University of Cincinnati

Date: 11/8/2021

I, Deven Mahesh Mistry, hereby submit this original work as part of the requirements for the degree of Master of Science in Computer Science.

It is entitled:

A Systematic Comparative Study of Sentence Embedding Methods Using Real-World Text Corpora

Student's name: **Deven Mahesh Mistry**

This work and its defense approved by:

Committee chair: Ali Minai, Ph.D.

Committee member: Raj Bhatnagar, Ph.D.

Committee member: Anca Ralescu, Ph.D.

41591

A Systematic Comparative Study of Sentence Embedding Methods Using Real-World Text Corpora

by

Deven Mahesh Mistry

B.E. in Computer Engineering, University of Mumbai 2019

A thesis submitted to the Graduate Faculty of
University of Cincinnati
in partial fulfillment of the
requirements for the Degree of
Master of Science

December 12, 2021

Approved by

Ali A. Minai, Ph.D., Chair
Raj Bhatnagar, Ph.D
Anca Ralescu, Ph.D

Abstract

Many natural language processing (NLP) tasks require the conversion of textual data to numeric representations. Vector-space representations are the most popular way to do this. Initially vector-space models were used to represent individual words, but several very complex language models have been developed recently that can generate vector-space representations of sentences, paragraphs, and even entire documents. These models use various deep learning architectures including simple RNNs, stacked LSTMs, and Transformers [54]. Typically, the models are evaluated on synthetic or carefully curated benchmark datasets such as GLUE [56], SQuAD [45], COCO [55], etc. and tasks such as sentiment analysis and text classification. However, it is often unclear whether performance on these controlled benchmarks can transfer to non-curated, real-world datasets with uncontrolled semantic noise and complex structure. The goals of this thesis are: 1) To develop a methodology for systematically comparing a representative set of sentence encoder models on real-world texts; and 2) To apply this methodology using several sizeable real-world texts to arrive at a definitive ranking of the methods. The methodology uses the pattern of semantic similarity between sentence pairs to obtain a representation of semantic structure for each document using each encoding method. These structures are then compared statistically, through visualization, and through manual scoring to assess the relative quality of the representations produced by each encoding method. An innovative aspect of this research is the use of multiple English language translations of the same text as a further cross-validation mechanism.

Acknowledgments

This thesis wouldn't have come to fruition without the help of my advisor Dr. Minai. He has been a constant source of motivation throughout my entire research. Over 500 email exchanges and countless late night email threads have brought this thesis to life. It has been a wonderful learning experience which I will remember for the rest of my life.

I would also like to thank the committee members Dr. Bhatnagar and Dr. Ralescu for reviewing my thesis and taking part in my defence. I am sincerely grateful to all my friends who volunteered for the survey during the final few weeks. Being at UC has been a great experience and this entire journey wouldn't have been possible without the support from my Mom and Dad, my friends and especially my roommate, Parth. He was the one who had to deal with my idiosyncrasies and frustrations in my research for the last one and a half year. And finally, I want to thank everyone who have helped me knowingly or unknowingly throughout these 2 amazing years.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Most machine learning techniques today require numeric data for training, predicting, and classifying. The field of natural language processing (NLP) deals specifically with natural language where all the data is in the textual format. In the initial years, representing this textual data numerically was a challenge. However, due to recent advances in the fields of neural networks and computational linguistics, researchers have found several ways to embed textual data into higher dimensional vector spaces. Representing words in vectors spaces is called word embedding. These word embeddings have the ability to capture the semantic essence of words and have proven to be extremely useful for various downstream tasks such as sentiment analysis, document classification, text generation and so on. The success of these word embeddings paved the way for generating embeddings for much larger pieces of texts such as sentences, paragraphs, and documents. Several sentence embedding methods are available for researchers today, with new methods being developed every year. However, it remains a challenge to identify which embedding methods are suited to different types of datasets and how well they capture the actual semantic content in real-world texts.

1.2 Goals and Aims

The research presented in this thesis has two goals:

1. To develop a methodology for analyzing the sequential dynamics of continuous text in semantic spaces as defined by the variation of sentence similarity.

