Measuring Pairwise Sentence Similarities

Aman Taxali

1. Motivation

Recent applications of deep learning methods to language modelling tasks have spawned a variety of context-free and contextual natural language representation models. Context-free embeddings map each word or phrase in the vocabulary to a single real vector in the continuous semantic space. A limitation of this approach is that each element of the vocabulary only has a single vector representation, even if it may have multiple meanings (for example: the word 'lead' in phrases 'lead astray' and 'lead pipe'). Contextual embeddings attempt to resolve this issue by incorporating information from neighboring words into each word's vector representation. The objective of contextual embeddings is to produce representation spaces for sequences of words (like sentences and paragraphs) that generalize well across NLP tasks.

The aim of this report is to investigate the performance of context-free and contextual embeddings in measuring the semantic similarity of sentence pairs. There are four questions we explore:

- How well can context-free and context-aware embeddings identify relatedness in sentence pairs? (section 4)
- How do NLP pre-processing techniques (such as removing stopwords, lemmatization, stemming) effect the performance of context-free and context-aware embeddings? (section 5)
- Can the performance of context-free models be improved by incorporating information about part-of-speech? (section 6)
- Do dimension reduction techniques (like PCA and TSNE) qualitatively capture the similarities between sentences? Do unsupervised clustering methods reveal meaning groups of sentences (section 7)?

2. Data Source

We let human judgement define the benchmark for measuring pairwise sentence similarities. As part of its 'Data for Everyone' platform[1], Figure Eight (formerly CrowdFlower) released a dataset of 555 English sentence pairs[2]. Each pair was evaluated for semantic relatedness, on a scale of 1 to 5, by a group of volunteers. The dataset reports the mean and variance of this 'sematic similarity score' for each sentence pair.

3. Methods

There are no missing, incomplete or erroneous records in the dataset. The 555 sentence pairs contain 1118 unique words and the average sentence length is 9.25 words.

3.1 Methods for Sentence Similarities from Embeddings

To start, we explore the differences between context-free and context-aware embeddings in estimating sentence similarities (section 4). We apply minimal preprocessing to the sentence strings here. All punctuations are removed, the sentence strings are split on whitespaces into tokens (words) and the token strings are converted to lowercase. These sequences of tokens are converted to the vector representation of a sentences using four different pretrained

1. Figure Eight - Data for Everyone: www.figure-eight.com/data-for-everyone/
2. Sentence Pair Dataset: www.figure-eight.com/wp-content/uploads/2016/03/1377882923_sentence_pairs.csv
3. Stanford GloVe: nlp.stanford.edu/data/glove.840B.300d.zip
4. Google word2vec: code.google.com/archive/p/word2vec/ (file - GoogleNews-vectors-negative300.bin.gz)
5. Facebook InferSent: github.com/facebookresearch/InferSent
6. Google Transformer Universal Sentence Encoder: tfhub.dev/google/universal-sentence-encoder-large/3

embeddings. The embeddings employed are displayed in Table 1.

Table 1

Model	Type	Dim	Pretrained Model
GloVe	Context-Free	300	Stanford [3]
Word2Vec	Context-Free	300	Google [4]
InferSent	Context-Aware	4096	Facebook [5]
Transformer	Context-Aware	512	Google [6]

With context-free embeddings, we define the vector representation of a sentence as the average its constituent token (word) vectors. Context-aware embeddings take as input a list of tokens (order sensitive) and return a single vector that represents the sequence of words in its semantic space.

The similarity score ***k*** between two sentence vectors ***x*** and ***y*** is calculated by their cosine similarity.

$$k(x, y) = \frac{xy^\top}{\|x\|\|y\|}$$

cosine_similarity equation, credit: scikit-learn.org/stable/modules/metrics.html#cosine-similarity

Lastly, the similarity scores from each of the embeddings are transformed to the same range as the human scores (1 to 5) using min-max range scaling. We analyze these scores in comparison to the human scores for accuracy of sentence semantic relatedness.

