Exploring Semantic Properties of Sentence Embeddings

Xunjie Zhu
Rutgers University
Piscataway, NJ, USA
xunjie.zhu@rutgers.edu

Tingfeng Li
Northwestern Polytechnical University, Xi'an, China
ltf@mail.nwpu.edu.cn

Gerard de Melo
Rutgers University
Piscataway, NJ, USA
gdm@demelo.org

Abstract

Neural vector representations are ubiquitous throughout all subfields of NLP. While word vectors have been studied in much detail, thus far only little light has been shed on the properties of sentence embeddings. In this paper, we assess to what extent prominent sentence embedding methods exhibit select semantic properties. We propose a framework that generate triplets of sentences to explore how changes in the syntactic structure or semantics of a given sentence affect the similarities obtained between their sentence embeddings.

1 Introduction

Neural vector representations have become ubiquitous in all subfields of natural language processing. For the case of word vectors, important properties of the representations have been studied, including their linear substructures (Mikolov et al., 2013; Levy and Goldberg, 2014), the linear superposition of word senses (Arora et al., 2016b), and the nexus to pointwise mutual information scores between co-occurring words (Arora et al., 2016a).

However, thus far, only little is known about the properties of sentence embeddings. Sentence embedding methods attempt to encode a variable-length input sentence into a fixed length vector. A number of such sentence embedding methods have been proposed in recent years (Le and Mikolov, 2014; Kiros et al., 2015; Wieting et al., 2015; Conneau et al., 2017; Arora et al., 2017).

Sentence embeddings have mainly been evaluated in terms of how well their cosine similarities mirror human judgments of semantic relatedness, typically with respect to the SemEval Semantic Textual Similarity competitions. The SICK dataset (Marelli et al., 2014) was created to better benchmark the effectiveness of different models across a broad range of challenging lexical, syntactic, and semantic phenomena, in terms of both similarities and the ability to be predictive of entailment. However, even on SICK, oftentimes very shallow methods prove effective at obtaining fairly competitive results (Wieting et al., 2015). Adi et al. investigated to what extent different embedding methods are predictive of i) the occurrence of words in the original sentence, ii) the order of words in the original sentence, and iii) the length of the original sentence (Adi et al., 2016, 2017). Belinkov et al. (2017) inspected neural machine translation systems with regard to their ability to acquire morphology, while Shi et al. (2016) investigated to what extent they learn source side syntax. Wang et al. (2016) argue that the latent representations of advanced neural reading comprehension architectures encode information about predication. Finally, sentence embeddings have also often been investigated in classification tasks such as sentiment polarity or question type classification (Kiros et al., 2015). Concurrently with our research, Conneau et al. (2018) investigated to what extent one can learn to classify specific syntactic and semantic properties of sentences using large amounts of training data (100,000 instances) for each property.

Overall, still, remarkably little is known about what specific semantic properties are directly reflected by such embeddings. In this paper, we specifically focus on a few select aspects of sentence semantics and inspect to what extent prominent sentence embedding methods are able to capture them. Our framework generates triplets of sentences to explore how changes in the syntactic structure or semantics of a given sentence affect the similarities obtained between their sentence embeddings.

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 1–6
Melbourne, Australia, July 15 - 20, 2018.

2 Analysis

To conduct our analysis, we proceed by generating new phenomena-specific evaluation datasets.

Our starting point is that even minor alterations of a sentence may lead to notable shifts in meaning. For instance, a sentence S such as *A rabbit is jumping over the fence* and a sentence S^* such as *A rabbit is not jumping over the fence* diverge with respect to many of the inferences that they warrant. Even if sentence S^* is somewhat less idiomatic than alternative wordings such as *There are no rabbits jumping over the fence*, we nevertheless expect sentence embedding methods to interpret both correctly, just as humans do.

Despite the semantic differences between the two sentences due to the negation, we still expect the cosine similarity between their respective embeddings to be fairly high, in light of their semantic relatedness in touching on similar themes.