2. To use this approach to compare several state-of-the-art models for sentence representation in complex, non-curated, real-world texts.

These goals are achieved through the following specific aims:

1. Identifying and implementing several methods for obtaining sentence similarity, including those based on: 1) Lexical networks; 2) Transformer based models; 3) Bi-LSTM models.

2. Identifying a set of real-world corpora and documents for evaluation, including: 1) Multiple translations of the same non-English language texts; 2) Texts from different genres such as poetry, fiction, philosophy, etc.

3. Processing all the documents in the selected corpora through all the sentence similarity models to obtain similarity data from each document, including: 1) Sentence similarity matrices; 2) Time-series of similarity variation between successive sentences.

4. Defining a suite of analytical tests, characteristics, and metrics to characterize the data.

5. Applying these methods to the characterization, comparison, and analysis of the embedding methods on the target documents to determine the relative quality of the metrics.

The methods developed in this thesis have other applications as well, including analysis of the writing styles of authors (stylometry), characterizing the semantic structure of texts, and even the cognitive dynamics of the minds that generated those texts.

1.3 Approach

This thesis focuses on the use of various sentence embedding methods to capture the sequential semantic dynamics of text in the semantic spaces created by each method. This is done by calculating the pairwise similarity of embedded sentences using cosine similarity and other metrics. These are then used for comparing various state-of-the-art sentence encoders on a variety of corpora. In this work, encoders are tested on works of fiction, non-fiction, and poetry. The encoders used are based on various techniques such as lexical networks, transformers and LSTMs, with supervised, unsupervised and self-supervised training objectives. One source of interest for this study is the fact that the corpora chosen to test the encoders are of general interest. The works of fiction and non-fiction in this study include *A Christmas Carol* by Charles Dickens, *Heart of Darkness* by Joseph Conrad, *Metamorphosis* by Franz Kafka, and *The Prophet* by Khalil Gibran. Another interesting aspect of the study is the use of multiple English translations of books such as the *Iliad* and the *Odyssey* by Homer, the *Aeneid* by Virgil, and the *Meditations* by Marcus Aurelius.

The basic approach used in this study is to tokenize sentences and then give these tokenized sentences to the encoders to generate their sentence embeddings. For a given corpus containing m sentences and an embedding method with dimensionality n, each method gives m n-dimensional vectors, one for each sentence. The pairwise similarities are these vectors are then calculated using a cosine similarity or Euclidean distance metric to obtain an $m \times m$ *sentence similarity matrix* (SSM) for the corpus. This is visualized as a heatmap to look at the global structutre of semantic similarity in the entire corpus.

Similarly, sequential variation in the corpus is visualized by taking cosine similarity between embeddings of consecutive sentences, which corresponds to the first super diagonal in the SSM. This results in a time-series representation of sequential semantic structure of the corpus.

Once the SSMs and similarity time-series are available for all embedding methods, they can be compared in several ways. We calculate the correlation coefficients between the SSMs and time-series for pairs of methods to see whether the methods are picking up the same semantic structure. We also threshold the SSMs to unmask patterns of strong semantic similarity and compare them across methods for the same text corpus.

To evaluate whether the similarity and dissimilarity inferred by the various embedding methods correspond to human evaluations, the most similar and most dissimilar sentence pairs identified by each method are evaluated by multiple human raters using a 5-point Likert scale [33], and the correspondence between human ratings and those of the embedding methods is evaluated.

Finally, multiple translations of the same text are used as an indirect way to evaluate which embedding methods are better at capturing the essential pattern of semantic similarity in the text *independent of the specific words being used*. A challenge here is that different translations of the same text can have different numbers of sentences. To overcome this, *dynamic time-warping* (DTW) [49] is used to equalize the length of time-series generated by each pair of translations and the amount of warping needed is measured as a *warp coefficient*. The similarity of the two time-series after warping is also measured using the Pearson correlation coefficient. The ratio of the similarity and the warp coefficient is then used as a metric of *semantic consistency* between the two translations as seen by that embedding method. Given the assumption that the translations convey the same *actual* meaning, the semantic consistency is a measure of validation for the embedding method. Mean semantic consistency values calculated across all translation pairs and all base texts allows an evaluation of which method is better at capturing essential meaning.

