You Are What You Read: The Effect of Corpus and Training Task on Semantic Absorption in Recurrent Neural Architectures

1st Nancy Fulda
Department of Computer Science
Brigham Young University
Provo, USA
nfulda@cs.byu.edu

***Abstract*—Recurrent neural networks are able to capture semantic meaning within the geometry of the resultant embedding space, but the impact of training corpus and training curriculum on the structure of the learned representations remains poorly understood. This paper sheds some light on the mystery by comparing the performance of a simple recurrent model on three training tasks and two strongly divergent training corpora. The learned representations are compared on tasks including the Semantic Textual Similarity Benchmark, the Stanford Natural Language Inference Corpus, and Google's Analogical Reasoning Test Set. Results show that context-based training produces the strongest semantic alignment within the embedding space, with reconstruction loss as an interesting close second. Pairwise comparisons of models trained on different corpora show that the choice of corpus also has powerful effects on the learned representations. Most importantly, we observe that the choice of input corpus and training task are not unilaterally independent, but instead interact with each other in interesting ways. This motivates a cautionary position against training neural models by simply throwing as many different training tasks as possible into the mix. Instead, it may be wiser to carefully select only tasks that are compatible with the chosen input corpus, and vice versa.**

***Index Terms*—natural language understanding, language modeling, semantic embeddings**

I. Introduction

Recent advances in language modeling rely on recurrent and attention-based architectures that are able to respect temporal ordering and learn high-quality representations that go beyond simple averages of component words. The mechanisms by which semantic information is reflected in the geometry of the resultant embedding space, and in particular the impact of various design choices such as training curriculum and training corpus, are not yet well understood.

The objective of this paper is to determine the impact of training corpus and training curriculum on the representations learned by a minimal recurrent model. The decision to restrict this study to a small model has two key advantages: (1) The lightweight model trains quickly, thus allowing a wider variety of experiments, (2) Because the representational power of the model is limited, the native advantages of the various corpora and training tasks are easier to identify. It is true that a more complex model may be able to compensate for the inherent challenges of a malformed task or poorly chosen training corpus, but that does not mean it is not affected by them. The impact may merely be more difficult to detect, manifesting as delays in training time and subtle imperfections in the learned representations. By forcing these limitations into the open with a small model, we hope to identify patterns that will be of value when training larger ones.

II. Related Work

Neural models that learn representations of language can be evaluated in multiple ways. One common method is to evaluate the models based on their effectiveness as pre-trained input features for downstream tasks. Bert [7], InferSent [5] and Google's Universal Sentence Encoder [4] were all evaluated using this method. However, while this evaluation structure can reveal *how well* a system is learning language data in a general sense, it cannot tell us *which aspects* of language are being encoded within the embedding space.

Because of this limitation, our work inclines in the direction of Zhu et al. [12] and Conneau et al. [6] in that we wish to explore which semantic properties are *directly* encoded within the structure of the learned representations. We therefore utilize a set of evaluation tasks that rely on cosign similarities between embedded sentences as the primary measure of semantic structure.

Our network architecture follows the example of Tong et al. [11] in that the hidden activations of a recurrent unit are averaged prior to passing through a fully connected layer, but our architecture is much simpler and designed with the intent of learning general purpose sentence representations rather than learning conversational behavior. The averaged hidden layers are also structurally similar to Iyyer et al.'s Deep Averaging Network [8]. However, our model includes a recurrent element which allows it to absorb contextual information in sequential fashion.

