Evaluating Compositionality of Sentence Representation Models

Hanoz Bhathena[1], **Angelica Willis**, **Nathan Dass**
Stanford University
hanozbhathena@gmail.com, arwillis@stanford.edu,
ndass@stanford.edu

Abstract

We evaluate the compositionality of general-purpose sentence encoders by proposing two metrics to quantify compositional understanding capability of sentence encoders. We introduce a novel metric, Polarity Sensitivity Scoring (PSS), which utilizes sentiment perturbations as a proxy for measuring compositionality. We then compare results from PSS with those obtained via our proposed extension of a metric called Tree Reconstruction Error (TRE) (Andreas, 2019) where compositionality is evaluated by measuring how well a true representation-producing model can be approximated by a model that explicitly combines representations of its primitives.

1 Introduction

Compositionality is the principle inherent in human language whereby the meaning of a complex, compound language expression can be deduced from the meanings of its constituent parts and how they are combined. Compositionality can be thought of as a key ingredient towards making artificial intelligence more like general human intelligence since it enables understanding of highly complex concepts by breaking them down into simpler, more manageable, and modular components. The last couple of years have seen a breathtaking expansion in the research around transfer learning for natural language understanding. BERT (Devlin et al., 2019) has proven to be a highly successful model for learning general, task-agnostic sentence representations that can equal or outperform task-specific ones. Given the strong intuitive connection between compositionality and generalization of representation learning, but the relative difficulty in often quantifying it, our goal is to propose evaluation metrics for compositional understanding of sentence encoders and evaluate the level of compositional understanding in the current state-of-the-art sentence encoder models.

We propose two new methods to evaluate the compositionality of sentence embedding models. First, we propose a new method called Polarity Sensitivity Scoring (PSS) which measures compositionality via the ability of sentence encoding models to be sensitive to minor perturbations in the input that would flip the sentiment polarity of a sentence. Next, we extend Tree Reconstruction Error (TRE) (Andreas, 2019) to work sentences.

2 Related Work

With the rapid improvement of natural language understanding models in recent years, there has simultaneously been a large increase in research on the nuances and pitfalls of these models, especially in the area of compositionality. Among other methods, measuring performance in classification tasks targeting semantic understanding (Ettinger et al., 2016), lexical composition (Shwartz and Dagan, 2019), synonym substitution (Hupkes et al., 2020), and divergence (Keysers et al., 2020) have all been proposed.

Many researchers have shown evidence that inducing compositionality into deep and shallow models have helped in generalization, data efficiency, and interpretability. Fyshe et al. (2015) evaluates compositionality at the phrase level to make representations more interpretable. Baroni (2020) finds that neural networks are capable of subtle grammar-dependent generalizations, but do not rely on systematic compositional rules. Dessì and Baroni (2019) found that, perhaps counter-intuitively, CNNs were able to significantly outperform LSTMs and GRUs on the more difficult *jump* and *around-right* tasks in the SCAN challenge proposed by Lake and Baroni (2017) and

[1] Stanford SCPD student

Proceedings of the 5th Workshop on Representation Learning for NLP (RepL4NLP-2020), pages 185–193

Loula et al. (2018). However, they still find that CNNs also are not good at learning rule-like compositional generalizations as the mistakes it makes are not systematic and they are evenly spread across different commands.

Stone et al. (2017) explores the compositional properties of deep CNNs for image recognition. Their method quantifies compositionality as the difference in higher layer CNN activations between a network which takes a normal multi-object image as input and masks all activations outside the spatial location of one of the objects and a network which takes as input the same image as above with all other objects except the target object zeroed out. The intuition is that if CNNs are inherently compositional, then the difference in two activations should be zero.