Hence, only comparing the similarity between sentence pairs of this sort does not easily lend itself to insightful automated analyses. Instead, we draw on another key idea. It is common for two sentences to be semantically close despite differences in their specific linguistic realizations. Building on the previous example, we can construct a further contrasting sentence S^+ such as *A rabbit is hopping over the fence*. This sentence is very close in meaning to sentence S, despite minor differences in the choice of words. In this case, we would want for the semantic relatedness between sentences S and S^+ to be assessed as higher than between sentence S and sentence S^*.

We refer to this sort of scheme as sentence triplets. We rely on simple transformations to generate several different sets of sentence triplets.

2.1 Sentence Modification Schemes

In the following, we first describe the kinds of transformations we apply to generate altered sentences. Subsequently, in Section 2.2, we shall consider how to assemble such sentences into sentence triplets of various kinds so as to assess different semantic properties of sentence embeddings.

Not-Negation. We negate the original sentence by inserting the negation marker *not* before the first verb of the original sentence A to generate a new sentence B, including contractions as appropriate, or removing negations when they are already present, as in:

A: The young boy is climbing the wall made of rock.

B: The young boy isn't climbing the wall made of rock.

Quantifier-Negation. We prepend the quantifier expression *there is no* to original sentences beginning with *A* to generate new sentences.

A: A girl is cutting butter into two pieces.

B: There is no girl cutting butter into two pieces.

Synonym Substitution. We substitute the verb in the original sentence with an appropriate synonym to generate a new sentence B.

A: The man is talking on the telephone.

B: The man is chatting on the telephone.

Embedded Clause Extraction. For those sentences containing verbs such as *say*, *think* with embedded clauses, we extract the clauses as the new sentence.

A: Octel said the purchase was expected.

B: The purchase was expected.

Passivization. Sentences that are expressed in active voice are changed to passive voice.

A: Harley asked Abigail to bake some muffins.

B: Abigail is asked to bake some muffins.

Argument Reordering. For sentences matching the structure "⟨somebody⟩ ⟨verb⟩ ⟨somebody⟩ *to* ⟨do something⟩", we swap the subject and object of the original sentence A to generate a new sentence B.

A: Matilda encouraged Sophia to compete in a match.

B: Sophia encouraged Matilda to compete in a match.

Fixed Point Inversion. We select a word in the sentence as the pivot and invert the order of words before and after the pivot. The intuition here is that this simple corruption is likely to result in a new sentence that does not properly convey the original meaning, despite sharing the original words in common with it. Hence, these sorts of corruptions can serve as a useful diagnostic.

A: *A dog is running on concrete and is holding a blue ball*

B: *concrete and is holding a blue ball a dog is running on.*

2.2 Sentence Triplet Generation

Given the above forms of modified sentences, we induce five evaluation datasets, consisting of triplets of sentences as follows.

1. **Negation Detection:** Original sentence, Synonym Substitution, Not-Negation

   With this dataset, we seek to explore how well sentence embeddings can distinguish sentences with similar structure and opposite meaning, while using Synonym Substitution as the contrast set. We would want the similarity between the original sentence and the negated sentence to be lower than that between the original sentence and its synonym version.

2. **Negation Variants:** Quantifier-Negation, Not-Negation, Original sentence

   In the second dataset, we aim to investigate how well the sentence embeddings reflect negation quantifiers. We posit that the similarity between the Quantifier-Negation and Not-Negation versions should be a bit higher than between either the Not-Negation or the Quantifier-Negation and original sentences.

3. **Clause Relatedness:** Original sentence, Embedded Clause Extraction, Not-Negation

   In this third set, we want to explore whether the similarity between a sentence and its embedded clause is higher than between a sentence and its negation.

4. **Argument Sensitivity:** Original sentence, Passivization, Argument Reordering

   With this last test, we wish to ascertain whether the sentence embeddings succeed in distinguishing semantic information from structural information. Consider, for instance, the following triplet.

