and manipulate symbolic structures.

Our paper focuses on such approach, namely Holographic Reduced Representation (HRR) proposed by (Plate, 1995). HRR use three operations circular convolution, circular correlation and element-wise addition, to perform encoding, decoding and composition, respectively. The cicircular convolution (denoted by the operator $\circledast$) of two vectors $\mathbf{x}$ and $\mathbf{y}$ of dimension d, is defined as $\mathbf{z} = \mathbf{x} \circledast \mathbf{y}$, in which

$$z_i = \sum_{k=1}^{d} x_{i \bmod d}\, y_{(i-k) \bmod d}, \quad i = 1 \dots d$$

$\mathbf{x} \circledast \mathbf{y}$ is called the binding of $\mathbf{x}$ and $\mathbf{y}$, or the encoding of the pair $(\mathbf{x}, \mathbf{y})$. Using this operation, the composition of a set of role-filler pairs $(\mathbf{r}_1, \mathbf{f}_1), (\mathbf{r}_2, \mathbf{f}_2), \dots, (\mathbf{r}_m, \mathbf{f}_m)$ is represented as $\mathbf{H} = \mathbf{r}_1 \circledast \mathbf{f}_1 + \mathbf{r}_2 \circledast \mathbf{f}_2 + \dots \mathbf{r}_m \circledast \mathbf{f}_m$, where + is element-wise addition used as a composition operator. The previous example *John loves his mom* can be represented as

$$\mathbf{r}_{agent} \circledast \mathbf{f}_{John} + \mathbf{r}_{predicate} \circledast \mathbf{f}_{loves} + \mathbf{r}_{patient} \circledast \mathbf{f}_{his\ mom}.$$

By definition, HRR guarantees that the composed representation remains a vector of dimension d, regardless of how many items are bound together by the $\circledast$ operation. This avoids the parameter explosion problem of other VSA approaches such as as Tensor Product Represenation (TPR) (Smolensky, 1990), and makes HRR a more practical choice for representing compositional structures.

To decode from an HRR and retrieve a role or filler representation, an approximate inverse operation of the circular convolution, named circular correlation, is defined as $\mathbf{t} = \mathbf{x} \circledcirc \mathbf{y}$ in which

$$t_i = \sum_{k=1}^{d} x_{i \bmod d}\, y_{(i+k) \bmod d}, \quad i = 1 \dots d \tag{1}$$

Now given a memory trace $\mathbf{z} = \mathbf{x} \circledast \mathbf{y}$, the correlation operation allows us to retrieve $\mathbf{y}$ from the cue $\mathbf{x}$ via $\mathbf{y} \approx \mathbf{x} \circledcirc \mathbf{z}$. We do not detail the exact conditions when this retrieval holds, but refer interested readers to the original paper (Plate (1995))

3 HOLOGRAPHIC REDUCED REPRESENTATION FOR LANGUAGE MODELING

We incorporate HRR into language models on two levels: word level and chunk level. Our HRR-enabled language models (HRRLM) posit an explicit decomposition of word or chunk representations, which enables our model to capture different aspects of linguistic regularities. Before delving into the details of our model, we first introduce notations and provide a brief account of the commonly used RNN-based LM.

3.1 RNNLM

RNN-based LMs estimate the probability of any given sentence $s = w_1 w_2 \dots w_n$ using an RNN. At each step t, RNN encodes the history $w_1 w_2 \dots w_t$ into a vector h, and tries to predict the next token

w_t. Prediction is generally modeled by a linear layer followed by softmax operation. Specifically,

$$\Pr(w_t|w_1,\ldots,w_{t-1}) = \frac{\exp(score(h, E(w_t)))}{\sum_{w'\in V}\exp(score(h, E(w')))}, \tag{2}$$

$$score(h, E(w)) = h \cdot E(w), \tag{3}$$

where $E(\cdot)$ is the embedding operation to embed a symbol into a continuous space $\mathbb{R}^d$, and V is the entire or a sampled subset of the vocabulary. The scoring function is defined by dot product, and therefore maximizing this probability encourages related words to form clusters, and away from other words in the embedded space.

