Which Sentence Embeddings and Which Layers Encode Syntactic Structure?

M. A. Kelly*[1], **Yang Xu**[2], **Jesús Calvillo**[1], and **David Reitter**[3,1]
(1) The Pennsylvania State University, University Park, PA, USA
(2) San Diego State University, San Diego, CA, USA
(3) Google Research, New York City, NY, USA
mak582@psu.edu, yxu4@sdsu.edu, jzc1104@psu.edu, reitter@google.com

Abstract

Recent models of language have eliminated syntactic-semantic dividing lines. We explore the psycholinguistic implications of this development by comparing different types of sentence embeddings in their ability to encode syntactic constructions. Our study uses contrasting sentence structures known to cause syntactic priming effects, that is, the tendency in humans to repeat sentence structures after recent exposure. We compare how syntactic alternatives are captured by sentence embeddings produced by a neural language model (BERT) or by the composition of word embeddings (BEAGLE, HHM, GloVe). Dative double object vs. prepositional object and active vs. passive sentences are separable in the high-dimensional space of the sentence embeddings and can be classified with a high degree of accuracy. The results lend empirical support to the modern, computational, integrated accounts of semantics and syntax, and they shed light on the information stored at different layers in deep language models such as BERT.

Keywords: syntactic priming; language models; neural networks; word embeddings; sentence embeddings

Introduction

For many natural language processing applications, there is limited data available to train the model on the specific task, often due to the high cost of annotating data. Pre-trained word embeddings are often used to address the problem of limited data. More recent efforts have focused on developing pre-trained sentence embeddings that work well on a broad range of natural language tasks (Cer et al., 2018; Conneau, Kiela, Schwenk, Barrault, & Bordes, 2017).

Understanding how the human mind represents sentences can inform the development of sentence embeddings in natural language processing models. How, exactly, the mind represents language remains an open question. However, syntactic priming (Bock, 1986) provides a window into the mind and a useful tool for validating computational representations. In turn, the ability of computational representations to account for human behaviour informs our understanding of the possible algorithms implemented by the mind.

Syntactic priming occurs when people are more likely to produce a sentence with a given structure after they have processed one with the same structure (Bock, 1986). Syntactic priming is interpreted as evidence that "some syntactic processes are organized into a functionally independent subsystem" (Bock, 1986) isolated from semantics. Conversely, we present evidence that integrated natural language processing models are compatible with syntactic priming effects.

Priming is evident not just in syntax, but also in semantics. *Semantic* priming is the finding that a word becomes more available in memory if preceded by a word with a similar meaning. The amount of semantic priming can be predicted using the distance between word embeddings generated by distributional semantic models (Günther, Dudschig, & Kaup, 2016; Jones, Kintsch, & Mewhort, 2006).

We show that distinctions evidenced for by syntactic priming can be accounted for using sentence embeddings. We build sentence embeddings by averaging the hidden state of a language model or by composing word embeddings. The language model we use is the Bidirectional Encoder Representations from Transformers (BERT; Devlin, Chang, Lee, & Toutanova, 2019). For word embeddings we use Global Vectors (GloVe; Pennington, Socher, & Manning, 2014), the Bound Encoding of the Aggregate Language Environment (BEAGLE; Jones & Mewhort, 2007), and the Hierarchical Holographic Model (HHM; Kelly, Reitter, & West, 2017).

We compare four techniques for combining word embeddings into a sentence embedding. Summing word embeddings is the most common, but does not preserve word order. We also investigate two order-preserving techniques based on permutation and one based on convolution.

In what follows, we demonstrate that different kinds of sentence structures, namely, dative double object sentences versus dative prepositional object sentences and passive versus active voice, are separable in the high-dimensional space of the sentence embeddings and can be classified with a high degree of accuracy. Our results illustrate that syntactic priming is compatible with modern, computational, integrated accounts of semantics and syntax.

Data Collection

Bock (1986) first demonstrated syntactic priming on dative double object (DO) versus dative prepositional object sentences (PO) and active versus passive voice. Accordingly, we use two data sets: (1) sentences from a syntactic priming experiment on PO versus DO priming and (2) a corpus annotated for passive versus active voice.

In a DO sentence, the *indirect object* comes before the **direct object**. For example, "The sailor mailed *his sweetheart* **a letter**". In a PO sentence, the **direct object** comes first and the *indirect object* follows after a preposition. For example, "The sailor mailed **a letter** to *his sweetheart*".