   S: Lilly loves Imogen.
   S^+: Imogen is loved by Lilly.
   S^*: Imogen loves Lilly.

   Here, S and S^+ mostly share the same meaning, whereas S^+ and S^* have a similar word order, but do not possess the same specific meaning. If the sentence embeddings focus more on semantic cues, then the similarity between S and S^+ ought to be larger than that between S^+ and S^*. If the sentence embedding however is easily misled by matching sentence structures, the opposite will be the case.

5. **Fixed Point Reorder:** *Original sentence*, *Semantically equivalent sentence*, *Fixed Point Inversion*

   With this dataset, our objective is to explore how well the sentence embeddings account for shifts in meaning due to the word order in a sentence. We select sentence pairs from the SICK dataset according to their semantic relatedness score and entailment labeling. Sentence pairs with a high relatedness score and the *Entailment* tag are considered semantically similar sentences. We rely on the Levenshtein Distance as a filter to ensure a structural similarity between the two sentences, i.e., sentence pairs whose Levenshtein Distance is sufficiently high are regarded as eligible.

   Additionally, we use the Fixed Point Inversion technique to generate a contrastive sentence. The resulting sentence likely no longer adequately reflects the original meaning. Hence, we expect that, on average, the similarity between the original sentence and the semantically similar sentence should be higher than that between the original sentence and the contrastive version.

3 Experiments

We now proceed to describe our experimental evaluation based on this paradigm.

3.1 Datasets

Using the aforementioned triplet generation methods, we create the evaluation datasets listed in Table 1, drawing on source sentences from SICK, Penn Treebank WSJ and MSR Paraphase corpus. Although the process to modify the sentences is automatic, we rely on human annotators to double-check the results for grammaticality and semantics. This is particularly important for synonym substitution, for which we relied on WordNet (Fellbaum, 1998). Unfortunately, not all synonyms are suitable as replacements in a given context.

Table 1: Generated Evaluation Datasets

Dataset	# of Sentences
Negation Detection	674
Negation Variants	516
Clause Relatedness	567
Argument Sensitivity	445
Fixed Point Reorder	623

3.2 Embedding Methods

In our experiments, we compare three particularly prominent sentence embedding methods:

1. GloVe Averaging (GloVe Avg.): The simple approach of taking the average of the word vectors for all words in a sentence. Although this method neglects the order of words entirely, it can fare reasonably well on some of the most commonly invoked forms of evaluation (Wieting et al., 2015; Arora et al., 2017). Note that we here rely on regular unweighted GloVe vectors (Pennington et al., 2014) instead of fine-tuned or weighted word vectors.

2. Concatenated P-Mean Embeddings (P-Means): Rücklé et al. (2018) proposed concatenating different p-means of multiple kinds of word vectors.

3. Sent2Vec: Pagliardini et al. (2018) proposed a method to learn word and n-gram embeddings such that the average of all words and n-grams in a sentence can serve as a high-quality sentence vector.

4. The Skip-Thought Vector approach (SkipThought) by Kiros et al. (2015) applies the neighbour prediction intuitions of the word2vec Skip-Gram model at the level of entire sentences, as encoded and decoded via recurrent neural networks. The method trains an encoder to process an input sentence such that the resulting latent representation is optimized for predicting neighbouring sentences via the decoder.

5. InferSent (Conneau et al., 2017) is based on supervision from an auxiliary task, namely the Stanford NLI dataset.

3.3 Results and Discussion

Negation Detection. Table 2 lists the results for the Negation Detection dataset, where S, S^+, S^* refer to the original, Synonym Substitution, and Not-Negation versions of the sentences, respectively. For each of the considered embedding methods, we first report the average cosine similarity scores between all relevant sorts of pairings of two sentences, i.e. between the original and the Synonym-Substitution sentences (S and S^+), between original and Not-Negated (S and S^*), and between Not-Negated and Synonym-Substitution (S^+ and S^*). Finally, in the last column, we report the Accuracy, computed as the percentage of sentence triplets for which the proximity relationships were as desired, i.e., the cosine similarity between the original and synonym-substituted versions was higher than the similarity between that same original and its Not-Negation version.