3.2 Methods for Effects of Pre-Processing on Embeddings' Performance

In section 5, we explore the effects of NLP pre-processing techniques on the performance of context-free and context-aware embeddings.

Stopword-removal, lemmatization and stemming are independently applied to each of the sentence strings. The procedures from section 4 are repeated to recompute the embedding-based similarity estimates; and results are compared to previous findings.

3.3 Methods for Augmenting Context-Free Embeddings with Part-of-Speech Information

In section 6, we attempt to improve the performance of context-free embeddings by introducing part-of-speech information as a preprocessing step. Each sentence string is tokenized, and each token is assigned a part-of-speech tag using spaCy. Only unique nouns, adjectives, verbs and adverbs are kept. The embedding vectors for words comprising each part-of-speech in the sentence are averaged to produce part-of-speech vectors. To compare two sentences, the cosine similarities between matching part-of-speech vectors are computed and averaged across all part-of-speech tags.

3.4 Methods for Dimension Reduction and Clustering on Embedding Vectors

In the final analysis section, we apply dimension reduction (PCA, TSNE) and an unsupervised clustering algorithm (K-Means) to the sentence vectors produced by the Word2Vec and Transformer models (with minimal preprocessing). We qualitatively explore the visualizations to draw any inferences about the dataset's overall semantic space.

4. Sentence Similarities from Embeddings

In this section, we aim to estimate sentence relatedness in sentence pairs using context-free and context-aware embeddings.

The results of the procedure covered in section 3.1 are presented in Table 2. This shows the mean squared errors between the similarity scores generated by the embeddings and the mean human scores.

Table 2

Model	Type	MSE
GloVe	Context-Free	1.3330
Word2Vec	Context-Free	0.6400
InferSent	Context-Aware	0.5660
Transformer	Context-Aware	0.4908

As expected, the context-aware embeddings consistently outperform the context-free models. However, the Word2Vec model is surprisingly competitive even thought it is a context-free embedding.

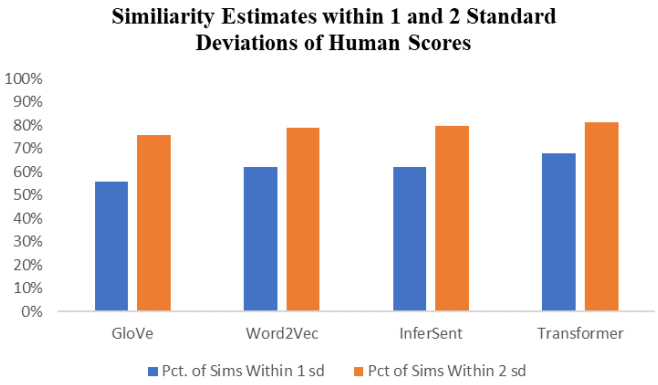
Table 3 and Figure 1 report the percentage of similarities scores generated by each embedding model that fall within 1 and 2 standard deviations of the mean human scores.

Table 3

Model	Pct. of Sims Within 1 sd	Pct of Sims Within 2 sd
GloVe	55.68%	75.68%
Word2Vec	62.16%	78.92%
InferSent	62.16%	79.64%
Transformer	67.75%	81.08%

We observe surprising consistency across embeddings when inspecting the ratio of similarity estimates that fall within 1 and 2 standard deviations of the mean human estimates. Paired with the previous MSE results, we can deduce that the context-free embeddings, GloVe and (to a lesser degree) Word2Vec, suffer from extreme errors; that is, there are instances where the context-free embeddings assign very low (or high) semantic similarity scores to sentence pairs that are actually estimated by humans to have very high (or low) sematic similarity.

