Modeling Semantic Composition over Word Vectors

Liam Geron
Cuny Graduate Center / 365 5th Ave, New York, NY
liams.geron@gmail.com

Abstract

In recent years word vectors have become near ubiquitous in downstream NLP tasks such as Schizophrenia detection (Kuperberg, 2010) or sarcasm detection (Ghosh et al., 2015), where each task is heavily dependent on a good semantic representation. Currently, the standard for utilizing word vectors as a way of representing a sentence is simple averaging or addition. This paper proposes an alternative linguistically motivated method for composing word vectors into a fixed length sentence representation utilizing a Recursive Neural Network (Socher et al., 2013) and a training objective inspired by Mikolov (2013).

1 Background

Word vectors have proven themselves to be highly useful in many downstream NLP tasks due to their ability to capture a large amount of intuitive semantic information utilizing a distributional approach. Algorithms such as Word2Vec (Mikolov et al., 2013) latently factorize a word-context coocurrence matrix (Levy and Goldberg, 2014). This distributional information has been shown to be extremely powerful; it allows them to represent analogies (Mikolov et al., 2013), cluster around semantically similar words (Huang et al., 2012), and even contain some form of polysemy (Arora et al., 2016). Utilizing these word representations to create fixed length sentence representations has remained an open issue since their inception, though, as was seen in Mikolov (2013), simple vector arithmetic can allow for intuitively consistent semantic representations of larger phrases. However, there are at least two problems with such a simple solution. One is that averaging or summing ignores word order, and works as more of a bag-of-words approach, ignoring the hierarchical structure of language. Another is that averaging or summing is a linear combination, and ignores the possibility that semantic composition can be more accurately modeled via some non-linearity. Socher (2013) presented a model architecture that is capable of composing word vectors through a Neural Network while incorporating a parse tree to utilize syntactic information. This approach solved both of the above problems with semantic composition, yet focused entirely on sentiment analysis as a downstream task rather than building an accurate semantic representation of phrases or sentences. This paper proposes a modification of the architecture found in Socher (2013) in order to develop accurate and length agnostic distributionally motivated phrase/sentence representations through the addition of the objective function from Mikolov (2014).

2 Related Work

Much work has been done in the domain of sentence representations, from Skip-Thought vectors (Kiros et al., 2015), to utilizing a Convolutional Neural Network to represent a sentence (Kalchbrenner et al., 2014), to even attempts at capturing the logical form of a sentence (Zettlemoyer and Collins, 2012). The utility of a good semantic or topical representation is clear; sentiment analysis, sarcasm detection, and even something as high-level as Schizophrenia detection all benefit from an accurate semantic/topical representation.

In the domain of Schizophrenia detection, for example, Kuperberg (2010) discussed a symptom found with patients who have Schizophrenia that they refer to as 'derailment'. Patients with derailment often go off on topical tangents, and have less semantic consistency on the sentence-to-sentence level. In order to create a feature capable

of modeling this phenomenon, we must model the overall semantic meaning of a sentence or phrase.

The two models that this paper will be focusing on and utilizing are the models presented in Socher (2013) and Mikolov (2014). Both have very desirable properties that this paper intends to combine and improve upon through a combination of both, and, hopefully, present a new model capable of modeling a semantic composition function.

2.1 Recursive Neural Networks

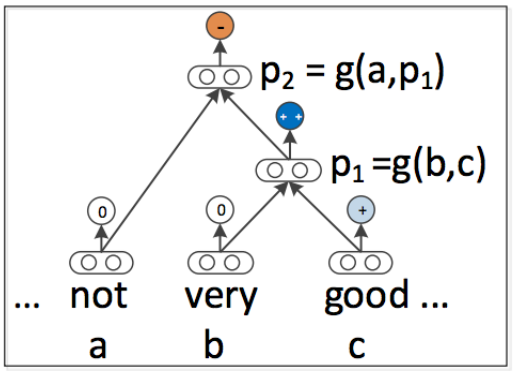

Figure 1: An example of a sentence run through an RNN from Socher et al., 2013

As seen in Figure 1, the general structure of the RNN is based off of a parse tree of the input sentence. From this parse tree, the order of compositions is found by traversing the tree and finding the mother node that has all of its daughters computed. That node is then computed via a Feed Forward Neural Network. An example of this function can be seen in the equation below which traverses the syntax tree found in Figure 1:

$$p_1 = f(W[b, c]), p_2 = f(W[a, p_1])$$

where $f = tanh$ is the element-wise hyperbolic tan nonlinearity, and $W \epsilon \mathbb{R}^{d \times 2d}$ is our main parameter to learn (i.e. our composition parameter)[1].

Socher et al. utilize an additional softmax classifier at each node that takes each hidden state and feeds it into a softmax layer in order to classify it's sentiment. Because they had labeled data corresponding to each parse tree, they were able to back-propagate the error to each node and ultimately to the weight matrix, W, itself. Because they are predicting sentiment, the use of a softmax classifier here is not prohibitively expensive as the number of classes was either 3 or 5 depending on the experiment (Socher et al., 2013).