Our data is collected using the research design from Branigan, Pickering, Liversedge, Stewart, and Urbach (1995) for written-language priming. Participants are given a partial sentence as a prompt and are asked to generate a complete sentence. At the priming stage, the sentence prompt biases participants towards producing either a PO or DO sentence. For example, "The young mother gave the car..." biases participants towards a PO completion such as "The young mother gave the car to her daughter". The DO completion "The young mother gave the car her daughter" is grammatical but nonsensical and thus unlikely to be produced.

At the target stage, participants are given an unbiased prompt (e.g., "The researcher sent...") which can elicit either a DO or PO completion (e.g., "The researcher sent his work to a colleague for review" or "The researcher sent a colleague his work for review"). When participants produce a PO prime they are more likely to produce a PO target, or, likewise, a DO target if they produced a DO prime.

Syntactic priming is indicative that in some important sense the brain's representations for PO sentences are similar to other PO sentences, and the representations for DO sentences are similar to other DO sentences (e.g., Kaschak, Kutta, & Jones, 2011; Reitter, Keller, & Moore, 2011). Thus, for our purposes, demonstrating that embeddings are able to discriminate PO and DO is sufficient to demonstrate that the models are capable of accounting for syntactic priming.

To assess the models, we use the completed sentences from the data collection experiment. Note that despite the experimental setup eliciting syntactic priming data, we do not model the priming effect, but rather the encoding of PO, DO, active, and passive sentences. What the experiment design allows us to do is to (1) collect DO and PO sentences that would be more onerous to extract from a corpus, and (2) use materials generated by speakers that reflect actual sensitivity to the priming of the PO and DO syntactic structures.

We recruited 298 participants on Amazon Mechanical Turk (AMT) and paid for participation. While we used only 68 unique sentence prompts, a total of 11 520 unique, completed sentences were produced by participants. After removing sentences with words outside of GloVe or HHM's vocabularies, we are left with 2441 PO sentences, 2607 DO sentences, and 1816 sentences that are neither (e.g., "The inventor showed me how it works"). Words outside BERT's vocabulary are replaced with an out-of-vocabulary token.

We sample data contrasting active and passive voice from the European Parliamentary corpus (Europarl; Koehn, 2005[1]). We select sentences with only one verb (though the verb may be compound, e.g., *was opened*) and contain only words in the vocabulary of both BEAGLE and GloVe, which gives us 1303 passive voice sentences and 20 124 active voice sentences. The Europarl sentences are difficult, as they tend to be long (up to 83 tokens) with many low frequency words.

Models

We use two distinct approaches to creating sentence embeddings. Our first approach is to take word embeddings generated by a distributional semantics model and combine. Our second approach is to take the hidden states of a neural language model and average over the length of the sentence.

Random

We randomly generate a unique 1024 dimensional embedding for each word by sampling from a zero-mean Gaussian distribution. The random word embeddings serve as a performance baseline, as sentence embeddings built from them are sensitive to word overlap (i.e., when sentences have words in common) but not semantics or part-of-speech. Random vectors are orthogonal in expectation, such that the representation of each unique word is highly distinctive. As such, random embeddings are more sensitive to the presence or absence of a specific word than vectors sensitive to semantics or part-of-speech, which may be advantageous for some language tasks.

BEAGLE and HHM

We use the the BEAGLE model (Jones & Mewhort, 2007) and the Hierarchical Holographic Model (HHM; Kelly et al., 2017) to generate 1024 dimensional word embeddings. HHM is an extension of BEAGLE that produces a second level of more abstract word embeddings, correlated with part-of-speech and syntactic relationships (Kelly et al., 2017). We train BEAGLE and HHM on a corpus of novels from Johns, Jones, and Mewhort (2016), with 10 238 600 sentences, 145 393 172 words and 39 076 unique words.

GloVe

We compare BEAGLE and HHM to the widely-used Global Vectors for Word Representation (GloVe; Pennington et al., 2014). GloVe embeddings are constructed by dimensional reduction on a word x word co-occurrence matrix. We use a set of 300 dimensional GloVe embeddings pre-trained on English Wikipedia and the Gigaword corpus, which combined have a total of six billion words[2]. A confound in comparing GloVe and HHM's performance is that GloVe is trained on a dataset that is 40x larger, which should give GloVe a considerable advantage. However, GloVe word embeddings are constructed by treating each sentence as an unordered bag of words. GloVe's insensitivity to word order may detrimentally affect GloVe's ability to represent syntactic structure.