Figure 1



An example of this failure in context-free embeddings is:

Sentence A: a dog is not running towards a ball

Sentence B: a dog is running towards a ball

The human mean semantic similarity score is 2.2 (variance 1.166); but the GloVe and Word2Vec similarity estimates are 4.94 and 4.86 respectively. Figure 2 shows the TSNE projections of the sentence vectors for these two phrases, for the two context-free embeddings. Both embeddings group these two sentences closely in their semantic space.

Figure 2

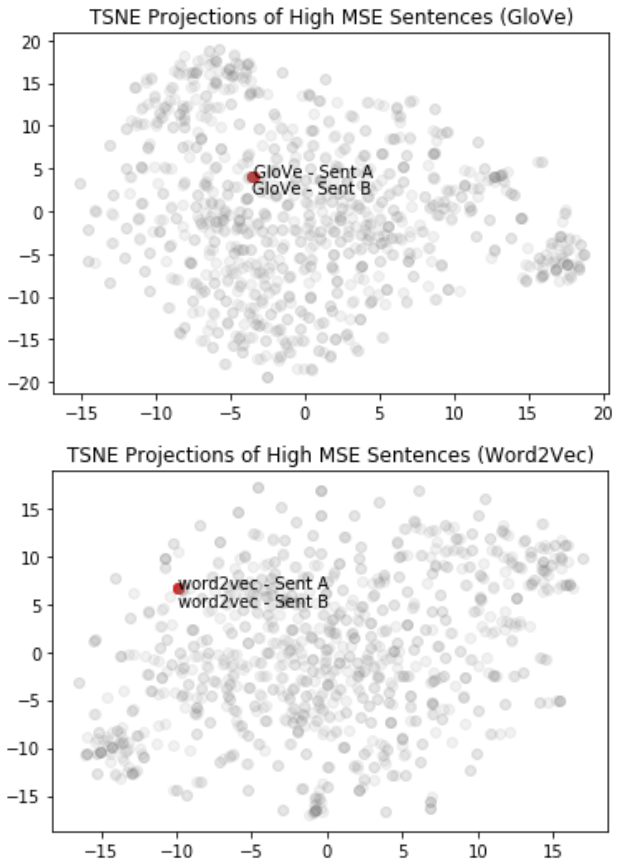
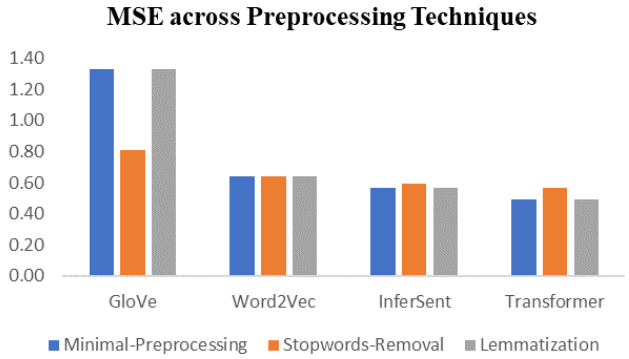


Figure 3



5. Effects of Pre-Processing on Embeddings' Performance

In Section 3, minimal preprocessing was applied to the embeddings' inputs (sentence strings). Next, we explore the effects of NLP pre-processing techniques on the performance of context-free and context-aware embeddings. Figure 3 shows the MSE results of applying stopword-removal and lemmatization to the sentence strings before converting them to embedding vectors.

We see that stopword-removal significantly improves the semantic similarity estimation performance of the GloVe model. The MSE decreases by approximately 40% (from 1.33 to 0.81). As expected, preprocessing has a negative effect on the performance of context-aware models (MSE increases as preprocessing is introduced). Lastly, the Word2Vec model showed negligible changes in sentence similarity estimation across all preprocessing schemes.

Lemmatization has no improvements over minimal-preprocessing across all embeddings. Stemming is another alternative for Lemmatization that reduces words to their base form. In our testing, we experiment with the PorterStemmer, SnowballStemmer, LancasterStemmer algorithms from NLTK. All these stemming techniques also result in minimal changes to MSE, in comparison to minimal-preprocessing.