1.4 Thesis Organization

The rest of thesis is organized as follows:

Chapter 2 describes the various language models and sentence encoders used in this study, and provides background on the *Transformer* Architecture [54], Bi-LSTMs [51], and dynamic time warping (DTW) [49] used for computing the warp coefficient.

Chapter 3 describes the methods used in this study in detail, including the process of cleaning the corpora, generating sentence embeddings, calculating sentence similarities, generating the plots, and the analysis of the results.

Chapter 4 presents all the results of the study and a discussion of the obtained results.

Chapter 5 summarizes the important conclusions of this work and suggests directions of future study .

Chapter 2

Background

2.1 Text Embedding

The purpose of text embedding methods is to generate vector space representations of textual elements – words, phrases, sentences, etc. – in a text via the process of *semantic embedding*. These representations are useful in a wide variety of NLP-based tasks such as document classification, sentiment analysis, text summarization, text segmentation, style analysis, discourse analysis, analysis of semantic coherence, etc. Early sentence representations were based on word counts and syntactic measures, but recent developments in deep learning have led to the development of ever more powerful sentence encoders obtained through machine learning with large amounts of training data. These include methods aimed explicitly at sentence encoding and others where sentence encoding is a by-product of some other NLP task such as language generation.

Before the advent of sentence encoders, there was a lot of prior work on vector space representations of words (word embedding), mainly using statistical methods. These included models such as *Latent Semantic Analysis* (LSA) [14, 31] and *Hyperspace Analogue to Language* (HAL) [36] that used dimensionality reduction methods to obtain word vectors. More recently, new machine learning approaches have led to better word embedding models such as *Word2vec* [40] and *GloVe* [42]. Word embeddings from these models are now used almost

universally in NLP applications. Computational models for embedding longer text objects such as sentences [26, 28, 1, 13] and even whole documents [32] have built on the results and insights of these word embedding models. One important class of models in this regard are *language models* that are trained to generate coherent text when primed by an initial piece of text [22, 44]. While language models do not aim explicitly to encode sentences, such encoding occurs implicitly when existing text must be encoded before subsequent text can be inferred from it. Thus, the internal representations from language models can also be used as sentence (or text) encodings.The present study uses both explicit sentence encoding models and language models in its comparative analysis. These sentence encoders are either based on the Transformer architecture [54] or the LSTM network [21]. All the encoders have been pre-trained on different corpora using different optimizers and hardware. We do not retrain or fine-tune any of the models in this study. The encoders used are described in Section 2.3.

2.2 The Distributional Hypothesis

Meaning is the core of language, but its nature is still far from clear. Most people assume – with good justification – that meaning derives from experience of the world, and is not simply a property of words. This is called *semantic grounding*, i.e., the anchoring of the meaning of a word to experienced things such as objects and situations, or internal states such as emotions and motivations. However, this is problematic from the perspective of artificial intelligence (AI) because most machine learning systems do not have access to direct experience of the world or to human-like internal states. This has led to the use of an alternative view of meaning based on the *distributional hypothesis*, which asserts that meaning lies in – or at least, can be inferred from – the joint statistics of words in natural language [52, 53]. This assumption makes it possible for a powerful statistical inference or pattern recognition system such as a deep neural network to learn semantic representations from large amounts of natural language data. While this is clearly insufficient to capture

the full complexity of language [43, 23], recent studies have shown that, remarkably, vector space word representations based on the distributional hypothesis correspond well with the brain's representations of words [59]. All the sentence encoders included in the present study are based at some level on the distributional hypothesis