How to Combine Word Embeddings

We test four techniques for combining embeddings: sum, permutation by absolute or relative position, and convolution.

Sum: The most common technique for combining word embeddings to create sentence embeddings is to sum or average the embeddings of the words that are in the sentence. For example, in "Dog bites man", the sentence embedding **s** is a sum of the word embeddings $\mathbf{w}_{word}$:

[1] Europarl corpus: `https://www.idiap.ch/dataset/tense-annotation/`

[2] Pre-trained GloVe: `https://nlp.stanford.edu/projects/glove/`

$$\mathbf{s} = \mathbf{w}_{bites} + \mathbf{w}_{dog} + \mathbf{w}_{man} \tag{1}$$

The summing technique has been in use since the first distributional semantic models (Landauer & Dumais, 1997) and is effective at summarizing the meaning of sentences, paragraphs, or documents (Mitchell & Lapata, 2010), even outperforming sentence embeddings created using neural language models if applied to tasks outside the neural model's training (Wieting, Bansal, Gimpel, & Livescu, 2016). But summing word embeddings does not preserve word order. Given the importance of word order to English syntax, a sum is unlikely to be effective at discriminating sentence structure.

Permutation by absolute position: Permutation is used in some models to encode word order (Cohen & Widdows, 2018; Recchia, Sahlgren, Kanerva, & Jones, 2015; Sahlgren, Holst, & Kanerva, 2008). The simplest approach is to generate a different permutation for each position in the sentence. For example, "Dog bites man" can be represented as:

$$\mathbf{s} = \mathbf{P}_1\mathbf{w}_{dog} + \mathbf{P}_2\mathbf{w}_{bites} + \mathbf{P}_3\mathbf{w}_{man} \tag{2}$$

where $\mathbf{P}_i$ is the permutation for the ith sentence position. A permutation is a reordering of the elements of a vector, such that the conjunction of an embedding and a permutation serves as a unique representation of a word and a position.

Cohen and Widdows (2018) generate the first permutation $\mathbf{P}_1$ randomly, then generate successive permutations by randomizing half of the prior permutation. Cohen and Widdows find that giving proximal sentence positions similar permutations improves the performance of word embeddings on syntactic analogy tasks. Thus, we use Cohen and Widdows's technique for generating permutations in what follows.

Permutation by relative position: Encoding word order by absolute position in a sentence is cognitively implausible. McCoy, Frank, and Linzen (2018) found that neural language models that learn to make predictions on the basis of absolute sentence position tend to make inhuman errors when generalizing learned grammatical rules to novel sentences. Kinder (2010) similarly finds that exemplar-based language models that use absolute position make qualitatively different judgements of grammaticality than humans.

Cohen and Widdows (2018) use a sliding window, such that positions are not relative to the start of the sentence, but relative to the position of the window. To mimic a sliding window, we permute each word embedding by each possible window position for that word in the given sentence. For example, "Dog bites man" can be represented as:

$$\begin{aligned}\mathbf{s} = {} & \mathbf{P}_0\mathbf{w}_{dog} + \mathbf{P}_1\mathbf{w}_{bites} + \mathbf{P}_2\mathbf{w}_{man} + \\ & \mathbf{P}_{-1}\mathbf{w}_{dog} + \mathbf{P}_0\mathbf{w}_{bites} + \mathbf{P}_1\mathbf{w}_{man} + \\ & \mathbf{P}_{-2}\mathbf{w}_{dog} + \mathbf{P}_{-1}\mathbf{w}_{bites} + \mathbf{P}_0\mathbf{w}_{man}\end{aligned} \tag{3}$$

where we denote the centre of the window by the permutation $\mathbf{P}_0$, window positions to the left of centre by negative indices, and window positions to the right by positive indices.

Convolution: Circular convolution ($*$) is used in holographic reduced representations to form associations (Plate, 1995). BEAGLE and HHM's word embeddings use convolution to represent word order. To construct sentence embeddings, we use the method from Jamieson and Mewhort (2011)'s model of grammaticality judgements.