Figure 4 shows the changes in TSNE projections of GloVe sentence vectors as stopwords-removal is added to minimal-preprocessing. In the second scatter plot, we observe clearer clustering.

Figure 4

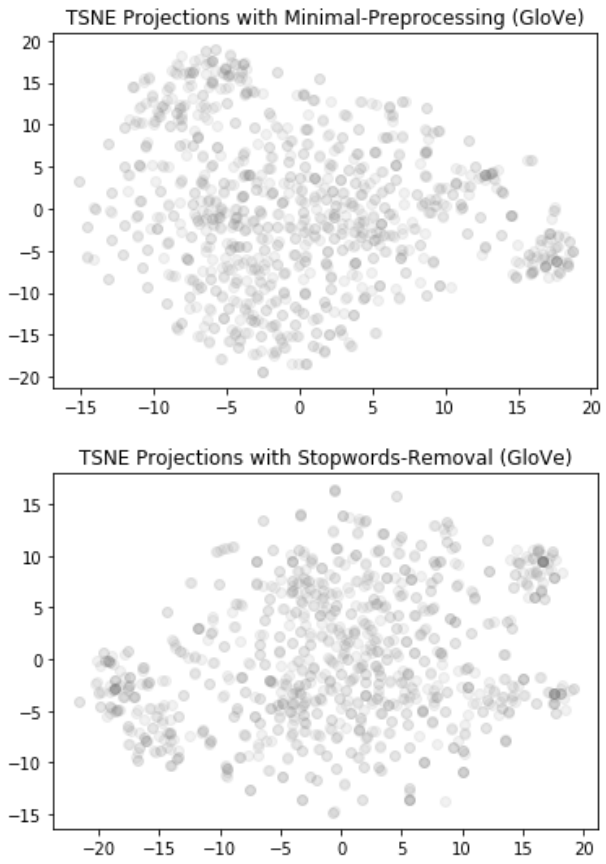
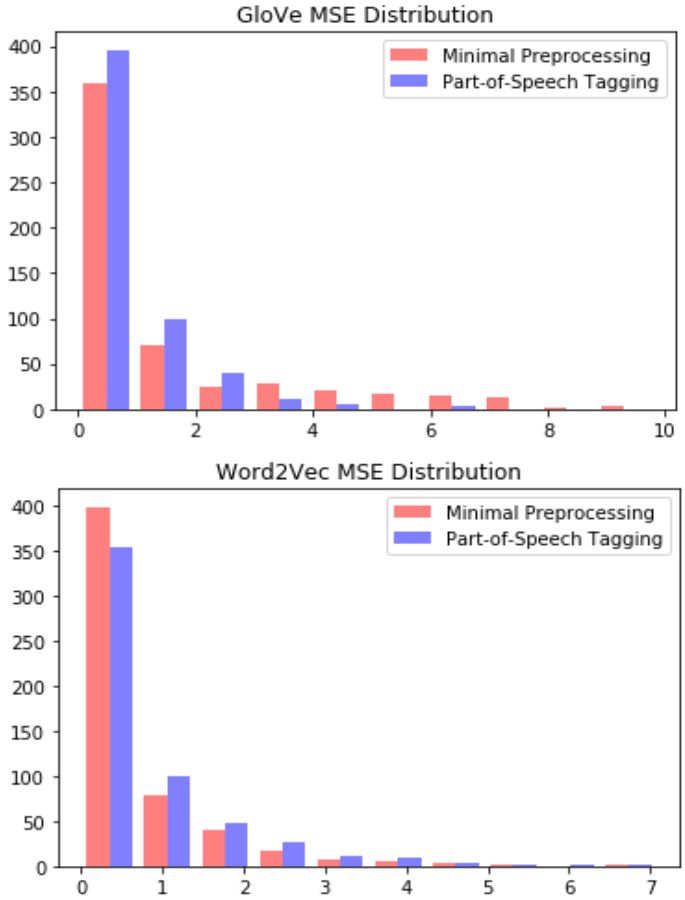