III. Methodology

We use the term *semantic absorption* to describe the degree to which a neural model is able to reflect generally accepted

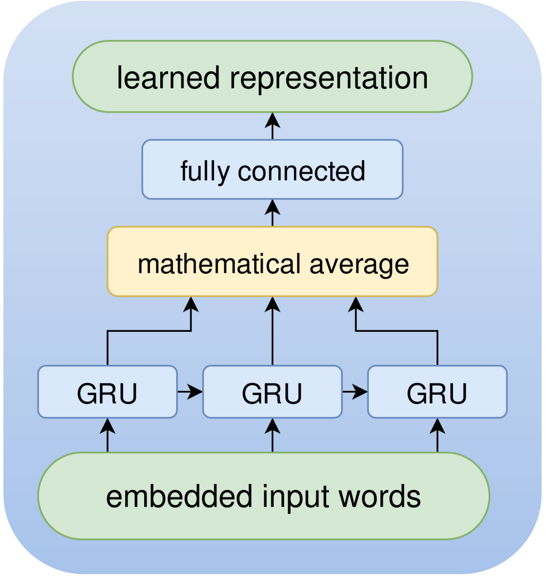

Fig. 1. Our neural model. Input words are embedded using the FastText weights [1], then fed in sequence into a Gated Recurrent Unit. Once all of the input text has passed through the GRU, the intermediate hidden states of the GRU are averaged and passed to a fully-connected layer.

semantic properties in the geometry of its learned representations.

To determine the impact of training corpus and curriculum on semantic absorption, we trained several models alternately on (a) a plain text version of Wikipedia or (b) the first half of the Toronto Book Corpus [13]. These corpora are notably different in composition and vocabulary, with Wikipedia having a larger vocabulary and generally more complex sentences, whereas the Book Corpus contains stronger temporal structure, more examples of cause and effect, and a notable emphasis on human emotion.

For each training corpus, we trained the model architecture depicted in Figure 1 on three natural language tasks:

1) **Reconstruction:** In this task, the model was trained in parallel with a simple recurrent word-level decoder. The task of the model was to learn a representation which contained enough information to reproduce the sentence exactly. The task of the decoder was to reconstruct each sentence given the learned representation. Loss for both systems was defined as the mean-squared error between the original sentence and the string of output words produced by the decoder.
2) **Context prediction:** Reminiscent of the training task first used by Skip-Thought vectors [9], a single input sentence was encoded using the recurrent model and then passed through two independent fully connected layers. The first layer attempted to predict the embedded representation of the previous sentence from the input corpus. The second layer attempted to predict the representation of the next sentence. Output representations from both prediction tasks were passed through a shared decoder architecture that decoded the predicted representations into text. Mean-squared error between the predicted and genuine sentences was again used as the network loss.
3) **Word prediction:** Given a single input word, the model was required to predict the most likely neighbors of that word. This training mechanism was inspired by the skip-gram task used by Mikolov et al. [10], but in our implementation the context window is bounded by the start and end tokens of each input sentence. On each forward pass, a randomly chosen window size of $n \in \{2,4,6\}$ is chosen. Although the task's input value is a single word, it is still passed through the entire network architecture of Figure 1 to produce its embedded representation. The embedding is then used as the input to a fully-connected layer whose outputs contain log probabilities for each word in the model's vocabulary.

Our objective in choosing these training tasks was to determine how training curriculum affects the type of semantic information absorbed by the neural network. The decision to use a small recurrent model was motivated by the ability to train quickly across multiple training tasks and input corpora, as well as by the desire to expose the inherent strengths and weaknesses of each design choice. To further facilitate training speed, only input sentences of fewer than fifty characters in length were used.

IV. Experiments

We evaluated semantic absorption by comparing model performance on a variety of natural language tasks. The objective of our experiments was not to outperform state-of-the-art methods, but rather to understand the relative effectiveness of each input corpus and training task at inducing semantic properties within the learned embedding space. By design, our evaluation metrics include both word-level and sentence-level tasks. This is because effective sentence representations must also be able to represent single words in semantically meaningful ways. If they do not, then attempts to use the representations to compare between phrases and single words will fail.

A. NLP Benchmarks

As a first level of exploration, we studied the performance of our trained models on standard natural language benchmarks including the 2017 Semantic Textual Similarity (STS) Benchmark [3], the Stanford Natural Language Inference (SNLI) Corpus [2], and the Google Analogy Test Set [10].