3 Polarity Sensitivity Scoring (PSS)

Our primary contribution is a method we propose is called Polarity Sensitivity Scoring. Here, we posit that a model that has strong compositional understanding can adapt to small changes in the constituent components of a sentence such as sentiment polarity. Generally, the sentiment of a sentence is localized to a small fraction of the words, which can be separated from the overall content of the sentence that is not sentiment bearing. We hypothesize that if a model can accurately detect a sentiment switch when its thematic content remains constant, but only its tonality changes, then it should have a good semantic understanding of the nuances of composition structure. We define the equation for PSS as:

$$PSS = \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}[\hat{y_s} = y_s \wedge \hat{z_{s'}} = z_{s'}]$$

where y_s is the ground truth label for the sentence s and $\hat{y_s}$ is the predicted label produced by a sentiment model trained using the sentence encoding model. Similarly, $z_{s'}$ and $\hat{z_{s'}}$ are the ground truth and predicted labels, respectively, for the sentence s' for which the polarity has been flipped. For PSS, we need sentence pairs which have the same content but differ only in certain sentiment specific attributes. Ideally, we would want human-generated pairs but since that can be cumbersome and expensive we utilize outputs of an off the shelf model for synthetic data generation which generates the sentiment switched sentence. The approach we use was proposed by Li et al. (2018) and the interested reader is encouraged to read the paper to gain a better understanding of the algorithm details. This formula for PSS calculation would be sufficient if the sentiment switching model was perfect, however, this is not the case. To account for this we manually reviewed a subset of examples to come up with a set of rules which removed error-prone switches making our synthetic pairs closer to a gold standard. Details are described in appendix B.

The perturbation-driven nature of PSS might lead one to question whether PSS really captures compositional understanding or is it just a test of the robustness of sentence representations to noise. We believe that, at least with respect to the required compositional understanding to correctly classify sentiment (Socher et al., 2013), it does and might also be more general than that. PSS can actually complement the *consistency score* proposed by Hupkes et al. (2020) which measures substitutivity, one of the five tests for compositionality. While they replace words with their synonyms and expect the same classification, we replace sentiment bearing words and expect the model to accurately reflect this change in sentiment. Since changing a classification label establishes a more direct causal link between change in text and change in label, we believe that our method is better at least for the substitutivity test.

3.1 Experimental Results

We leverage the same dataset used in Li et al. (2018) for our experiments: a sentiment corpus of Yelp Business Reviews. The dataset contains 270K positive examples and 180K negative ones in the train set and an equally balanced 4000-example development set and 1000-example test set. Since our end goal is evaluating compositionality and not developing the most performant sentiment model, we use static hyperparameter configurations (learning rate=2e-5 for BERT and 3e-3 for others) and report test accuracy by combining dev and test sets. For the sentiment switched pairs we directly utilize the 500 reference test pairs released [1]. After the cleaning rules (see appendix) to remove problematic pairs we are left with 353 example pairs for which we have a high degree of confidence that they belong to opposite sentiments. Our PSS metric is therefore calculated on these 353 sentence pairs. If our models predict the correct label for both the

[1] https://github.com/rpryzant/delete_retrieve_generate

Encocder Type	Final Layers	Finetune mode	Test Accuracy	PSS	Relative
BERT	Linear	FB	79.2	59.5	75.1
BERT	Linear	FT	81.3	73.7	90.7
ELMo	Linear	FB	75.3	57.2	76.0
ELMo	Linear	FT	75.8	67.9	89.6
ELMo	DNN	FB	77	57.5	74.7
ELMo	DNN	FT	77.2	70	90.7
USE DAN	DNN	FB	75.3	62.3	82.8
USE DAN	DNN	FT	79.6	69.4	87.2
USE DAN	Linear	FB	69.3	49.9	71.9
USE DAN	Linear	FT	78.1	69.1	88.5
USE Transformer	Linear	FB	76.5	65.7	85.9
USE Transformer	Linear	FT	82.3	73.7	89.5
USE Transformer	DNN	FB	78.7	65.7	83.5
USE Transformer	DNN	FT	80.2	69.4	86.5

Table 1: Results from Polarity Sensitivity Scoring (PSS). Linear: Linear projection from sentence embedding to labels. DNN: 2 layer deep neural network. Relative: PSS / Test Accuracy$\times$100. Finetune mode: FT: encoder finetuned, FB: encoder parameters frozen with final layers only trained

positive and negative versions of the sentence, an example gets a score of 1 else 0 and these values are averaged to get the PSS score for a model.