Each sentence is represented as a sum of n-grams, for $n = 1$ to 20. Each n-gram is constructed as the convolution of the embeddings for the n words in the n-gram. To preserve the order of the words in the n-gram, the left operand of convolution is permuted by the permutation $\mathbf{P}_{left}$. For example, "Dog bites man" can be represented as:

$$\begin{aligned}\mathbf{s} = {} & \mathbf{w}_{dog} + \mathbf{w}_{bites} + \mathbf{w}_{man} + \\ & (\mathbf{P}_{left}\mathbf{w}_{dog}) * \mathbf{w}_{bites} + (\mathbf{P}_{left}\mathbf{w}_{bites}) * \mathbf{w}_{man} + \\ & (\mathbf{P}_{left}((\mathbf{P}_{left}\mathbf{w}_{dog}) * \mathbf{w}_{bites})) * \mathbf{w}_{man}\end{aligned} \tag{4}$$

BERT

Bidirectional Encoder Representations from Transformers (BERT; Devlin et al., 2019) is based on the transformer architecture (Vaswani et al., 2017), which has largely supplanted recurrent neural networks (RNNs) as the state-of-the-art in natural language processing (e.g., Karita et al., 2019; Zeyer, Bahar, Irie, Schlüter, & Ney, 2019). Sentence embeddings derived from transformers, such as BERT and the Universal Sentence Encoder (Cer et al., 2018), outperform sentence embeddings derived from earlier neural language models (e.g., ELMo, InferSent; Hassan, Sansonetti, Gasparetti, Micarelli, & Beel, 2019). The distinct layers of BERT models have been associated with different representations relevant to natural language processing (Tenney, Das, & Pavlick, 2019).

We use a pre-trained BERT with 12 hidden layers, each with 768 dimensions[3]. Devlin et al. (2019) trained BERT on English Wikipedia (2500 million words) and the BookCorpus (800 million words). BERT is trained on half as much data as GloVe, but 20x more data than HHM. To generate sentence embeddings, we use the BERT-As-Service tool [4]. BERT-As-Service generates sentence embeddings by averaging the hidden states at a given layer over locations in the sentence. We generate sentence embeddings for each hidden layer.

Spatial Separability

Sentence embeddings with different syntactic structures may form distinct, linearly separable clusters in the high dimensional space. To evaluate the spatial separability of the sentence embeddings, we use the sensitivity index, d', a statistical measure of the separability of a signal from noise. The sensitivity index is a function of the means, μ_S and μ_N, and standard deviations, σ_S and σ_N, of the signal and noise distributions, S and N:

[3] Pre-trained BERT: https://storage.googleapis.com/bert_models/2018_10_18/uncased_L-12_H-768_A-12.zip

[4] BERT-As-Service: https://github.com/hanxiao/bert-as-service

Table 1: Separability of PO sentences by embedding.

Models	sum	abs.	rel.	conv.
Random	0.06	0.69	1.03	0.53
GloVe	0.69	0.69	0.85	0.41
BEAGLE	0.48	0.67	**1.10**	0.78
HHM	0.00	0.68	0.94	0.79

Table 2: Separability of DO sentences by embedding.

Models	sum	abs.	rel.	conv.
Random	0.00	0.31	0.25	0.08
GloVe	0.09	0.19	0.14	0.11
BEAGLE	-0.18	0.36	0.37	0.30
HHM	0.00	0.24	0.22	**0.39**

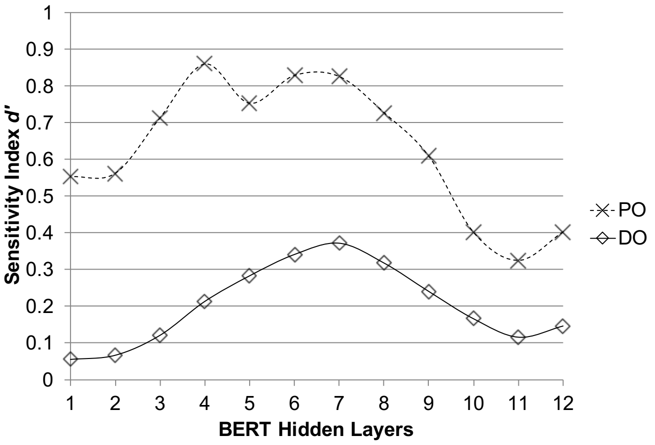

Figure 1: Separability of PO and DO sentences by averaging over BERT hidden layers.

$$d' = \frac{\mu_S - \mu_N}{\frac{1}{2}\sqrt{\sigma_S^2 + \sigma_N^2}} \quad (5)$$

We use the vector cosine as the metric of similarity between the sentence embeddings. The signal distribution is the cosine similarities between all sentences of a given type. The noise distribution is the cosine between sentences of a given type and all sentences not of that type. A sentence type with a higher d' is more distinguishable from other sentence types.