3.2 WORD-LEVEL HRRLM

Encoding We first use HRR to directly encode the underlying structures of words. We assume there is a decomposition of representations along N directions. Specifically, we embed a word w as

$$\tilde{E}(w) = \sum_{i=1}^{N} \mathbf{r}_i^{word} \circledast \tilde{E}_i(w), \tag{4}$$

where $\mathbf{r}_i^{word}$'s are *basis* role embeddings, shared by all words. Each basis role embedding is bound to its distinct set of filler embeddings, modeled by $\tilde{E}_i$. The motivation is that when properly trained, different bindings should capture disparate aspects of word representation. For instance, the first binding might be relevant for syntactic categories, and the second one for semantic relatedness. With this particular decomposition, the word *getting* should be close to other gerunds such as *giving* and *forgetting* in the first embedding space, and *get*, *got* or *received* in the second. In this case, the composite (i.e., the sum) of these bindings essentially encodes *getting* $= \{\texttt{semantics}\text{: GET}, \texttt{syntax}\text{: GERUND}\}$.

We additionally assume that each set of filler embeddings $\tilde{E}_i$ resides in a separate linear subspace. This is achieved by modeling each $\tilde{E}_i(w)$ with a linear combination of its associated *basis* filler embeddings $\mathbf{f}_{i,j}$. Specifically,

$$\tilde{E}_i(w) = \sum_{j=1}^{d'} s_{i,j}^w \mathbf{f}_{i,j} = [\mathbf{f}_{i,1}; \mathbf{f}_{i,2}; \ldots; \mathbf{f}_{i,d'}] \begin{bmatrix} s_{i,1}^w \\ s_{i,2}^w \\ \vdots \\ s_{i,d'}^w \end{bmatrix} = \mathbf{F}_i s_i^w$$

where $s_i^w \in \mathbb{R}^{d'}$ is a word-specific d'-dimensional vector. In other words, $\mathbf{F}_i$ projects $s_i^w \in \mathbb{R}^{d'}$ to $\tilde{E}_i(w) \in \mathbb{R}^d$. This assumption has two advantages. First, the total number of parameters for the embedding layer is now VNd'. We can set a smaller value d' to prevent overparameterization, while maintaining a d-dimensional vector as the input to RNN. Second, by having separate bases for different role-filler bindings, we introduce an inductive bias for the model to learn a decomposition of word representation. Our preliminary experiments show that this separation is essential for obtaining decomposed representations. The entire encoding operation is illustrated in the bottom half of Figure 1(b).

Decoding The composite embedding $E(w)$ is fed as input to RNN. From its output h given by the top hidden layer, we decode all the filler vectors using circular correlation, and factorize the scoring function into N parts. Specifically,

$$f_i = \mathbf{r}_i^{word} \circledcirc\!\!\!\!\# \, h,$$

$$score(h, w) = \sum_{i=1}^{N} \alpha_i \Big[f_i \cdot \tilde{E}_i(w) \Big] \tag{5}$$

where α_i's are scalar hyperparameters that we use to break the symmetry of the scoring function. Specifically when $N = 2$, α_1 is set to a constant 1.0, and α_2 is linearly annealed from 0.0 at the start of training to 1.0 at a specified time step T, then remain constant afterwards. Note that dot products are only computed between co-indexed filler embeddings, namely between f_i and $\tilde{E}_i(w)$. This ensures the model only learns relatedness in the i-th subspace, without interference from other subspaces. The entire word-level HRRLM is illustrated is Figure 1(b).