2.3 Sentence Encoders

2.3.1 DeCLUTR

Deep **C**ontrastive **L**earning for **U**nsupervised **T**extual **R**epresentations (DeCLUTR) is an explicit sentence encoder proposed by Giorgi et al [19]. It is motivated by the fact that achieving state of the art results on sentence embeddings requires labelled data. To deal with this issue, the authors developed a new self supervised objective which does not require labelled training data. Their model bridges the gap between the supervised and unsupervised sentence encoders. Most state-of-the-art sentence encoders which are at top of the leaderboards for various benchmarks utilize pre-trained transformer based language models to generate vector representations of sentences. A key to their success is the Masked Language Modelling (MLM) objective that poses the problem of completing masked text. However, this objective cannot completely bridge the gap with the supervised sentence encoders trained on the Stanford Natural Language Inference (SNLI) [3] and MNLI [58] datasets. Taking inspiration from metric learning, a *pretext* task (often self-supervised) is used to train the model, without the knowledge of the downstream task the model would be used on. Once the model is trained, a simple classifier is built using the learnt features to test on the downstream task. The new objective selects an anchor point in the document and selects textual segments (or *spans*) spanning up to a paragraph around it. A contrastive loss function is then used to minimize the distance the anchor and the positive data point and maximize the distance between the anchor and negative data points. When trained with the constrastive loss or when combined with MLM, the network is able to achieve state-of-the-art results on the SentEval [11] dataset. In this thesis, two models of DeCLUTR are used which

differ in the size of the models used. DeCLUTR-small is pretrained on DistilRoBERTa, it follows the same training procedure as DistilBERT [50] and DeCLUTR-base is pretrained on RoBERTa-base [34].

2.3.2 DistilBERT

DistilBERT, developed by huggingface [50], is a lighter, faster and cheaper version of the BERT language model [15] , which is a widely used transformer-based language model trained on the MLM objective. With the size of language models increasing every year, training them is becoming extremely expensive and limits the widespread adoption of such models. DistilBERT reduced the size of the original BERT model by 40 % and retains 97 % of its capabilities. Due to its size compression, DistilBERT can also be used on IoT and mobile devices. This reduced size is possible due to the knowledge distillation techniques proposed in [20]. The model has the same architecture as BERT, but makes use of a triple loss function which combines language modelling, distillation and cosine distances.

2.3.3 RoBERTa

A **R**obustly **O**ptimized **BERT** approach (RoBERTa) is an optimized pretraining approach developed by Facebook AI [34] for BERT. RoBERTa was a replication study of BERT to study the importance of key hyper-parameters and pretraining. Their study found that BERT was significantly undertrained. Increasing the duration of pretraining on a bigger corpus leads to better results on benchmarks like GLUE [56], SQuAD [45] and RACE [30]. The original BERT model was trained on an MLM objective, where 15% of the tokens in the corpus are masked and the model tries to predict those masked tokens for several target tasks. The objective gets minimized by calculating cross entropy on the predicted tokens. RoBERTa improves the performance of BERT by making four modifications:

- Training for longer duration using longer batch sizes
- Removing the Next Sentence Prediction (NSP) objective

- Training on longer sequences of text
- Dynamically changing the masking pattern on the training data

2.3.4 InferSent

InferSent is a supervised explicit sentence encoder proposed by Conneau et al. [12]. InferSent does not use the traditional unsupervised approach to generate sentence embeddings. The authors claim that embeddings obtained through unsupervised learning do not reach the desired performance, and propose a supervised approach. The sentence encoder is trained on the Stanford Natural Language Inference (SNLI) dataset [3]. Their paper compares seven different sentence encoder architectures using LSTMs, Bi-LSTMs, GRUs, Self-Attention, and Hierarchical Convolutional nets. Their comparisons show that Bi-LSTMs with max pooling work best on transfer tasks. The model takes sentence vectors of the *premise* and the *hypothesis* of each pair from the SNLI dataset generated using same encoder, and performs three operations to extract relations between them: *concatenation*, *element-wise multiplication*, and *absolute element-wise difference*. The resultant vector is then fed into a 3-way classifier consisting of fully connected (dense) layers. The resulting trained model was shown to outperform *Skip-Thought* [28]. Two different models of the InferSent architecture are used in this thesis. InferSent-FastText which internally uses the FastText word embeddings prescribed in [25], [2]. InferSent-GloVe uses the GloVe embeddings given in [42].