On this dataset, we observe that GloVe Avg. is more often than not misled by the introduction of synonyms, although the corresponding word vector typically has a high cosine similarity with the original word's embedding. In contrast, both InferSent and SkipThought succeed in distinguishing unnegated sentences from negated ones.

Table 2: Evaluation of Negation Detection

	$S \wedge S^+$	$S \wedge S^*$	$S^+ \wedge S^*$	Accuracy
Glove Avg	97.42%	98.80%	96.53%	13.06%
P Means	98.49%	99.47%	98.13%	6.82%
Sent2Vec	91.28%	93.50%	85.30%	41.99%
SkipThought	88.34%	81.95%	73.74%	78.19%
Infersent	94.74%	88.64%	85.15%	91.10%

Negation Variants. In Table 3, S, S^+, S^* refer to the original, Not-Negation, and Quantifier-Negation versions of a sentence, respectively. Accuracy in this problem is defined as percentage of sentence triples whose similarity between $S+$ and S^* is the higher than similarity between S and $S+$ and S^+ and S^* The results of both averaging of word embeddings. and SkipThought are dismal in terms of the accuracy. InferSent, in contrast, appears to have acquired a better understanding of negation quantifiers, as these are commonplace in many NLI datasets.

Clause Relatedness. In Table 4, S, S^+, S^* refer to original, Embedded Clause Extraction, and Not-Negation, respectively. Although not particularly more accurate than random guessing, among the considered approaches, Sent2vec fares best in distinguishing the embedded clause of a sentence

Table 3: Evaluation of Negation Variants

	$S \wedge S^+$	$S \wedge S^*$	$S^+ \wedge S^*$	Accuracy
Glove Avg	96.91%	97.99%	97.05%	1.56%
P Means	98.66%	99.07%	98.49%	0.19%
Sent2Vec	90.53%	90.59%	86.87%	1.56%
SkipThought	71.94%	75.40%	73.11%	22.96%
InferSent	84.51%	88.45%	91.63%	85.78%

from a negation of said sentence.

For a detailed analysis, we can divide the sentence triplets in this dataset into two categories as exemplified by the following examples:

a) Copperweld said it doesn't expect a protracted strike. — Copperweld said it expected a protracted strike. — It doesn't expect a protracted strike.

b) "We made our own decision," he said. — "We didn't make our own decision," he said. — We made our own decision.

For cases resembling a), the average SkipThought similarity between the sentence and its Not-Negation version is 79.90%, while for cases resembling b), it is 26.71%. The accuracy of SkipThought on cases resembling a is 36.90%, and the accuracy of SkipThought on cases like b is only 0.75% It seems plausible that SkipThought is more sensitive to the word order due to the recurrent architecture. Infersent also achieved better performance on sentences resembling a) compared with sentences resembling b), its accuracy on these two structures is 28.37% and 15.73% respectively.

Table 4: Evaluation of Clause Relatedness

	$S \wedge S^+$	$S \wedge S^*$	$S^+ \wedge S^*$	Accuracy
Glove Avg	94.76%	99.14%	94.03%	4.58%
P Means	97.40%	99.61%	97.08%	2.46%
Sent2Vec	86.62%	92.40%	79.23%	32.92%
SkipThought	54.94%	84.27%	45.48%	19.51%
Infersent	89.47%	95.12%	85.22%	18.45%

Argument Sensitivity. In Table 5, S, S^+, S^* to refer to the original sentence, it Passivization form, and the Argument Reordering version, respectively. Although recurrent architectures are able to consider the order of words, unfortunately, none of the analysed approaches prove adept at distinguishing the semantic information from structural information in this case.