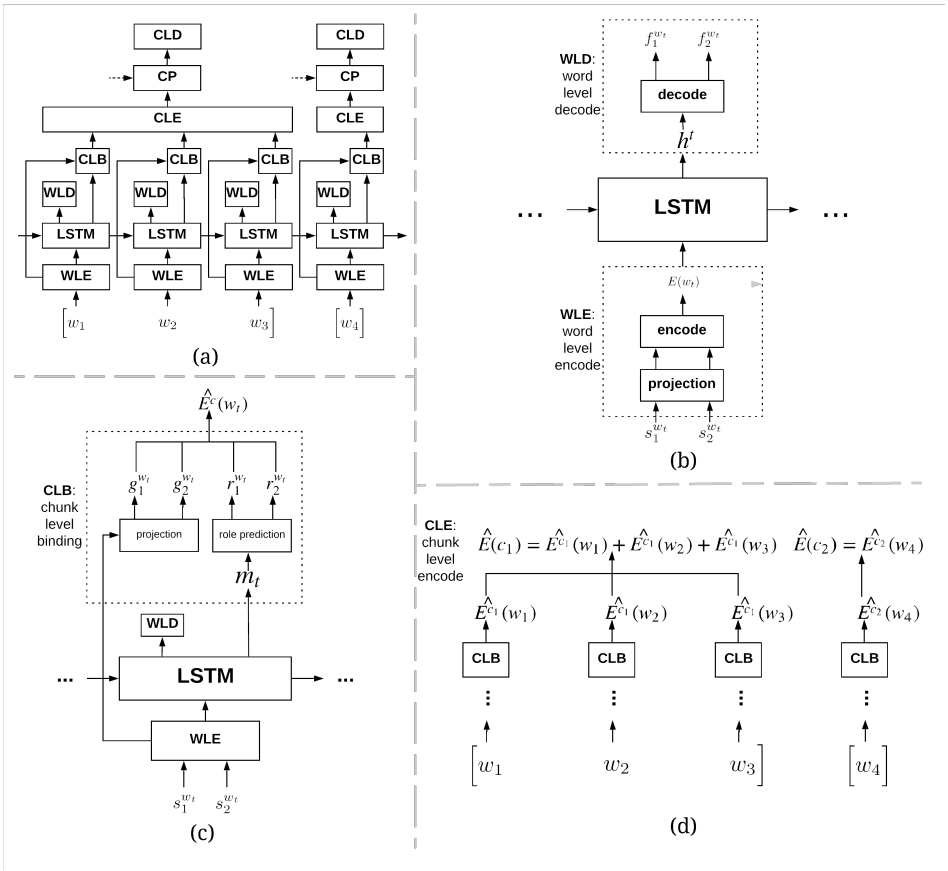

Figure 1: Architecture of HRRLMs. (a) Chunk-level HRRLM. (b)Word-level HRRLM. (c) Step 1 and 2 of chunk-level model encoding. (d) Step 3 of chunk-level model encoding. See Section 3 for details.

Regularization on basis embeddings Basis embeddings are chosen so that they are not correlated with each other. Therefore we add an isometric regularization term $\|F_i^\top F_i - \mathbf{I}\|^2$ to the fillers to promote orthonormality, where $\mathbf{I}$ is a unit matrix. Similarly, we add a regularization term for $\mathbf{r}_i^{word}$'s. As an alternative, we also consider using fixed random vectors for basis embeddings since high-dimensional random vectors are approximately orthogonal to each other.

3.3 CHUNK-LEVEL HRRLM

A direct extension of the model from word to chunk level is not straightforward, due to two major difficulties. First, Equation (4) stipulates that each unique work token is assigned a vectorial parameter. However, this is computationally infeasible due to the vast number of unique chunks. Second, it is known that for languages with a poor case system, the same phrase can carry different semantic roles, without being morphologically marked. For instance, the sentences *His mom loves John* and *John loves his mom* have the same noun phrases, but with their roles of agent and patient switched. For English, there is no information from the comprising words that can convey this difference.

Encoding Unlike the word-level HRR representation in Equation (4) where roles are fixed, the chunk roles are represented by linear combinations of M *basis* role embeddings. This formulation allows us to model context-dependency. Specifically, we construct chunk-level HRR representations in three steps:

Step 1: Within a chunk $c = w_1 w_2 \dots w_m$, we predict a tuple for each word w in the chunk:

$$r^w = (r_1^w, r_2^w, \dots, r_M^w) = (a_1^w \mathbf{r}_1^{chunk}, a_2^w \mathbf{r}_2^{chunk}, \dots, a_M^w \mathbf{r}_M^{chunk})$$

in which $\mathbf{r}_i^{chunk}$ are basis role embeddings for chunks, shared by all words, and a_i^w's are predicted by the same RNN used to predict the next token. This is done by splitting output vectors from RNN into two parts, and the second part is used to predict role weights (m_t in Figure 1(c)).