2.3.5 USE

The Universal Sentence Encoder (USE) model was developed by Cer et al. [7] at Google. The goal of USE was to create sentence embeddings which could be transferred to a majority of NLP tasks. Two sentence encoders were developed, allowing trade-offs between accuracy and compute resources. The encoder achieving higher accuracy used the Transformer architecture [54]. The other encoder used a deep averaging network (DAN) [24], which gives slightly lower performance with a significant reduction in computation. The encoder based

on DAN was used in the study reported in this thesis. In a DAN, input embeddings for word tokens and bi-grams (consecutive word pairs) are averaged and fed to a deep feed forward neural network to generate sentence embeddings. DAN is able to compute sentence embeddings in linear time, where the time is dependent on the length of the input sentence.

The data used to train the encoder comes from various sources such Wikipedia, webnews, web question and answers. The unsupervised training is augmented using SNLI [3]. The DAN takes a PTB tokenized string and generates a 512 dimensional vector.

Chapter 3

Methods

3.1 Overview

The quality of sentence embeddings – and semantic representations in general – is typically evaluated on downstream tasks such as text classification, segmentation, etc. However, this only provides implicit evaluation. There is no *direct* way to assess the quality of the embeddings produced because no ground truth is available for the representations. Once an embedding is obtained, one cannot determine its 'quality' from the embedded vector itself. The embeddings generated by the deep neural network based models can have hundreds, or even thousands of dimensions, so even visualizing them is a challenge [9]. The work in this thesis uses a new approach to evaluate the quality of semantic representations: Using the pattern of semantic similarity between the sentences in sizeable real-world documents (books). As discussed in Chapter 1, this is accomplished through two studies:

- **Study I - Comparison of Embedding Methods:** In this study, sentence embeddings are obtained from several real-world documents using seven different embedding models. These embeddings are then compared pairwise to evaluate the similarities and differences between the representations produced by the models. Overall, this shows

whether all models are picking up on similar semantic information, or if they are paying attention to different aspects of meaning.

- **Study II - Evaluation of Relative Semantic Validity:** While Study I provides a comparison of the representations produced by different models, it does not make any quality assessment. In Study II, two methods – one very direct, the other a little indirect – are used to assess the actual semantic validity of the embedding methods. In the direct method, sentence pairs deemed most similar and least similar in a document by each method are given to several human raters and the resulting ratings are compared with those generated by the embedding methods. In the indirect method, each method is used to embed multiple translations of the same non-English document and the consistency between the embeddings is seen as a measure of the embedding method's semantic inference.

Since the sentence embedding models are compared using the pattern of sentence similarity, this work also provides the opportunity to compare these results with a lexical networks-based approach to calculating sentence similarity that has been used in previous research [39, 38].

Both studies and the lexical network-based model are described in detail later in this chapter.

3.2 NLP 'In the Wild'

A primary motivation for the research in this thesis is to evaluate semantic representation models "in the wild", i.e., on texts that have not been carefully selected and curated, but are truly real-world, non-trivial natural language texts.

Carefully curated benchmark text corpora [45, 30, 56] are used widely to evaluate the performance of the embeddings and obtaining a numeric value as a metric. In some cases, they may actually be artificially constructed (e.g., benchmark sets of sentences [45, 30, 56]

or carefully chosen segments from multiple texts [8]), but even when they are real texts, they are often chosen to be relatively noise-free. In contrast, we use books of general interest and distinct types (poetry, fiction, philosophy) without any curation or filtering.

This approach also provides deeper insight into the relative strengths and weaknesses of the semantic models. While recent experiments in neuroscience have suggested that word embeddings produced by machine learning (ML) correspond to mental representations of concepts [59], the issue of the cognitive accuracy of ML-based sentence representation models remains open. By evaluating these models on documents such as works of literature that are the result of individual authors' natural trains of thought, the present research is also of interest from the perspectives of cognitive science and digital humanities.

3.3 The Principle of Mutual Consistency

As pointed out above, no ground truth is typically available to validate the language models. To address this, the research in this thesis proposes a *principle of mutual consistency* as the basis of collective validation. This section describes this principle and its underlying assumptions:

1. **Assumption 1:** Every document has a specific (but latent) *intrinsic meaning* and any successful semantic representation method must capture this.