The Semantic Textual Similarity Benchmark contains pairs of sentences (s_1,s_2) annotated with a human-generated similarity score between 0 and 5 for each pair. We evaluated our learned representations by using using cosign similarity $sim_{cos}(A,B) = \frac{AB}{\|A\|\|B\|}$ between embedded sentence pairs to calculate Pearson's r and Spearman's rho for each model.

The Stanford Natural Language Inference Corpus contains sentence pairs (s_1,s_2) along with annotator labels for the categories {'entailment','neutral','contradiction'}. To enable evaluation, we assigned numeric values of {0.0,0.5,1.0} to each of the categories respectively, and then calculated Pearson's r and Spearman's rho based on cosign similarity between embedded sentences. Dataset entries for which no category majority existed were ignored.

The Google Analogy Test Set contains word tuples of the form {A,B,C,D} representing the analogical reasoning query "A is to B as C is to ?" . Given the first three words in the series, the task of the algorithm is to calculate the unknown value D via vector offsets within the embedding space

$$D = argmax_{v \in V'} sim_{cos}(v, c + b - a) \tag{1}$$

where a, b and c are the embedded representations of A, B, and C, respectively and $V' = V - \{A, B, C\}$ is the vocabulary of the neural model excluding the source words A, B, and C.

Experimental results are shown in Figure 2. We have also included scores from state of the art embedding models, not because our objective in this experiment is to outperform them (our model is far too small and simple for that purpose) but to provide a comparison between our experimental results and the current highest achievements.

Analysis of the results reveals interesting patterns. For example, training based on context produces far better STS scores than training on either the reconstruction or word prediction tasks. This pattern holds across both input corpora, but note the curious behavior on the SNLI task: Context+Book Corpus strongly outperforms all of our other variants on that task, with scores exceeding even some of the state-of-the-art models. One is led to wonder: what is it about the intersection between context-based training and the structure of the Book Corpus that enable these dependencies to be learned, when all other design combinations fail to absorb that semantic information?

Note also that it is not the case that the context-based training produces the highest performance in all categories. Google Analogy results are highest when the Wikipedia Corpus is paired with reconstruction loss. We can therefore conclude that there is a complex interplay between input corpus and training task. Choosing the optimal design configuration is far more complicated than simply hailing any particular corpus or training task as the superior choice.

B. *Semantic Triplets*

We next evaluate our learned representations on the triplets dataset introduced zhu et al. [12]. The triplets dataset was designed to measure distinct semantic properties of sentence embeddings, with an emphasis on their respective geometries rather than on their usefulness for downstream tasks. Each evaluation is structured as set of sentence triplets, and a triplet is solved correctly if the cosine similarity between the embedded representations of the first two sentences is greater than the similarity between the first and third sentences. Subtasks fall into the following categories.

1) **Argument sensitivity**: Measures whether the representations encode a sentence and a passivized version of the sentence into similar regions of the embedding space.
2) **Fixed point re-ordering**: The order of words in the sentence is restructured in a way that garbles semantic meaning. The representations are expected to place the fixed point inversion farther from the original sentence than a semantically similar sentence extracted from the SICK dataset.
3) **Negation detection:** The word 'not' is inserted into a sentence at a grammatically appropriate location. The distance between the original sentence and a sentence with one word swapped for a synonym is expected to be less than the distance between the original sentence and the not-negation.
4) **Clause relatedness:** An embedded clause is extracted from within a sentence. The distance between the original sentence and the extracted clause is expected to be less than the distance between the original sentence and its not-negation.
5) **Negation variants:** The distance between a sentence's not-negation and its quantifier-negation should be less than the distance between the not-negation and the original sentence.

Results are shown in Figure 3. Here, too, interesting patterns rise to the forefront. The Wikipedia corpus again results in higher average performance than models trained using Book Corpus. The reconstruction task consistently produces the best results in Fixed Point Reordering, regardless of training corpus. But note the way performance on the Argument Sensitivity task is highly sensitive to the intersection between corpus and training task. When the Wikipedia corpus is used, the word prediction training task has the highest performance. When Book Corpus is used, the context task is most effective.