Figure 5



6. Augmenting Context-Free Embeddings with Part-of-Speech Information

In this section, we attempt to improve the performance of context-free embeddings by introducing part-of-speech tagging as a preprocessing step. We follow the methods covered in section 3.3 and repeat the analysis from section 4. Table 4 shows the mean squared errors between the similarity scores generated by the part-of-speech augmented embedding method and the mean human scores.

Table 4

Model	MSE - Minimal Preproc	MSE - POS Tagging	Improvement
GloVe	1.3330	0.7508	44%
Word2Vec	0.6400	0.7936	-24%

Interestingly, this scheme shows slightly better improvements in MSE for the GloVe model than stopwords-removal. However, the performance of the Word2Vec model worsens, compared to all our previous results.

The objective of introducing part-of-speech tagging was to limit extreme MSE errors as seen before. Thus, we revisit the previous example of failure in context-free embeddings:

Sentence A: a dog is not running towards a ball

Sentence B: a dog is running towards a ball

We saw, the human mean semantic similarity score for the pair above is 2.2 (variance 1.166) and with minimal preprocessing GloVe and Word2Vec give similarity estimates of 4.94 and 4.86 respectively. With part-of-speech tagging preprocessing, the GloVe and Word2Vec similarity estimates are 3.95 and 4.09 (both closer to human mean score). Figure 5 compares

the distribution of errors across the two preprocessing schemes and embeddings.

We can see that introducing part-of-speech information greatly reduces extreme errors for GloVe. Unfortunately, at the same time this scheme also results in an increase in small to medium errors (MSE between 0 and 3). As expected from previous results, the performance of Word2Vec is overall worse with this scheme.

7. Dimension Reduction and Clustering on Embedding Vectors

The objective of this section is to apply dimension reduction and clustering algorithms to the sentence vectors produced by the Word2Vec and Transformer models (with minimal preprocessing), in order to qualitatively draw inferences about the dataset's semantic space.

We start by revisiting the TSNE projections for embedding sentence vectors. In Figure 6, we can identify some accurate semantic clusters of sentences. The sentences in the lower left concern music and instruments, the sentences towards the center mention dogs and the sentences in the upper right area discuss cooking and kitchen items (pan, wok, carrot, onion).

Figure 7 gives the TSNE projection for the sentence vectors returned by the Transformer model. Compared to Figure 6, clustering is more evident in the Transformer vectors. We can identify closely projected pairs of sentences that discuss playing musical instruments, preparing vegetables, swimming-like activities, dogs/puppies playing and groups of men. Informally, these clusters of sentences seem semantically meaningful.