In what follows, we compare the sensitivity index of the sentence embeddings on the PO, DO, and other (non-PO, non-DO) sentences. We then select the models with the highest sensitivity to assess on the active and passive sentences.

Distributional Semantic Models

Tables 1 and 2 shows the sensitivity index for four different word embedding models on PO and DO sentences: randomly generated vectors, BEAGLE, HHM, and GloVe. For each model, we compare four ways of combining the embeddings to construct a sentence embedding: summation, permutation by absolute and relative position, and convolution.

Sum: Summing is the least effective, which is hardly surprising, as the sum does not capture the structure of a sentence, only what words occur within it. However, the ability of the sum to separate PO from non-PO (Table 1) when provided with the right word embeddings (BEAGLE or GloVe) suggests that certain words, such as the preposition *to*, make separating PO from non-PO an easy problem. The preposition *to* occurs in 100% of the PO sentences but only 6% of the DO sentences in our AMT data set.

Absolute position (abs.): Encoding word position markedly improves separability for PO and DO sentences. For randomly generated or GloVe embeddings, permuting by absolute position yields the highest sensitivity index for DO sentences (random: $d' = 0.31$, GloVe: $d' = 0.19$).

Relative position (rel.): For all models, permutation by relative position produces the sentence embeddings that most easily separate PO from non-PO. Furthermore, relative position encoding is approximately as sensitive as absolute position for discriminating between DO and non-DO when using BEAGLE or HHM. For both PO and DO sentences, the distributions are more separable when using sentence embeddings sensitive to lexical semantics (BEAGLE vs. Random, $d' = 1.10$ vs. 1.03 for PO, $d' = 0.37$ vs. 0.25 for DO).

Convolution (conv.): Using convolution to construct sentence embeddings works well when using holographic vectors (either BEAGLE or HHM). For HHM, convolution yields the highest sensitivity for DO versus non-DO ($d' = 0.39$). However, convolution works poorly with GloVe embeddings, performing even worse than summation on PO versus non-PO (convolution: $d' = 0.41$, sum: $d' = 0.69$). Convolution may be too reliant on the properties of the word embeddings (see Kelly, Blostein, & Mewhort, 2013 for discussion) and is perhaps best used with holographic vectors.

BERT

Figure 1 shows the sensitivity index for sentence embeddings constructed as an average over time at different layers of BERT. Hidden layers are numbered from 1, the layer immediately after the input, to 12, the last layer before the output.

Layers close to the middle provide the most appropriate representations to separate PO and DO sentences. Layer 7 has the highest sensitivity index for DO sentences, $d' = 0.37$, whereas layer 4 has the highest for PO sentences, $d' = 0.86$, though layer 7 also makes this discrimination well, $d' = 0.83$.

Discussion

Of the methods for composing word embeddings to create sentence embeddings that we consider here, convolution with the HHM word embeddings achieves the highest sensitive index for DO versus non-DO ($d' = 0.39$), followed closely by BERT layer 7 and BEAGLE with relative position encoding (both $d' = 0.37$). For the PO versus non-PO, BEAGLE with relative position encoding achieves the highest sensitivity index ($d' = 1.10$). Despite being trained on a corpus 40x larger, GloVe embeddings are not more sensitive to the PO and DO distinctions than HHM embeddings. For GloVe embeddings, convolution is a poor method for constructing sentence embeddings, but both permutation methods work well.

Classification

As an alternate means of assessing the ability of each type of embedding to discriminate PO and DO from non-PO and non-DO, we train a classifier using each of the best performing sentence embeddings. For each type of sentence embedding, we use five-fold cross validation to train three generalized linear model regressions: one for classifying PO versus non-PO, one for DO versus non-DO, and one for active versus passive voice. For our Amazon Mechanical Turk (AMT) data, sentences that begin with the same prompt are placed in the same fold. For the Europarl corpus, each fold is assigned an equal number of passive and active sentences, with the remaining active sentences used at test. Table 3 shows classification accuracy for the linear models. Percent correct for the PO and DO linear models is the mean of the accuracy on PO, DO, and other (non-PO, non-DO) sentences. Conversely, the third linear model's accuracy is shown as two separate columns: accuracy on active and accuracy on passive sentences.

For PO and DO, classification accuracy largely mirrors the sensitivity index of the models. However, we find that the random model gets 100% correct on PO versus non-PO classification. The result suggests that knowing the presence and location of the preposition *to* is sufficient for perfect PO classification in our AMT data set, and the addition of semantic or part-of-speech (i.e., GloVe, BEAGLE, or HHM) merely serves to add noise to the classification. Conversely, the random model is the least accurate model for DO versus non-DO classification, providing evidence that lexical semantics plays an important role in detecting the DO sentence structure.