As in the previous subsection, we have included the performance of several state-of-the-art neural embedding models for comparison, but it is not our intent to pit our experimental models against this gold standard. Rather, we are interested in the ways that our selection of input corpus and training task affect the geometric properties of the learned sentence representations. Comparison with state-of-the-art models is useful primarily to place our performance numbers within the larger context of linguistic modeling.

V. Analysis and Recommendations for Future Work

Our results reveal that there is a complex interplay between training task and input corpus that powerfully affects the geometries of the model's learned sentence representations. If we had to choose a "winner", overall results would suggest that Wikipedia is the best input corpus and the context training task is generally the most effective curriculum choice, but the internal dynamics of neural language models are not that simple. It is the intersction of both design elements, and not either in isolation, that determines the final outcome.

Although not covered in the current experiments, we are convinced that this interplay also has a third component: network architecture. Thus, there is no such thing as a "best" input corpus or "most effective" training task. There is only

	STS r	STS rho	SNLI r	SNLI rho	Google Analogy
State of the art					
GPT-2	-0.052	0.092	-0.007	0.019	6.47%
InferSent	0.718	0.702	0.273	0.279	81.81%
Google use lite	0.751	0.737	0.366	0.367	52.12%
Skip-thought	0.214	0.296	0.046	0.108	50.86%
BERT	0.495	0.490	0.166	0.174	46.56%
FastText BoW	0.547	0.543	0.248	0.257	77.20%
Glove BoW	0.404	0.440	0.241	0.247	82.39%
Average	**0.440**	**0.471**	**0.190**	**0.207**	**56.77%**
RNN - Wikipedia Corpus					
reconstruction	0.393	0.421	0.075	0.081	55.37%
context	0.401	0.460	0.043	0.071	52.49%
word prediction	0.087	0.232	0.054	0.079	47.81%
Average	**0.294**	**0.371**	**0.057**	**0.077**	**51.89%**
RNN - Book Corpus					
reconstruction	0.212	0.289	0.007	0.014	42.16%
context	0.410	0.395	0.141	0.139	51.79%
word prediction	0.057	0.086	-0.012	-0.002	28.40%
Average	**0.226**	**0.257**	**0.045**	**0.050**	**40.78%**

Fig. 2. Model performance on the SemEval 2017 Semantic Textual Similarity Benchmark, the Stanford Natural Language Inference Corpus, and the Google Analogy Test Set. The first two datasets were evaluated using Pearson's r and Spearman's rho (higher is better), the Google dataset was evaluated based on response accuracy. The Wikipedia models have higher average scores than the Book Corpus models in every category, however the combination of the Book Corpus with the context training task performs better than its Wikipedia counterpart on the SNLI task.

	arg sen	fixed point	neg detect	clause	neg variants	average
State of the art						
GPT-2	25.11%	98.40%	61.19%	38.23%	64.97%	59.80%
InferSent	1.57%	70.35%	97.48%	48.50%	92.17%	65.54%
Google use lite	1.79%	75.80%	77.78%	2.48%	81.02%	50.83%
Skip-thought	4.48%	99.84%	61.48%	20.53%	18.40%	44.95%
BERT BoW	1.57%	95.85%	89.48%	6.02%	71.23%	56.97%
FastText BoW	0.45%	0.16%	37.93%	28.85%	2.54%	15.42%
Glove Bow	0.45%	0.0%	20.74%	22.48%	22.90%	13.68%
Average	**5.06%**	**62.91%**	**63.73%**	**23.87%**	**50.46%**	**43.88%**
RNN - Wikipedia Corpus						
reconstruction	11.21%	90.22%	13.93%	7.26%	24.46%	30.95%
context	4.71%	95.19%	16.15%	3.89%	14.48%	29.07%
word prediction	26.68%	84.45%	51.70%	33.81%	39.73%	49.24%
Average	**14.2%**	**89.95%**	**27.26%**	**14.99%**	**26.22%**	**36.42%**
RNN - Book Corpus						
reconstruction	6.5%	98.24%	17.33%	2.48%	24.07%	31.76%
context	24.89%	73.56%	24.25%	16.81%	13.11%	31.73%
word prediction	7.17%	82.85%	31.41%	11.33%	28.77%	34.36%
Average	**14.18%**	**84.88%**	**24.33%**	**10.21%**	**21.98**	**32.62%**