2. **Assumption 2:** A specific intrinsic meaning implies a specific *semantic structure* in a document, and any successful semantic representation method must infer the same semantic structure for a given document

3. **Assumption 3:** The semantic structure of a document can be represented as the *pattern of semantic similarity* between the sentences of the document.

4. **Assumption 4:** Meaning is an *emergent property* of text, and is not contained wholly in its word content. Similarity or difference in the word content of two texts is neither necessary nor sufficient to determine similarity or difference of meaning.

The Principle of Mutual Semantic Consistency (PMSC):

1. **PMSC-a**:If two sufficiently different semantic representation methods infer mutually consistent semantic structures for a document, they must both be inferring its underlying true intrinsic meaning.

2. **PMSC-b** If two semantic representation methods infer very different semantic representations for the same document, one or both must have failed to capture its intrinsic meaning.

3. **PMSC-c** If two documents have the same intrinsic meaning, a successful semantic representation method should infer the same (or very similar) semantic structure for both of them.

Essentially, the PMSC proposes that the specific semantic structure of a document, as represented in its sentence similarity pattern, can be used as an observable surrogate representation for its meaning, and if very different semantic representation methods infer consistent structure for a document, they must be capturing the ground truth, even though the ground truth is not known explicitly.

The Principle of Mutual Consistency can also be used to validate the quality of an individual semantic model by applying it to multiple well-regarded, unabridged translations of the same non-English text. Since the original text is assumed to have a well-defined intrinsic meaning, each translation can be seen as a somewhat distorted sample of it that still conveys the same meaning as a whole. Thus, if the semantic structures extracted from these translations by a semantic model turn out to be similar, it can be argued that these representations are capturing the deeper underlying meaning each translation has inherited from the original. And, *since each translation uses different words and, of course, no translation uses the same words as the (non-English) original, any similarity in the extracted structures is*

purely semantic, i.e., independent of the words used. This can be seen as a kind of "semantic cross-validation" based on the Principle of Mutual Consistency.

3.4 Text Corpora

The books chosen in this thesis are of general public interest. These books have been around for a long time and provide a good sample of literary writing of different types. In the case of multiple translations, the genres of philosophy and poetry are chosen because it was important that the translators should be aiming to convey the meaning of the original completely, faithfully, and without abridgment. The books and their respective abbreviations used in this thesis are listed in table below.

Name of the Book	Author	Label
A Christmas Carol	Charles Dickens	
Metamorphosis	Franz Kafka	
Heart of Darkness	Joseph Conrad	
The Prophet	Khalil Gibran	
The Iliad	Homer (translated by Alexander Pope)	I1
The Iliad	Homer (translated by Samuel Butler)	I2
The Iliad	Homer (translated by George Chapman)	I3
The Iliad	Homer (translated by Andrew Lang, Walter Leaf and Ernest Meyers)	I4
The Odyssey	Homer (translated by Samuel Butler)	O1
The Odyssey	Homer (translated by Alexander Pope)	O2
The Odyssey	Homer (translated by Butcher & Lang)	O3
The Odyssey	Homer (translated by William Cowper)	O4
The Aeneid	Virgil (translated by John Dryden)	A1
The Aeneid	Virgil (translated by Rolfe Humphries)	A2
The Aeneid	Virgil (translated by J. W. Mackail)	A3
Meditations	Marcus Aurelius (translated by Meric Casaubon)	M1
Meditations	Marcus Aurelius (translated by George Chrystal)	M2

Table 3.1: List of Books

3.5 Semantic Models

3.5.1 Sentence Embedding

Embeddings for these seventeen books are generated by the seven methods discussed in 2.3. DeCLUTR, InferSent, USE are explicit sentence encoders whereas DistilBERT and RoBERTa are language models. The code implementations to generate sentence embeddings from DistilBERT and RoBERTa are taken from [46] whereas for DeCLUTR, InferSent and USE their official git repositories are used. All these sentence encoders and language models need a list of n sentences as input and they output a $n \times d$, where d is the dimensionality of the encoder, mentioned in 3.2. The dimensionality of the embeddings produced by those methods is given below.