Step 2: For each of the basis roles, we predict their associated filler embeddings. This is done by projecting the HRR word representation $\tilde{E}(w)$ into M vectors. These fillers are then bound with their corresponding roles and then summed together. Specifically,

$$g_1^{w_1}, g_2^w, \ldots, g_M^w = [W_1; W_2; \ldots; W_M]^\top \tilde{E}(w)$$
$$\hat{E}^c(w) = \sum_{i=1}^{M} (a_i^w \mathbf{r}_i^{chunk}) \circledast g_i^w$$

Note that the binding $\hat{E}^c(w)$ embeds the word w into a space that is specific to the chunk c. The first two steps are illustrated in Figure 1(c).

Step 3: After obtaining binding for all words within a chunk, the chunk embedding is defined as $\hat{E}(c) = \sum_{k=1}^{m} \hat{E}^c(w_k)$ (Figure 1(d)). It can be easily verified that

$$\hat{E}(c) = \sum_{i=1}^{M} \mathbf{r}_i^{chunk} \circledast \left[\sum_{k=1}^{m} a_i^{w_k} W_i^\top \tilde{E}(w_k) \right] = \sum_{i=1}^{M} \mathbf{r}_i^{chunk} \circledast \hat{E}_i(c) \tag{6}$$

Note that $\hat{E}(c)$ has the same form as Equation (4), and $\hat{E}_i(c)$ can be interpreted as the chunk filler embedding for the i-th chunk role. However, chunk embeddings are different in two key aspects. First, the filler embeddings for chunks are projected from the composition of word embeddings, instead of being a set of independently trainable parameters. This enables utilization of the explicit chunk structures consisting of a sequence of words, and builds a complex structured embedding from embeddings of atomic units. Second, chunk embeddings rely on weights a_i^w from predictions, which provide a natural vehicle for carrying contextual information.

Prediction The chunk prediction module (CP in Figure 1(a)) predicts the next chunk embedding based on chunk history. In our experiments, we simply concatenate the chunk embeddings from the last two steps, and feed it through a linear layer followed by $\tanh$ activation.

Decoding Similar to word-level HRR, we decode the filler embeddings from the predicted chunk embedding using $\mathbf{r}_i^{chunk}$ as cue, and then use the decoded embeddings to factorize the scoring function. To provide negative examples in the denominator of the softmax Equation (2), we use all the chunks in the mini-batch. These chunks form a pseudo "chunk vocabulary" that is constructed on the fly. Role annealing, and regularization on basis embeddings are also applied.

Chunk boundaries Chunk boundaries have to be supplied in order to construct a chunk embedding. We reply on a third-party chunker to provide such annotations.

4 EXPERIMENTS

4.1 SETUP

Data Sets We train and evaluate all models on Penn Treebank (PTB) (Marcus et al., 1994) and report perplexity on the test set. We additionally use the Semantic-Syntactic Word Relationship test set released by ((Mikolov et al., 2013)) to perform word analogy task.[1] It contains 8869 semantic and 10675 syntactic questions, categorized into 14 distinct types of relations. Examples with unknown words are skipped for evaluation.

Baseline We report the results for our models against a standard RNNLM baseline. We use the same architecture for all models on both LM data sets, with a single-layer LSTM (Hochreiter & Schmidhuber (1997)) and tied input and output embeddings.

[1] Available at `www.fit.vutbr.cz/~imikolov/rnnlm/word-test.v1.txt`.

Model	Word	Chunk
Baseline	100.5	-
Isometric-50F	100.9	107.0
Fixed-50F	103.9	110.5
Isometric-100F	95.5	106.7
Fixed-100F	97.5	109.9
Isometric-250F	92.7	107.6
Fixed-250F	92.4	109.1

Table 1: Perplexity on PTB test set. 50F means we use 50 basis word-level filler embeddings.

Training details We use ADAM (Kingma & Ba (2014)) with an initial learning rate of 0.002 to train all models. The hidden size for LSTM and the word embedding size are both 512, and all parameters are uniformly initialized in (-0.08, 0.08). A dropout rate of 0.5 is used for PTB.