We compare four encoder types: BERT (Devlin et al., 2019), ELMo (Peters et al., 2018), Universal Sentence Encoder (USE) deep averaging network (DAN), and USE Transformer (Cer et al., 2018). For all encoders except BERT, we experiment with different final layer types to isolate impact from the classification layer to the encoding layers: single linear layer (Linear) or 2 layer feedforward deep neural network (DNN) with 500 and 100 units in the first and second layers, respectively. For all encoders, we experiment with the finetuning mode (FT): train the all encoder and classification layers and feature-based mode (FB): freeze the encoder layers and only train the classification layers.

Table 1 shows the results of our polarity sensitivity experiment. Since each encoder has a different sentiment classification performance, we also consider the relative PSS, defined as $(PSS/TestAcc.) \times 100$, which helps us normalize the compositional understanding capability against its task performance.

We observe that BERT FT and USE Transformer FT, both Transformer Vaswani et al. (2017) architectures, are the leading models on absolute PSS and BERT FT and ELMo in DNN FT configuration are joint leaders on relative PSS. The fact that BERT leads in both categories is no surprise given its well known superior performance on wide-ranging tasks. On absolute PSS alone, ELMo and USE DAN are the least compositional. Given USE DAN's bag of words type architecture this makes sense but is slightly surprising for the ELMo LSTM architecture even though ELMo does better when we normalize by the sentiment classification accuracy. We note that the consistency score of Hupkes et al. (2020) shows quite similar results where the Transformer architecture outperforms both LSTM and CNN architectures substantially. Across the board, models that finetune the sentence encoder decidedly do better in absolute and relative terms than not finetuning which makes sense as encoders should generally be better equipped to pick up compositional generalizations than the classification layers which are the only trainable layers in FB mode. Additionally, comparison of the performance of DNN vs linear classifier types is less conclusive further suggesting that the difference in compositional understanding is most dependent on the sentence encoder versus the classifier chosen.

Since BERT FT is joint best with USE Transformer FT Linear on absolute PSS and they also are both among the top-performing models on test accuracy it validates our key motivation that good compositional understanding contributes towards good downstream performance. However, if we look at when we do not finetune BERT and USE Transformer encoders, we see that even though BERT FB has decent test accuracy, the PSS of BERT FB is 59.5% compared to 65.7% for USE

Transformer FB (linear and DNN). On the surface one would expect that true measurement of raw compositional understanding of a representation must be calculated without encoder fine-tuning however, we must remember that the pre-training mechanism of BERT and USE is quite different. While BERT is completely pre-trained using unsupervised Masked LM and next sentence prediction, USE is also trained using the supervised SNLI dataset (Bowman et al., 2015) which the authors note improves the transfer learning capability of USE. Given that natural language inference is a task that would be very hard to do well without some compositional understanding, it stands to reason that the pre-training phase of USE provides some implicit compositional advantages. This is equally valid for other types of models and so for accurate comparisons across models, we must default to FT mode.

Given that the above results correlate well with our *a priori* expectations based on both theoretical and empirical knowledge about these encoders, we feel confident that absolute and relative PSS can be good estimates of compositional understanding of sentence representation models.

4 Tree Reconstruction Error (TRE)

TRE (Andreas, 2019) measures the vector space distance between a target vector representation produced by an encoding model and a vector representation that is generated from compositions of its primitive units. In the case of sentences, the target representation is produced by a sentence encoding model and the primitives are generally the words in the sentence. The compositions are represented by syntactic parses of the sentences where at every subtree, the representations of the child nodes are composed using some composition function. The primitive representations (word vectors) are trained using RMSProp, fixing the sentence representation and compositional functions, to minimize the cosine distance between the sentence encoding and the output of the compositional function applied to primitives.

We aim to extend TRE [2] from phrases to sentences. Unfortunately, there are not many open source datasets with human-labeled compositionality scores for sentences that we could find. Therefore, using the Stanford Sentiment Treebank (SST) we propose two automated methods to generate ground truth compositionality labels for the SST dataset by using phrase-level sentiment labels in SST.