2.2 Paragraph Vectors

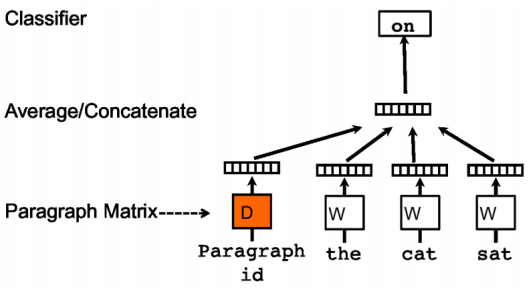

Figure 2: An example of the model presented in Mikolov (2014)

Mikolov (2014), calculated what they call "paragraph vectors" through a very similar approach to the near ubiquitous Word2Vec model for word representations (Mikolov et al., 2013). The general approach is to predict a context of words given a central word, or vice versa, via a Neural Network, and subsequently train the network based off of those predictions. This approach traditionally involves a softmax layer over the entirety of the vocabulary which can quickly become prohibitively expensive. To overcome this, Mikolov (2013), used a variant of Noise Contrastive Estimation[2] which they call Negative Sampling to train their network.

The main variant on this architecture for the expansion to sentence or paragraph representation is the addition of what they call the paragraph vector (See Fig. 2), which is concatenated or averaged along with the other context words in order to predict the central word or vice versa. For each paragraph, they iterate over every pair of (context, center) or (center, context) training points and keep the paragraph vector consistent at prediction time. Intuitively, these vectors form an external memory for the network to update with the "topic" of the paragraph.

These paragraph vectors are very good at downstream tasks, as they estimate their values distributionally. In fact, they do better than the sentence vectors found in Socher (2013) at sentiment analysis despite not being directly optimized for it. It

[1] All these figures and equations are found in Socher et al., 2013

[2] To be explained within the next section

is this quality that is desirable for our own sentence representations. Paragraph vectors have no compositional component to them, however, and ignore the hierarchical structure of language.

2.2.1 Noise Contrastive Estimation

Noise Contrastive Estimation (Gutmann and Hyvärinen, 2010) is an approximation of the softmax function for classification problems with a prohibitive number of classes, such as the models in Mikolov et al., 2013/2014. It approximates this by treating each softmax as an instance of a logistic regression instead, where the classifier is distinguishing between the actual correct class and noise drawn from a given distribution. In Mikolov (2013), this noise is sampled randomly from the training data, and a small variant is added in which low frequency words are used as the noise more often than high frequency words.

3 RNN Sentence Vectors

The RNN Sentence Vector model utilizes the same RNN structure as Socher (2013), including the same composition function. With this basic scaffolding we can change the model in order to get more generalized semantic representation in two possible ways.

3.1 Distributional Objective Function

The main difference in this architecture will be in the objective function of each node of the tree. Instead of classifying sentiment at each node however, each node will be used to predict the next word in the sentence utilizing NCE or Negative Sampling. This addition will prevent the prohibitive train time of softmax while still approximating the semantic advantage gained by it.

In this architecture, the final semantic representation will be the hidden state at the root note, or the composition of the two largest constituents. One advantage of this approach to that of Socher (2013) is that it does not require labeled data, i.e. can be trained in an unsupervised manner. The training dataset can be a sequence of parse trees, where each constituent of each parse tree is a train point and the next word in that sentence serves as that point's label. Errors can be summed from the NCE/NEG for the total batch error, and can be back-propagated through the network[3] via some standard optimization algorithm (i.e. Gradient Descent, Adam (Kingma and Ba, 2014), AdaGrad (Duchi et al., 2011), etc.).

An additional advantage of composing sentence vectors this way is that smaller constituents are also mapped to the same vector space as the larger ones. Sentences, phrases, and words will all share a continuous vector space in this model, more closely resembling human intuition where there is no clear semantic distinction between words and phrases/sentences.

3.2 External Memory

For the second experiment we will utilize the same architecture as the experiment above, however we will additionally include the main contribution of Mikolov (2014) (i.e. the addition of the paragraph vector to the prediction step) and apply it to our RNN. More concretely, for each node the prediction step will be calculated as follows:

$$p_1 = f(W[b, c]), \hat{\mathrm{y}} = softmax([p_1, s_i])$$

where $S \epsilon \mathbb{R}^{\mathrm{d} \times \mathrm{N}}$ is the sentence matrix where each column represents a specific sentence $s_i \epsilon \mathbb{R}^{\mathrm{d}}$ in the training set. At each node of the tree, the prediction will now be computed as a softmax over the hidden state h which is a concatenation of the hidden state of the node p_1 and the sentence vector s_i.

The addition of the sentence vector s_i will allow our network to take advantage of an external memory, much like Mikolov (2014). This sentence vector can be either additional to, or used instead of, the hidden state at the root node of the compositional tree. One potential advantage to this approach would be that it can avoid the bottlenecking of the dimensionality of the vectors by encoding each sentence into two d-dimensional vectors, or one $2 \times d$ dimensional vector.