For our models, We experimented with two word-level roles and two chunk-level roles. We experimented with different number basis word-level fillers. Additionally, the basis embeddings are either trained with isometric constraint, with hyperparameter set to 100, or fixed as constant after random initialization. We used a third-party chunker to provide chunk boundaries for the entire PTB data set Daelemans & Van den Bosch (2005).[2]

We also note that for PTB, we do not assume that the contiguous sentences in the raw data are fed sequentially as input. For this reason, in contrast to previous work, we do not initialize the hidden state of LSTM with the last state from the last batch. This is done to assure that the chunk-level models only consider intra-sentential information, without additional signal from other sentences.

4.2 PERPLEXITY RESULTS

Although not a prime motivation of our approach, Table 1 shows that our HRR models significantly outperform the baseline. Our best word-level HRRLM outperforms the baseline on PTB by about 8.0 points in perplexity. We do note that chunk-level HRRLM seems to perform worse than word-level HRRLM. We also note that the increasing the number of basis fillers has a positive impact on word-level models, but produces a mixed result for chunk-level models. We also find that using fixed random bases yields roughly the same performance as trained bases with isometric constraints.

4.3 WORD LEVEL ANALYSIS

We then demonstrate that our word-level HRRLM can effectively separate certain aspects of word representation. Specifically, we look at our word-level models with two word-level roles, and find that the first set of filler embeddings captures mostly syntax-related categories, especially tense and agreement for verbs , whereas the second set focuses more on the semantic content of words. This decomposition is illustrated in Figure 2, where we visualize both sets of filler embeddings for the most frequent 2500 words in PTB via t-SNE Maaten & Hinton (2008). In the first set (in blue), verbs with different inflectional markers form distinct clusters. For instance, bare forms, gerunds, preterites, and third-person-singular verbs each form a visually distinguishable group. In contrast, for the second set of filler embeddings (in orange) , semantically related words tend to be close regardless of their morphosyntactic markers. For instance, *give*, *gave*, and *giving* are close in the second space.

We quantitatively investigate the quality of learned embeddings by evaluating the baseline and our word level on word analogy task. Table 2 shows that NF250 scores 0.271 overall vs 0.233 for baseline. A breakdown of the test categories reveals that NF250 does noticeably better than baseline in verb-related categories such as past tense and plural verbs. This is consistent with our previous observation that the first set of filler embeddings effectively captures the representation of different verb forms.

To further quantify how decomposed are our word-level filler embeddings, we use them to classify whether two words are in the same category. Specifically, we use a threshold θ for cosine

[2] `https://www.clips.uantwerpen.be/pages/MBSP`

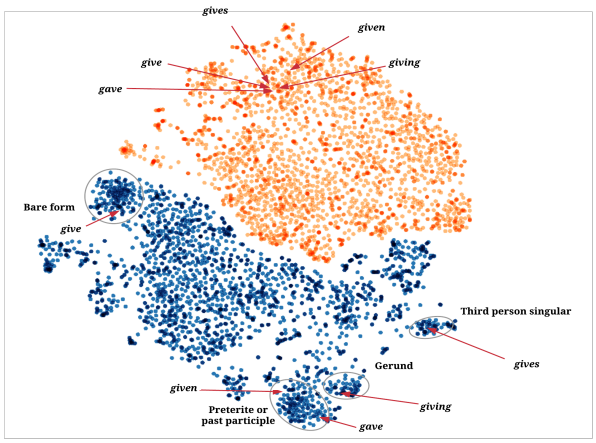

Figure 2: t-SNE visualizations for two sets of filler embeddings (blue and orange).

	Overall	Semantics	Syntax	Past tense	Present participle	Plural verbs
baseline	0.233	0.131	0.256	0.198	0.208	0.292
Fixed-250F	0.271	0.164	0.294	0.312	0.371	0.371

Table 2: Top 10 accuracy for word analogy test. We provide overal score, scores for semantic and syntactic categories, as well as three categories where our model has the most gains.

similarity to determine whether any pair of words (w, w') belong in the same category. In other words, w and w' are in the same group if $\cos(w, w') > \theta$, and otherwise not. We then use different values of θ to plot ROC curve and compute its AUC. For the ground truths, we use the same semantic-syntactic data set, and extract them from each $a : b = c : d$ example. Two sets of categorizations are obtained, driven by semantics and syntax respectively. For instance, for the example *make* : *making* = *give* : *giving*, we obtain $(\{making, giving\}, \{make, give\})$ for syntactic considerations, and $(\{making, make\}, \{giving, give\})$ for semantics. As Table 3 summarizes, the first set of embeddings does much better than the second in the first experiment, while the second set is much better at the second experiment. Figure 3 shows two categories. This confirms that the decomposition does make sense on a crude syntax-semantics level.