While BERT has a lower d' than HHM with convolution on DO sentences, BERT has a higher DO accuracy. The discrepancy may arise from differences in BERT's embeddings compared to HHM's holographic vectors. Holographic vectors represent information *holographically*: all information is fully distributed across all dimensions. Conversely, in neural models, information may be distributed unevenly, such that d' may underestimate BERT's ability to classify sentences.

Classification accuracy for active and passive voice is lower than for PO and DO sentences. The best accuracy is achieved by BERT (85% correct on active sentences, 87% correct on passive sentences) followed by GloVe using relative position (83% correct on active, 78% correct on passive).

While absolute and relative position encoding work almost equally well for GloVe on the PO and DO sentences, relative position is slightly better at classifying the active and passive sentences, likely due to sentence length. The Europarl sentences are long. Knowing that a word is the 55th in a sentence may not be useful for making classification decisions.

BEAGLE and HHM perform worse than GloVe on the passive and active sentences. We suspect the lower accuracy is due to the many low frequency words in the Europarl corpus. BEAGLE and HHM are trained on much less data than BERT or GloVe, and as such, the embeddings for low frequency words are based on fewer instances and are much noisier.

Table 3: Classifier accuracy on PO versus DO and active (Act.) versus passive (Pass.) across models.

Models		PO	DO	Act.	Pass.
Random	rel.	**100%**	81%	73%	69%
GloVe	rel.	91%	86%	83%	78%
GloVe	abs.	92%	86%	79%	78%
BEAGLE	rel.	92%	88%	80%	76%
HHM	conv.	97%	93%	77%	74%
BERT layer 7	avrg.	98%	**96%**	**85%**	**87%**

General Discussion

Sentence embeddings created using either distributional semantic models or neural language models spatially separate sentences with distinct syntactic structures.

We evaluate the psycholinguistic plausibility of four methods for composing word embeddings into sentence embeddings. The most common method, taking a sum of the word embeddings, does not preserve word order, and as such, is insufficient to account for English syntactic structure. Of the three methods that do preserve word order, convolution works best with HHM embeddings. In a convolutional model, sentences are represented as a set of n-grams, where each n-gram is constructed as a convolution of word embeddings (as described in Jamieson & Mewhort, 2011; Jones & Mewhort, 2007). When using convolution, we find evidence that sensitivity to more abstract relationships between words (HHM; Kelly et al., 2017) provide additional useful information for discriminating syntactic structure. However, the advantage for HHM over BEAGLE is not robust, as it is not present when using permutation to construct sentence embeddings.

Permutation by relative position in a sliding window is consistently the best at discriminating PO from non-PO sentences. Indeed, we find that randomly generated word embeddings combined using relative position encoding are sufficient to get 100% accuracy on our AMT data set. However, the ability to discriminate DO from non-DO sentences is improved by the use of trained embeddings.

We find inconsistent performance for permutation by absolute position. While it works reasonably well on the AMT sentences, it works poorly on the long sentences of the Europarl corpus. Prior work has found that encoding word order by absolute position in a sentence is cognitively implausible (Kinder, 2010; McCoy et al., 2018). Thus we prefer permuting by relative position, which proves to be an effective and computationally efficient method of encoding word order that is robust across different word embeddings.

We use the BERT language model to construct sentence embeddings as a mean of hidden layer activation values and find that BERT yields the highest classification accuracy for sentence type. However, distributional semantic models are less computationally intensive and perform comparably well.

The best-performing BERT layer is exactly in the middle of the network. This layer is likely the most abstract, as it is the most removed from input or output. Similarly, the best performing word embeddings on PO versus DO are the HHM embeddings, which are sensitive to abstract, syntactic associations between words (Kelly et al., 2017).

Prior work has shown that semantic priming is predicted by the distance between word embeddings (Günther et al., 2016; Jones et al., 2006). Likewise, we find that the distances between sentence embeddings allow for the distinctions in syntactic structure evidenced in syntactic priming. Sentence embeddings do not require a system that processes syntax distinct from semantics. Rather, sentence embeddings can be constructed by composing word embeddings or by averaging the activation in a language model's hidden layer.