Fixed Point Reorder. In Table 6, S, S^+, S^* to refer to the original sentence, its semantically equivalent one and Fixed Point Inversion Version. As Table 6 indicates, sentence embeddings based on means (GloVe averages), weighted means (Sent2Vec), or concatenation of p-mean embeddings (P-Means) are unable to distinguish the fixed point inverted sentence from the semantically equivalent one, as they do not encode sufficient word order information into the sentence embeddings. Sent2Vec does consider n-grams but these do not affect the results sufficiently.SkipThought and InferSent did well when the original sentence and its semantically equivalence share similar structure.

Table 5: Evaluation of Argument Sensitivity

	$S \wedge S^+$	$S \wedge S^*$	$S^+ \wedge S^*$	Accuracy
Glove Avg	96.17%	99.96%	96.17%	0.00%
P Means	97.94%	99.98%	97.94%	0.00%
Sent2Vec	89.11%	99.80%	89.13%	0.00%
SkipThought	83.44%	95.57%	82.32%	4.71%
Infersent	93.70%	97.98%	94.11%	2.24%

Table 6: Evaluation of Fixed Point Reorder

	$S \wedge S^+$	$S \wedge S^*$	$S^+ \wedge S^*$	Accuracy
Glove avg	97.74%	100.00%	97.74%	0.00%
P-Means	98.68%	100.00%	98.68%	0.00%
Sent2Vec	92.88%	100.00%	92.88%	0.00%
SkipThought	89.83%	39.75%	37.28%	99.84%
InferSent	95.53%	94.26%	90.64%	72.92%

4 Conclusion

This paper proposes a simple method to inspect sentence embeddings with respect to their semantic properties, analysing three popular embedding methods. We find that both SkipThought and InferSent distinguish negation of a sentence from synonymy. InferSent fares better at identifying semantic equivalence regardless of the order of words and copes better with quantifiers. SkipThoughts is more suitable for tasks in which the semantics of the sentence corresponds to its structure, but it often fails to identify sentences with different word order yet similar meaning. In almost all cases, dedicated sentence embeddings from hidden states a neural network outperform a simple averaging of word embeddings.

Acknowledgments

This research is funded in part by ARO grant no. W911NF-17-C-0098 as part of the DARPA SocialSim program.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained Analysis of Sentence Embeddings Using Auxiliary Prediction Tasks. *arXiv.org*.

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Analysis of sentence embedding models using prediction tasks in natural language processing. *IBM Journal of Research and Development*, 61(4):3:1–3:9.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016a. A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016b. Linear algebraic structure of word senses, with applications to polysemy. *arXiv preprint arXiv:1601.03764*.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *Proceedings of ICLR 2017*.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James R. Glass. 2017. What do neural machine translation models learn about morphology? *CoRR*, abs/1704.03471.

A. Conneau, G. Kruszewski, G. Lample, L. Barrault, and M. Baroni. 2018. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *ArXiv e-prints*.

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

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. In *Proceedings of NIPS 2015*. MIT Press.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of ICML 2014*. PMLR.

Omer Levy and Yoav Goldberg. 2014. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 171–180, Ann Arbor, Michigan. Association for Computational Linguistics.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella bernardi, and Roberto Zamparelli. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. European Language Resources Association (ELRA).

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised learning of sentence embeddings using compositional n-Gram features. In *NAACL 2018 - Conference of the North American Chapter of the Association for Computational Linguistics*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1532–1543.

Andreas Rücklé, Steffen Eger, Maxime Peyrard, and Iryna Gurevych. 2018. Concatenated p-mean embeddings as universal cross-lingual sentence representations. *arXiv*.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural mt learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534, Austin, Texas. Association for Computational Linguistics.

Hai Wang, Takeshi Onishi, Kevin Gimpel, and David A. McAllester. 2016. Emergent logical structure in vector representations of neural readers. *CoRR*, abs/1611.07954.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. Towards universal paraphrastic sentence embeddings. *CoRR*, abs/1511.08198.