4.4 Chunk level analysis

Evaluation at chunk-level automatically is a challenging task, we therefore perform a human analysis focused on the phrase *the company*, which is the most frequent noun phrase in PTB. We randomly

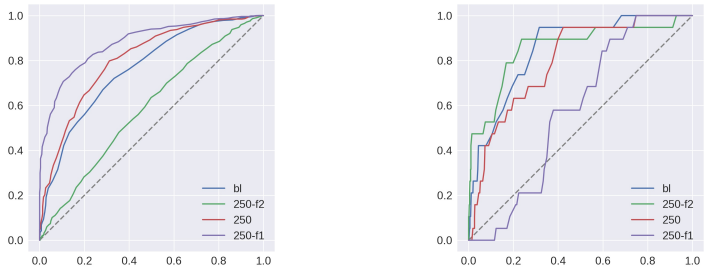

Figure 3: Examples of ROC curves for two categories. The left figure is for a syntax-driven categorization of gerunds, while the right is a semantics-driven categorization of plural verbs. bl: baseline, 250: Fixed-250F, 250-f1: first filler embedding, 250-f2: second filler embedding.

	Baseline	First filler set	Second filler set
syntactic	0.696	0.738	0.620
semantic	0.714	0.590	0.724

Table 3: Average AUC for the baseline, and also the two sets of filler embeddings from Fixed-250F.

	Role	Cluster Size	Percentage
1	Object	10	80%
2	Begin of sentence	27	100%
3	Prepositional object	25	75%
4	Subject	13	84.6%
5	Subject	26	88.5%

Table 4: Cluster analysis for sentences containing *the company*.

select 100 occurrences, and cluster their chunk embeddings using K-means into 5 categories based on the chunk-level filler embeddings.[3] For each sentence cluster we manually identify the dominating role which *the company* played in that cluster of sentences. Table 4 shows the roles we identified for each of the clusters, the number of sentences in that cluster and the percentage of sentences in which *the company* plays that role. It can be seen that there is a clear syntactic role performed by the phrase in each of the clusters. We also performed similar analysis using the filler embedding corresponding to the second chunk role for clustering. In this experiment we observed sentences in each luster contains phrases with semantics related to *the company*, for example *stock*, `market` etc. From this analysis we see that the first role in chunk-level HRR captures different syntactic roles *the company* plays (dependent on the context), while the second role captures its semantics.

5 Related Work

Perhaps mostly related to our work are recent attempts to integrate tensor product structure with neural networks (Palangi et al., 2017; Huang et al., 2018). While these work and ours share common goal of incorporating neural models with symbolic structures, there are several difference. First of all, we makes use of HRR instead of tensor product as basis for our representation to enable long sequence encoding without parameter explosion. Moreover, we aim to induce linguistic structures from a task that requires as little supervision as possible like language modeling, whereas their work is focused on question answering which provides stronger guidance signal from labeled data. On the other hand, recent work (Shen et al., 2018) also proposes a novel network architecture that is capable of learning syntactic roles and semantics jointly, but it is not based on structured representation like HRR.

There has been many attempts of using symbolic architectures like HRR and TPR for linguistic analysis, see for example (Jones & Mewhort, 2007; De Vine & Bruza, 2010; Recchia et al., 2015; Prince & Smolensky, 1997; Clark et al., 2008; Clark & Pulman, 2010; Grefenstette et al., 2011). HRR itself as a variable binding and association mechanism, has also been integrated with neural networks with different motivations like associative memory modeling (Danihelka et al., 2016), relationship reasoning (Weiss et al., 2016) etc. While most of the work mainly focuses on symbolic and formal analysis via algebraic operations and logic derivations, our work aims to enable neural language models to learn linguistic roles by taking advantage of the HRR properties.