4.1 Tree Impurity

We start by traversing the constituency parsed tree of each SST sentence and collect the labels of all sub-components and phrases within the parse tree. To compute the Tree Impurity, we take the absolute difference between the root label and the average of all phrase labels within a tree. To understand why this metric is meaningful, let's consider the following example sentence from SST:

"A coda in every sense, The Pinochet Case splits time between a minute-by-minute account of the British court's extradition chess game and the regime's talking-head survivors."

As seen in Figure 3 (appendix), the phrase labels of the two children of the root and all of their children have a label of 2 (neutral). However, at the top, the root level label is 4 (highly positive). This constitutes an example of a sentence with a high degree of compositionality i.e. the overall meaning of a sentence is not just the meaning of the components but also how they are composed.

4.2 Weighted Node Switching (WNS)

Tree Impurity loses crucial information regarding the compositionality within subtrees. Weighted node switching seeks to counteract this by introducing more local compositional information. Here, for every subtree where both children have a sentiment label, we calculate the absolute difference between the sentiment label of the root of the subtree and the average sentiment labels of its children. To introduce global information, we weight this label difference by the height of the root node of the subtree, wherein nodes closer to the tree root are given higher weights than those closer to the leaves. These weighted absolute differences are then averaged to get a measure of the overall compositionality of the entire sentence.

Both methods are generalizable to subtrees whose roots have multiple children and so can be used with constituency and dependency parses. Going forward in our experiments we solely use WNS as our approximation of compositionality scores given that it is more linguistically robust than Tree Impurity.

[2] https://github.com/jacobandreas/tre

Encoder Type	SST Correlation
BERT	-0.1997
ELMo	-0.344
USE DAN	-0.485
USE Transformer	-0.168

Table 2: Rank correlations.
* p-value indicated that they are uncorrelated

4.3 Experimental Results

We use SST for compositionality evaluation with TRE. Using TRE, we evaluated the compositionality of sentence representations from BERT (Devlin et al., 2019), ELMo (Peters et al., 2018), Universal Sentence Encoder (USE) deep averaging network (DAN) and USE Transformer (Cer et al., 2018). The lower the value of TRE, the more compositional a given phrase or sentence is. Since we hypothesize that WNS is positively correlated with the degree of compositionality, then the more negatively correlated WNS is with TRE, the more compositional the sentence representation is overall. We evaluate compositionality using the rank correlation between TRE and the WNS compositionality scores.

In Table 2, we notice that the Spearman rank correlations are all negative, indicating that all the sentence representations encode some level of compositionality in their sentence representations. The more negative the correlation, the more compositional the sentence representation. By this metric, USE Transformer seems to be the least compositional while USE DAN seems to be the most compositional.

The under-performance of BERT, at least as measured by compositionality, is quite surprising given the widespread success of BERT on a multitude of downstream tasks and also the PSS metric we proposed and tested above. Given that our results are dependent on machine-generated ground truth compositionality scores, more investigation is crucial.

5 Discussion

Even though our observations from PSS and TRE approaches are not directly correlated, we observe certain consistencies and see that Transformer architectures are different compared to others. While they are more compositional as measured by PSS, they appear to be less compositional according to TRE. We believe this could be because the two methods are quantifying different kinds of compositionality. Pelletier (2011) described two different senses of compositionality; ontological and functional. TRE seems to measure more of the former since it, by nature of its definition, tries to make combination of primitives equal to the whole while PSS measures functional compositionality as it calculates a type of sensitivity which only a model with good compositional understanding can grasp. Furthermore, phrases that are similar in vector space can have opposite sentiment. For example, the *warm* and *cool* could be close in vector space, but could have a high impact on WNS.

In the current state, we believe that PSS is a more mature method to estimate compositionality for sentences especially since our extension of TRE to sentences depends on the efficacy of WNS as a good estimate of sentence compositionality. Furthermore, even if we did not use WNS and had humans tag sentences with scores for compositionality, this would still be quite hard to quantify even for humans given how subjective it can be. Expert labelers would be needed for such labeling tasks. However, looking at a positive sentence and switching its sentiment to negative or vice-versa is a much easier task for a human, so dataset creation and evaluation for PSS is much more practical.