Figure 6

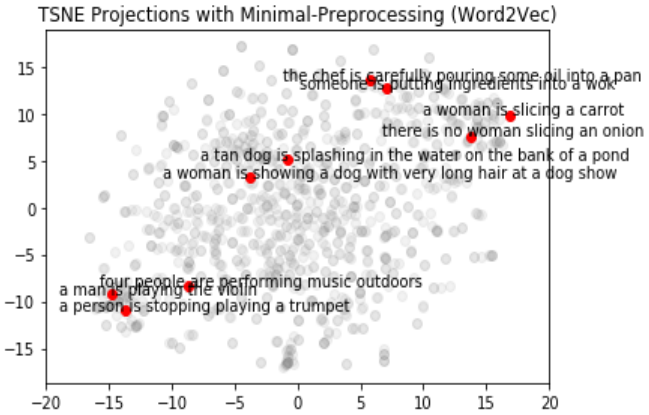


Figure 7

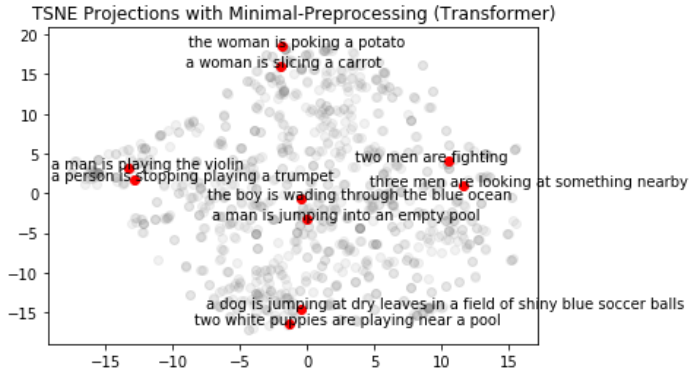
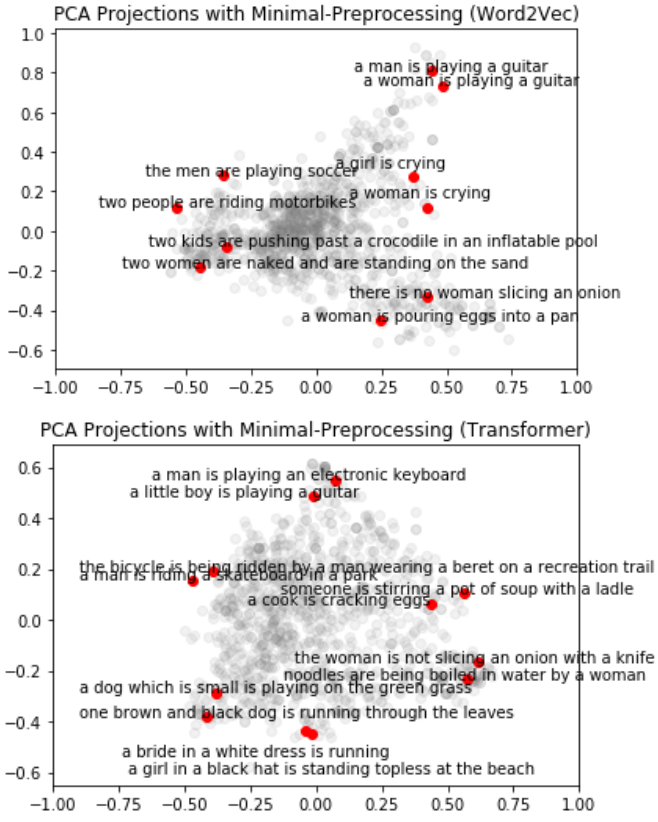


Figure 8



Figures 8 displays the projections generated by the first two principal components of PCA dimension reduction. Figure 9 shows the cumulative explained variance as the number of principal components increases. The lack of a clear kink (or elbow) in these plots suggests that there does not exist a clear PCA generated lower dimensional representation of these sentence vectors (for both embedding models).

As with the TSNE projections, semantically meaningful grouping of sentences can be identified here. Interestingly, unlike the TSNE projections, Figure 8 suggests that the PCA projections of Word2Vec sentence vectors display stronger clustering behavior than the Transformer model. However, the first two PC components only capture 14.7% and 16.7% of the total variance in the sentence vectors (Word2Vec, Transformer respectively); thus, it can be argued that these lower dimensional projections do not capture enough of the information from the sentence vectors.

Finally, we apply unsupervised clustering to the sentence vectors generated by the two embeddings. We employ the K-Means clustering algorithm and choose the ‘number of clusters’ hyperparameter based on the Scree and Silhouette scores (Figure 10). For the Word2Vec vectors, we set the number of K-Means clusters to 7, and 14 for the Transformer vectors.

In Figure 11, we label the TSNE projections with the K-Means clusters using different colors. There is strong coherence between the K-Means clusters and TSNE projections, for both models. Paired with our previous findings, we can claim that unsupervised clustering (K-Means) of embedding based sentence vectors is able to sufficiently identify semantically related groups of sentences.