Fig. 3. Classification accuracy on the triplet task introduced by zhu et al, exploring five distinct semantic properties of the learned representations. On these tasks as well, the models trained on Wikipedia have higher average accuracies than the ones trained on Book Corpus, and even outperform the state-of-the-art averages in two categories. Note, however, the unusually high performance of reconstruction+Book Corpus on the the Fixed Point Reordering task. In some areas, it seems, the combination of input corpus and training curriculum can have unexpectedly dramatic results.

the question of which training tasks are most compatible with recurrent networks trained on Wikipedia, or with transformers trained on Book Corpus, and so forth.

Future work in this area should expand these empirical studies to include more network architectures. The effect of using more than one training task or input corpus at a time should also be examined. The prevailing current opinion when training deep neural representations seems to be "more data is always better", and models are often trained on large corpora aggregated from sources with many different kinds of text, and with training curricula that combine context prediction, natural language inference, masked word prediction, sentiment classification, and many other tasks. We question whether this is the most effective approach, and strongly suspect that the mingling of multiple corpora or training tasks inhibits the learning of effective representations by diluting and in some cases muddling the training signals. A key indicator in this regard is the performance of the state-of-the-art InferSent model, which was trained on only a single task, and which nevertheless outperforms all competing models in approximately half of the evaluation tasks used (see Figures 2 and 3).

We theorize that there may be fundamental issues of compatibility between specific pairs of training tasks, and that some harmonize well with each other while others do not. Extensive and carefully structured experimentation is necessary to determine which training tasks are compatible and which may be working at cross purposes. Similarly, further investigation is required in order to determine whether training corpora can be indiscriminately aggregated, or whether there is a cost in terms of final model performance when text corpora with different distributional properties are merged together during training.

VI. Conclusion

This work has investigated the impact of design decisions such as input corpus and training task on the final geometry of the learned sentence representations of a simple recurrent network. We discover that, while the Wikipedia corpus results in better performance on average, it is the complex interplay between the corpus and training task in combination that truly produces excellence. As a result of our studies, we encourage researchers to carefully consider these design choices, and to experiment with the various elements in combination rather than selecting corpus and training tasks independently of one another. In particular, we have observed that the combination of Book Corpus and a context-prediction task are particularly effective at inducing semantically aligned representations, as evidenced by the Semantic Textual Similarity Benchmark, and that the combination of a Wikipedia corpus and a sentence reconstruction task results in impressively high performance on a fixed point reordering task.

Future extensions of this work should focus on the responsiveness of various network architectures to the choice of input corpus and training task(s), with particular emphasis on the sensitivity of transformers and convolutional networks to these design decisions. We also encourage the principled examination of training tasks used in isolation versus in aggregate, and of input corpora with greater or lesser degrees of variance in syntactic structure, sentence length, and vocabulary.

References

[1] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[2] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

[3] Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity - multilingual and crosslingual focused evaluation. *Proceedings of SemEval 2017*, 2017.

[4] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal sentence encoder. *CoRR*, abs/1803.11175, 2018.

[5] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. *CoRR*, abs/1705.02364, 2017.

[6] Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *arXiv preprint arXiv:1805.01070*, 2018.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[8] Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daume III. Deep unordered composition rivals syntactic methods for text classification. In *Association for Computational Linguistics*, 2015.

[9] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-thought vectors. pages 3294–3302, 2015.

[10] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[11] Yikang Li Xupeng Tong and Chao-Ming Yen. Variational neural conversational model. *ICML*, 2014.

[12] Xunjie Zhu, Tingfeng Li, and Gerard De Melo. Exploring semantic properties of sentence embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 632–637, 2018.

[13] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.