6 Conclusion

We explored two approaches to measure the compositionality of sentence representations. Our primary contribution was proposing polarity switching as a possible measure of compositionality which correlated well with empirical results and our knowledge about inductive biases in sentence encoders. We also extended TRE as proposed in Andreas (2019) beyond bigram phrases to sentence representations. To do this, we needed to come up with a heuristic approximation of a compositional score for a sentence which we did by using weighted node switching.

7 Acknowledgments

This paper originated from our class project for Stanford's CS224U class. We would like to thank our project mentor, Ignacio Cases, for his support and guidance. This work is partly supported by the NSF Graduate Research Fellowship under Grant No. DGE − 1656518.

References

Jacob Andreas. 2019. Measuring compositionality in representation learning. In *International Conference on Learning Representations*.

Marco Baroni. 2020. Linguistic generalization and compositionality in modern artificial neural networks. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 375(1791):20190307.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Roberto Dessì and Marco Baroni. 2019. Cnns found to jump around more skillfully than rnns: Compositional generalization in seq2seq convolutional networks. *arXiv preprint arXiv:1905.08527*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Allyson Ettinger, Ahmed Elgohary, and Philip Resnik. 2016. Probing for semantic evidence of composition by means of simple classification tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 134–139, Berlin, Germany. Association for Computational Linguistics.

Alona Fyshe, Leila Wehbe, Partha P Talukdar, Brian Murphy, and Tom M Mitchell. 2015. A compositional and interpretable semantic space. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 32–41.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. 2020. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*.

Brenden M Lake and Marco Baroni. 2017. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. *arXiv preprint arXiv:1711.00350*.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874, New Orleans, Louisiana. Association for Computational Linguistics.

Joao Loula, Marco Baroni, and Brenden M Lake. 2018. Rearranging the familiar: Testing compositional generalization in recurrent networks. *arXiv preprint arXiv:1807.07545*.

Jeff Pelletier. 2011. *Compositionality*. Oxford University Press.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Vered Shwartz and Ido Dagan. 2019. Still a pain in the neck: Evaluating text representations on lexical composition. *Transactions of the Association for Computational Linguistics*, 7:403–419.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Austin Stone, Huayan Wang, Michael Stark, Yi Liu, D. Scott Phoenix, and Dileep George. 2017. Teaching compositionality to cnns. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

A Encoder Models

For BERT, we use Bert as a service[3] with parameter weights of BERTbase from Google Research[4]. For ELMo, we use AllenNLP[5] for TRE and TensorflowHub [6] for PSS. For Universal Sentence Encoder, we used TensorflowHub[7]

A.1 BERT

This model architecture is a multi-layer bidirectional Transformer (Vaswani et al., 2017). BERT was able to outperform the previous state-of-the-art on the GLUE Benchmark by 7%. The input representation can be an individual sequence or a sequence pair, such as a sentence or question/answer pairs, respectively. The final embeddings are a combination of token embeddings and special classification and segmentation tokens. For our experiments, we take the average of token embeddings to obtain the sentence embeddings. BERT was pre-trained using two novel unsupervised learning tasks: Masked Language Model (LM) and Next Sentence Prediction. BERT is that it is trained in a bidirectional manner, while other language models can only be trained using one direction at a time since being able to see the next word in the classical setting trivializes the task. In Masked LM, a certain percentage of the input tokens are masked at random, and the model is asked to predict the masked words. This allows for the preservation of a learning objective, because the transformer's encoder will not know which words it will need to predict in the future or which words have been replaced by random words, so it is forced to keep a contextual representation of every word in the vocabulary.

A.2 ELMo

ELMo (Embeddings from Language Models) vectors are derived from a bidirectional LSTM that is trained with a coupled language model (LM). We learn a weighted linear combination of the vectors stacked above each input word for each end task. Since ELMo generates three layers of embedding outputs for each word, we leverage the common pooling strategy of averaging across the layers to create a final word-level representation. Sentence-level embeddings are created by simply averaging the final word-level vectors.