Conclusion

For the purpose of either natural language processing or modelling human behaviour: (1) More abstract representations extracted from the mid-layers of neural language models are best able to account for syntactic distinctions; (2) Combining word embeddings by permuting by relative position in a sliding window produces robust sentence embeddings sensitive to syntax; (3) The strong performance of HHM on DO sentences suggests that convolution-based approaches, warrant further investigation.

Our results suggest that more abstract representations (HHM or BERT mid-layers) are better able to make the distinctions evidenced for in syntactic priming experiments (i.e., active versus passive voice or DO versus non-DO sentences). However, some syntactic distinctions can be made trivial by representations sensitive to the presence and location of function words (e.g., PO versus non-PO). We speculate that humans use information at varying levels of abstraction in language processing, perhaps similar to a deep neural model. An overall picture emerges of integrated representations with rich connections between traditionally distinct layers (within and across languages; Putnam, Carlson, & Reitter, 2018).

We leave as a matter of future work testing the methods of word embedding composition we explore here on languages with richer morphologies or free word order. Modelling languages with rich morphologies requires either using sub-word embeddings or word embeddings sensitive to sub-word units (e.g., Cotterell & Schütze, 2015). Free word order languages typically use word order to convey non-syntactic information (e.g., emphasis or new information), such that while preserving word order may not be important for syntax *per se*, order remains important for conveying meaning.

Acknowledgments

This research was funded by a Natural Sciences and Engineering Research Council of Canada (NSERC) Post-Doctoral Fellowship to M. A. Kelly and a National Science Foundation grant (BCS-1734304) to David Reitter and M. A. Kelly.

References

Bock, J. K. (1986). Syntactic persistence in language production. *Cognitive Psychology*, *18*(3), 355 - 387. doi: https://doi.org/10.1016/0010-0285(86)90004-6

Branigan, H. P., Pickering, M. J., Liversedge, S. P., Stewart, A. J., & Urbach, T. P. (1995). Syntactic priming: Investigating the mental representation of language. *Journal of Psycholinguistic Research*, *24*(6), 489–506. doi: 10.1007/BF02143163

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., St. John, R., … Kurzweil, R. (2018). Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (pp. 169–174). Brussels, Belgium: Association for Computational Linguistics. doi: 10.18653/v1/D18-2029

Cohen, T., & Widdows, D. (2018). Bringing order to neural word embeddings with embeddings augmented by random permutations (EARP). In *Proceedings of the 22nd Conference on Computational Natural Language Learning* (pp. 465–475). Brussels, Belgium: Association for Computational Linguistics. doi: 10.18653/v1/K18-1045

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., & Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 670–680). Copenhagen, Denmark: Association for Computational Linguistics. doi: 10.18653/v1/D17-1070

Cotterell, R., & Schütze, H. (2015). Morphological word-embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 1287–1292). Denver, Colorado: Association for Computational Linguistics. doi: 10.3115/v1/N15-1140

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Vol. 1, pp. 4171–4186). Minneapolis, Minnesota: Association for Computational Linguistics. doi: 10.18653/v1/N19-1423

Günther, F., Dudschig, C., & Kaup, B. (2016). Latent semantic analysis cosines as a cognitive similarity measure: Evidence from priming studies. *Quarterly Journal of Experimental Psychology*, *69*(4), 626-653. doi: 10.1080/17470218.2015.1038280

Hassan, H. A. M., Sansonetti, G., Gasparetti, F., Micarelli, A., & Beel, J. (2019). BERT, ELMo, USE and InferSent sentence encoders: The panacea for research-paper recommendation? In *CEUR Workshop Proceedings* (Vol. 2431, p. 6–10).

Jamieson, R. K., & Mewhort, D. J. K. (2011). Grammaticality is inferred from global similarity: A reply to kinder (2010). *The Quarterly Journal of Experimental Psychology*, *64*, 209-216. doi: 10.1080/17470218.2010.537932

Johns, B. T., Jones, M. N., & Mewhort, D. J. K. (2016). Experience as a free parameter in the cognitive modeling of language. In A. Papafragou, D. Grodner, D. Mirman, & J. C. Trueswell (Eds.), *Proceedings of the 38th Annual Meeting of the Cognitive Science Society* (p. 1325-1330). Austin, TX: Cognitive Science Society.