4 Evaluation

One challenge that comes with a good semantic representation is, how do we measure what "good" actually means quantitatively? Mikolov (2013) had the option of using qualitative observation through the analogy task (Man : King :: Woman : Queen), but that task is much more difficult on the sentence and the phrase level. The way Socher (2013) approaches this is to measure qualitatively how the vectors do on a downstream task

[3] A more detailed explanation of this process can be found in Socher (2013)

that necessitates some "understanding" of the semantics. We propose a mixture of both qualitative and quantitative tasks to compare and understand our sentence vectors.

4.1 Quantitative

4.1.1 Sentiment Analysis

The clear choice for downstream tasks where we can have a direct comparison to Socher (2013) and Mikolov (2014) would be sentiment analysis. We can use the Stanford Sentiment Treebank introduced in Socher (2013) and the guidelines detailed in that text to have a dataset where each datapoint is a constituent with a label representing sentiment. From our trained network we can generate the constituent representation (whether that be the representation from section 3.1 or 3.2) which we then can feed into a logistic regression to classify the final sentiment.

This task successfully requires some knowledge of the sentiment, however it does not test our representations full knowledge as well as other downstream tasks.

4.1.2 Natural Language Inference

Natural Language Inference is the task of inferring if two sentences entail, contradict, or have a neutral relationship with each other. As detailed in Bowman (2015), NLI has an important role in evaluating sentence representations as it necessitates a much larger amount of semantic knowledge than Sentiment Analysis.

The Stanford NLI corpus introduced by Bowman (2015) should be used as a standardized dataset to compare the various algorithms against one another. In order to compare properly, the unsupervised RNN proposed in this paper should be compared against the supervised models used in Bowman (2015) and the Paragraph Vectors in Mikolov (2014). Particularly in the area of NLI the RNN's strengths should play a large role due to how compositional the task of NLI is inherently. Negation, for example, could play a huge role in sentence entailment, and RNNs have been shown to be sensitive to the effects of negation (Socher et al., 2013).

4.2 Qualitative

While not as rigorous of testing as quantitative, qualitative evidence should also be used to examine how well the sentences match the intuitions of a native speaker. Projecting the vectors down to 2 or 3 dimensional space and examining the neighbors to phrases provide a good form of direct observation of the spacial relationship between the vectors. In theory we should be able to observe semantically similar sentences cluster in similar locations.

The word analogy evaluation (as in Mikolov (2013)) can also provide qualitative evidence. Though it should not be considered as rigorous as the word vector analogy, sentence level analogy is possible, though the obfuscation of the compositional interactions of constituents can impair our ability to reason intuitively about them. For example, the sentence pair:

> "The capital of France"
>
> "The capital of Germany"

should be similar in their distance to the representation of the words "Paris" and "Berlin". Because we embed our sentences and phrases in the same semantic space as words, this analysis, too, should bear informative results.

5 Conclusion and Further Research

In this paper we have introduced an extension to the Recursive Neural Network introduced in Socher (2013). Using our methodology, we are able to achieve more generalized meaning-oriented vectors rather than the targeted task-specific vectors generated in Socher (2013). It is our hope that these vectors will capture a larger amount of semantic information than previous efforts at sentence representation so their use in a semantically driven task can dramatically improve results. One example of this usage would be to utilize these representations to detect topic shifts as are indicative of Schizophrenia (Kuperberg, 2010), which could be captured as average cosine distance between two sentences in the subject's speech.

Further research into this architecture should be conducted, and could include extending the compositional function to have hidden layers, changing the architecture of the compositional function to an alternative form of Neural Network such as the LSTM-RNN (Hochreiter and Schmidhuber, 1997), or utilizing other variants of the RNN such as those discussed in Irsoy (2014).

References

Arora, S., Li, Y., Liang, Y., Ma, T., and Risteski, A. (2016). Linear algebraic structure of word senses, with applications to polysemy. *arXiv preprint arXiv:1601.03764*.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Ghosh, D., Guo, W., and Muresan, S. (2015). Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words. In *EMNLP*, pages 1003–1012.

Gutmann, M. and Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, volume 1, page 6.

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

Huang, E. H., Socher, R., Manning, C. D., and Ng, A. Y. (2012). Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics.

Irsoy, O. and Cardie, C. (2014). Deep recursive neural networks for compositionality in language. In *Advances in Neural Information Processing Systems*, pages 2096–2104.

Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. *CoRR*, abs/1404.2188.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R. S., Torralba, A., Urtasun, R., and Fidler, S. (2015). Skip-thought vectors. *CoRR*, abs/1506.06726.

Kuperberg, G. R. (2010). Language in schizophrenia part 1: an introduction. *Language and linguistics compass*, 4(8):576–589.

Le, Q. V. and Mikolov, T. (2014). Distributed representations of sentences and documents. *CoRR*, abs/1405.4053.

Levy, O. and Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 2177–2185. Curran Associates, Inc.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. P. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*.

Zettlemoyer, L. S. and Collins, M. (2012). Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. *CoRR*, abs/1207.1420.