A.3 Universal Sentence Encoder

We use two models of the Universal Sentence Encoder: one where the encoder is a deep averaging network (DAN) (Conneau et al., 2017) and one where the encoder is a Transformer (Vaswani et al., 2017). The embeddings are trained on tasks that demand to extract information beyond the word-level. Both models are trained with the aim of dynamically accommodating a wide variety of natural language understanding tasks. The input is variable-length English text and the output is a 512-dimensional vector.

B Synthetic data considerations

The formula for PSS calculation would be sufficient if we were fully confident that our sentiment switching model was always 100% accurate. However, as we know from Li et al. (2018), this is not the case. The polarity switching model at times generates an exact duplicate of its provided input and at other times only removes certain sentiment specific words. For the former case, it is not fair to expect any model to switch polarity, so we remove such examples. Furthermore, we also remove examples where the model only deletes (does not add) sentiment specific keywords as on manual evaluation, the model more often would remove a word/phrase that would not fully preserve the content and only at times removals resulted in negative sentiment switching to positive (e.g. removing "not"). Therefore, we only consider examples where the model adds some words in its generation that were not present in its input. Given that the model adds positive words (for a negative to positive switch), it is much more likely that if a sentiment classifier gets such an example switch wrong (cannot detect negative to positive switch), it is more a function of the sentiment classifier and therefore the sentence encoder and not an error of the data generator.