Jones, M. N., Kintsch, W., & Mewhort, D. J. K. (2006). High-dimensional semantic space accounts of priming. *Journal of Memory and Language*, *55*, 534-552. doi: 10.1016/j.jml.2006.07.003

Jones, M. N., & Mewhort, D. J. K. (2007). Representing word meaning and order information in a composite holographic lexicon. *Psychological Review*, *114*, 1-37. doi: 10.1037/0033-295X.114.1.1

Karita, S., Chen, N., Hayashi, T., Hori, T., Inaguma, H., Jiang, Z., … Zhang, W. (2019). A comparative study on transformer vs RNN in speech applications. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop* (p. 449-456). Singapore: IEEE. doi: 10.1109/ASRU46091.2019.9003750

Kaschak, M. P., Kutta, T. J., & Jones, J. L. (2011). Structural priming as implicit learning: Cumulative priming effects and individual differences. *Psychonomic Bulletin&Review*, *18*(6), 1133–1139. doi: 10.3758/s13423-011-0157-y

Kelly, M. A., Blostein, D., & Mewhort, D. J. K. (2013). Encoding structure in holographic reduced representations. *Canadian Journal of Experimental Psychology*, *67*, 79-93. doi: 10.1037/a0030301

Kelly, M. A., Reitter, D., & West, R. L. (2017). Degrees of separation in semantic and syntactic relationships. In M. K. van Vugt, A. P. Banks, & W. G. Kennedy (Eds.), *Proceedings of the 15th International Conference on Cognitive Modeling* (p. 199-204). Warwick, U.K.: University of Warwick.

Kinder, A. (2010). Is grammaticality inferred from global similarity? Comment on Jamieson and Mewhort (2009). *The Quarterly Journal of Experimental Psychology*, *63*(6), 1049–1056. doi: 10.1080/17470211003718713

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT Summit* (Vol. 5, pp. 79–86).

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, *104*, 211-240. doi: 10.1037/0033-295X.104.2.211

McCoy, R. T., Frank, R., & Linzen, T. (2018). Revisiting the poverty of the stimulus: Hierarchical generalization without a hierarchical bias in recurrent neural networks. In T. T. Rogers, M. Rau, X. Zhu, & C. W. Kalish (Eds.), *Proceedings of the 40th Annual Conference of the Cognitive Science Society* (pp. 2093–2098). Austin, TX: Cognitive Science Society.

Mitchell, J., & Lapata, M. (2010). Composition in distributional models of semantics. *Cognitive Science*, *34*, 1388-1429. doi: 10.1111/j.1551-6709.2010.01106.x

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (pp. 1532–1543). doi: 10.3115/v1/D14-1162

Plate, T. A. (1995). Holographic reduced representations. *IEEE Transactions on Neural Networks*, *6*, 623-641. doi: 10.1109/72.377968

Putnam, M. T., Carlson, M., & Reitter, D. (2018). Integrated, not isolated: Defining typological proximity in an integrated multilingual architecture. *Frontiers in Psychology: Language Sciences*. doi: 10.3389/fpsyg.2017.02212

Recchia, G., Sahlgren, M., Kanerva, P., & Jones, M. N. (2015). Encoding sequential information in semantic space models: Comparing holographic reduced representation and random permutation. *Computational Intelligence and Neuroscience*, *2015*. doi: 10.1155/2015/986574

Reitter, D., Keller, F., & Moore, J. D. (2011). A computational cognitive model of syntactic priming. *Cognitive Science*, *35*(4), 587–637. doi: 10.1111/j.1551-6709.2010.01165.x

Sahlgren, M., Holst, A., & Kanerva, P. (2008). Permutations as a means to encode order in word space. In B. C. Love, K. McRae, & V. M. Sloutsky (Eds.), *Proceedings of the 30th Annual Conference of the Cognitive Science Society* (p. 64-70). Austin, TX: Cognitive Science Society.

Tenney, I., Das, D., & Pavlick, E. (2019, July). BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 4593–4601). Florence, Italy: Association for Computational Linguistics. doi: 10.18653/v1/P19-1452

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is all you need. In I. Guyon et al. (Eds.), *Advances in neural information processing systems 30* (pp. 5998–6008). Curran Associates, Inc.

Wieting, J., Bansal, M., Gimpel, K., & Livescu, K. (2016). Towards universal paraphrastic sentence embeddings. In Y. Bengio & Y. LeCun (Eds.), *4th International Conference on Learning Representations.* San Juan, Puerto Rico.

Zeyer, A., Bahar, P., Irie, K., Schlüter, R., & Ney, H. (2019). A comparison of transformer and LSTM encoder decoder models for ASR. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop* (p. 8-15). Singapore: IEEE. doi: 10.1109/ASRU46091.2019.9004025