Some groups of sentences identified by K-Means from the Transformer model are:

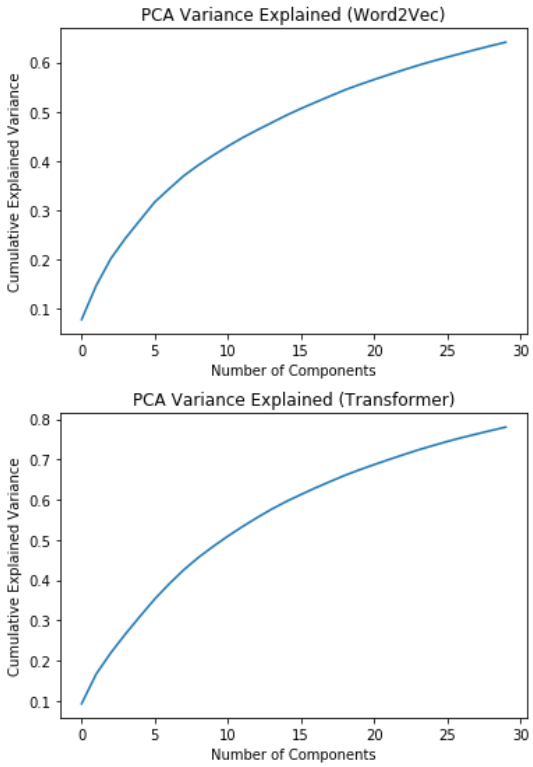
‘there is no one peeling a banana’, 'there is no woman cutting an onion', 'the men are not playing soccer'

'a woman is slicing an onion', 'a woman is frying chicken', 'the potato isnt being peeled by a woman'

'a woman is speaking on a stage', 'a girl is dancing', 'a woman is crying', 'a girl is crying'

'a man is slicing an orange', 'a man is slicing a tomato', 'a cook is cracking eggs', 'a man is cleaning a bowl'

Figure 9



8. Conclusion

In closing, both context-free and context-aware embeddings are able to sufficiently identify semantic similarities between sentences (in comparison to human estimates). Preprocessing techniques, such as stopwords-removal and part-of-speech tagging, are only beneficial for the GloVe model. The performance of the Word2Vec, Infersent and Transformer worsen with preprocessing. Dimension reduction techniques and K-Means clustering also capable of identifying semantic groupings from embedding generated sentence vectors.

Figure 10

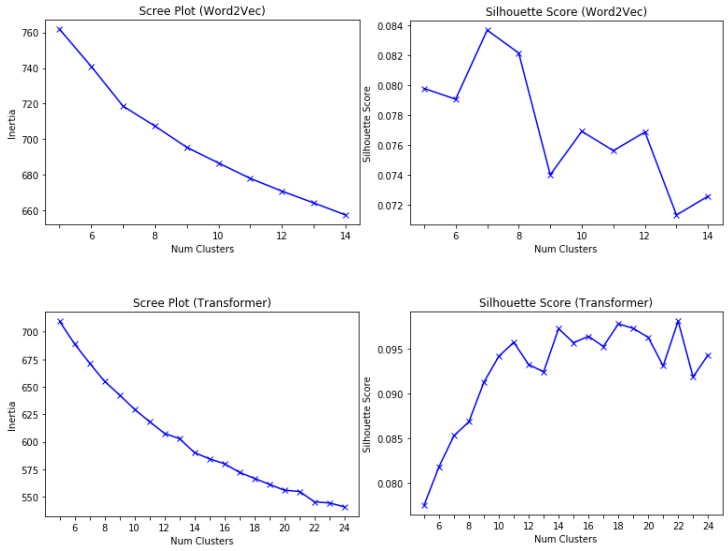


Figure 11