[3] https://github.com/hanxiao/bert-as-service

[4] https://github.com/google-research/bert

[5] https://allennlp.org/

[6] https://tensorflow.org/hub

[7] https://tensorflow.org/hub

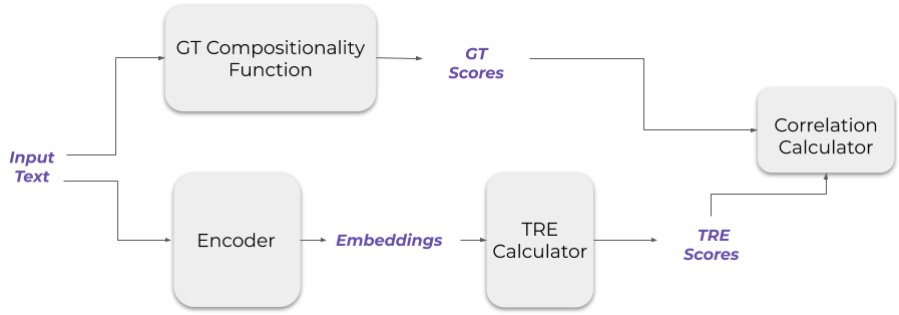

Figure 1: Experimental workflow design: TRE

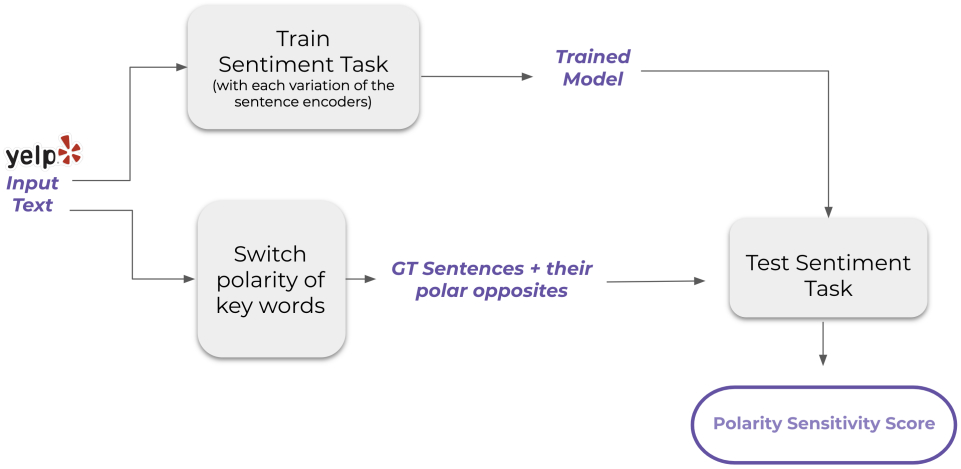

Figure 2: Experimental workflow design: Polarity Sensitivity Scoring

C Effects of the direction of sentiment switch

Given the way PSS is defined, it does not depend on the direction of the sentiment switch. As long as our ground truth sentiment label before and after switching is accurate, PSS does not differentiate between positive to negative or negative to positive switch. As mentioned above, the only source of sensitivity to the polarity switching direction comes from the sentiment switching model. Li et al. (2018) does not highlight any major differences in the direction.

Original Sentence (Negative)	Generated Sentence (Positive)
so , no treatment and no medication to help me deal with my condition . failure	so good , honest treatment and easy to help me deal with my condition .
at this location the service was terrible .	at this location the service was great .
overcooked so badly that it was the consistency of canned tuna fish .	so good that it was the best consistency of tuna fish .

Table 3: Examples of sentences output by the Polarity Switching Model.

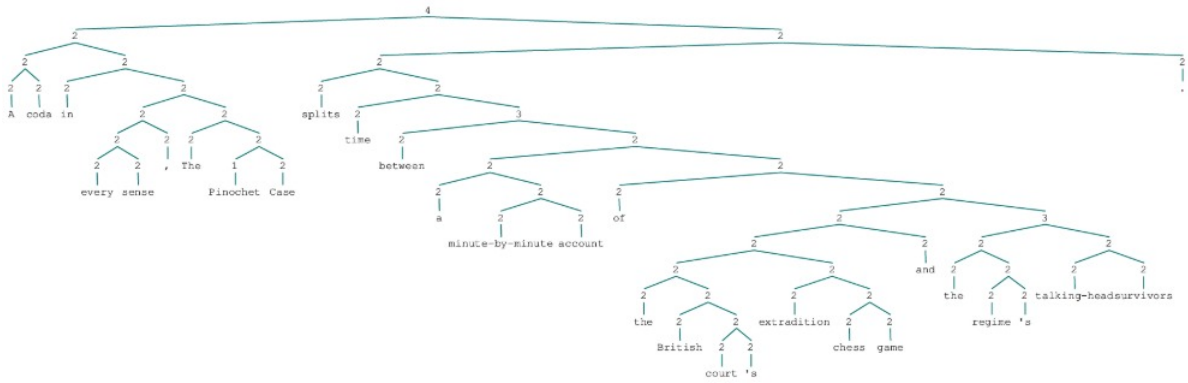

Figure 3: Examples of a sentence sentiment parse tree. The Tree Impurity metric for compositionality gives a somewhat high score of 1.95 while Weighted Node Switching gives it a lower score of 0.37. The higher score TI score is likely due to high numbers of label 1 and 2 nodes, contributing most to the overall average, which is quite different from the root node of 4. WNS, however, considers more local compositionality information which shows that most of the subtrees are not very compositional, that coupled with the overall large quantity of those subtrees, leads to the lower WNS. Additionally, WNS brings in global information via its weighting scheme which more correctly gives higher weights to when local node switches have a sentence level effect.

Sentence	TI	WNS
If Steven Soderbergh 's ' Solaris ' is a failure it is a glorious failure	2.51	1.65
A sober and affecting chronicle of the leveling effect of loss .	0.0	0.23
Cool ?	0.33	2.5
Nothing is black and white .	0.0	0.0

Table 4: Examples of sentences and their ground truth compositionality scores via both proposed metrics: Weighted Node Switching (WNS) and Tree Impurity (TI) methods. Higher scores equate to higher compositionality of the sentence. These examples represent the far ends of the spectrum on on method or the other, as 2.51 is the highest score in TI and 2.5 is the highest score in WNS, and 0.0 is the lowest possible compositionality score for both methods. One of the most telling examples of WNS's superiority over TI can be show in sentence three. Adding a "?" to "Cool" completely changes the tone of the sentence; WNS captures that nuance where TI struggles.