6 Conclusion

In this paper, we employ HRR to provide a principled decomposition of representation. We design our HRR language models to work on both word-level and chunk-level. Our analysis revealed that by introducing an inductive bias, our models can learn disentangled representations, which roughly corresponds to syntax and semantics.

[3]We used the package `sklearn` for clustering, with the default parameters and settings provided by the toolbox.

REFERENCES

Stephen Clark and Stephen Pulman. Combining symbolic and distributional models of meaning. In *Proceedings of AAAI*, July 2010.

Stephen Clark, Bob Coecke, and Mehrnoosh Sadrzadeh. A compositional distributional model of meaning. 2008.

Walter Daelemans and Antal Van den Bosch. *Memory-based language processing*. Cambridge University Press, 2005.

Ivo Danihelka, Greg Wayne, Benigno Uria, Nal Kalchbrenner, and Alex Graves. Associative long short-term memory. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1986–1994, New York, New York, USA, 2016. PMLR.

Lance De Vine and Peter Bruza. Semantic oscillations: Encoding context and structure in complex valued holographic vectors. In *AAAI Fall Symposium: Quantum Informatics for Cognitive, Social, and Semantic Processes*, 2010.

Ross Gayler. Vector symbolic architectures answer jackendoff's challenges for cognitive neuroscience. In *ICCS/ASCS International Conference on Cognitive Science*, 1 2003.

Edward Grefenstette, Mehrnoosh Sadrzadeh, Stephen Clark, Bob Coecke, and Stephen Pulman. Concrete sentence spaces for compositional distributional models of meaning. In *Proceedings of the Ninth International Conference on Computational Semantics*, pp. 125–134. Association for Computational Linguistics, 2011.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Qiuyuan Huang, Paul Smolensky, Xiaodong He, Li Deng, and Dapeng Xu. Tensor product generation networks for deep nlp modeling. In *Proceedings of NAACL-HLT 2018*, 6 2018.

Michael. N Jones and Douglas. J. K. Mewhort. Representing word meaning and order information in a composite holographic lexicon. *Psychological Review*, 114:1–37, 2007.

Rafal Józefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. *CoRR*, abs/1602.02410, 2016.

Pentti Kanerva. Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive Computation*, 1(2):139–159, Jun 2009. ISSN 1866-9964.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. The penn treebank: Annotating predicate argument structure. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, pp. 114–119, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.

Tomas Mikolov, Martin Karafit, Luks Burget, Jan Cernock, and Sanjeev Khudanpur. Recurrent neural network based language model. In Takao Kobayashi, Keikichi Hirose, and Satoshi Nakamura (eds.), *Interspeech*, pp. 1045–1048, 2010.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Hamid Palangi, Paul Smolensky, Xiaodong He, and Li Deng. Deep learning of grammatically-interpretable representations through question-answering. *CoRR*, abs/1705.08432, 2017. URL `http://arxiv.org/abs/1705.08432`.

Tony Plate. Holographic reduced representations. *IEEE Transactions on Neural Network*, 6, 1995.

Alan Prince and Paul Smolensky. Optimality: From neural networks to universal grammar. *Science*, 275(5306):1604–1610, 1997.

Gabriel Recchia, Magnus Sahlgren, Pentti Kanerva, and Michael N. Jones. Encoding sequential information in semantic space models: Comparing holographic reduced representation and random permutation. *Intell. Neuroscience*, 2015:58:58–58:58, January 2015.

Yikang Shen, Zhouhan Lin, Chin-wei Huang, and Aaron Courville. Neural language modeling by jointly learning syntax and lexicon. In *Proceedings of the International Conference on Learning Representations (ICLR), Workshop Trak*, 2018.

Paul Smolensky. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46(1):159 – 216, 1990.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 27*, pp. 3104–3112. 2014.

Eric Weiss, Brian Cheung, and Bruno Olshausen. A neural architecture for representing and reasoning about spatial relationships. In *Proceedings of the International Conference on Learning Representations (ICLR), Workshop Trak*, 2016.