Method	Dimensionality
DeCLUTR Base	768
DeCLUTR Small	768
InferSent GloVe	4096
InferSent FastText	4096
DistilBERT	768
RoBERTa	1024
USE	512

Table 3.2: Dimensionality of embeddings

No retraining or fine-tuning is done on any of the models.

Calculating Sentence Similarity

Many distance metrics are available for vector spaces, but they vary in their utility for measuring similarity between semantic embeddings, which are very high-dimensional. Perhaps the most widely used metric is *cosine similarity*, which is the cosine of the angle between the two vectors [41], [42] and [6]. If A and B are two vectors in the same vector space, their cosine similarity is given by:

$$sim(A, B) = \frac{A \cdot B}{\|A\|\|B\|} \tag{3.1}$$

This is 1 when the vectors point in the same direction in the embedding space, 0 when they are orthogonal, and -1 when they point in opposite directions. Since semantic embeddings typically use only the positive hyperquadrant of the embedding space, cosine similarities are between 0 and 1.

Throughout this thesis, the similarity between two sentences is calculated using cosine similarity.

3.5.2 Lexical Weights Model

An alternative to neural sentence encoders and language models is to directly use co-occurrence information between words to calculate sentence similarity, though this does not explicitly produce representations of sentences. This method has several advantages over the embedding-based approaches:

1. It can be calculated directly over any corpus rather than just using pre-trained representations. This is important because the semantic relationships in specific corpora can be very different from those in the pre-training corpora.

2. The computations involved are quite simple and can be performed on relatively small corpora. No iterative computation or training is required.

3. The sentence similarities calculated can easily be interpreted and explained because they are based explicitly on the words in the two sentences and their relationships.

4. There is a prior history of using PMI-based metrics in semantic representations [35, 4, 47, 16, 39, 38], and it is interesting to see how much sentence-level semantics can be captured through it.

On the negative side, the lexical approach – because of its simplicity – may not be able to capture deep semantic relationships, and, while it can be inferred from smaller corpora, applying it to very large corpora can be computationally expensive.

The lexical network approach begins by constructing a *lexical network*, whose nodes are all the words in the corpus vocabulary (or some subset of these), and the edges represent some semantic association between the pair of words they connect as inferred from the corpus. Several metrics of association have previously been considered in work in our lab [16, 39, 17, 5, 37, 38]. Broadly, the association between two words is based on whether they occur close to each other in the corpus. Proximity is evaluated by using a *reading frame*, which can be a fixed size neighborhood around each word token or, as in the case of the present study, an individual sentence. For a given text corpus, the *co-occurrence probability* p_{ij} of words w_i and w_j is the probability that they occur in the same reading frame, i.e., the same sentence. Thus, p_{ij} is the fraction of sentences in the corpus that include both words. Similarly, the *marginal probability* p_i of word w_i is calculated as its the fraction of reading frames (sentences) containing the word. The association weight between w_i and w_j can be calculated from p_{ij}, p_i, and p_j. The three types of association weights we have considered previously are:

1. **Joint (Co-occurrence) Probability (CP):** The probability (relative frequency) of words w_i and w_j occurring in the same reading frame:

$$a_{ij} = p_{ij} \tag{3.2}$$

2. **Correlation Coefficient (CC):** A measure of the covariance in the occurrence of w_i and w_j in the same reading frame [48, 16]:

$$a_{ij} = \frac{p_{ij} - p_i p_j}{\sqrt{p_i(1-p_i)}\sqrt{p_j(1-p_j)}} \tag{3.3}$$

3. **Pointwise Mutual Information (PMI):** This is a measure of the statistical dependence in the occurrence of w_i and w_j in the same reading frame [10, 35]:

$$a_{ij} = \log \frac{p_{ij}}{p_i p_j} \tag{3.4}$$

Since both CC and PMI produce small and uniformative negative association values between most word pairs, the measures are rectified by zeroing out these negative values [48]. Other related metrics such as the log-odds ratio are also be used [35].

Of these measures, PMI is the most widely used, and that is used in the present study. It is usually normalized to give values between 0 and 1. In this thesis, the following calculation is used:

$$a_{ij} = PMI_{ij} = \log \frac{N\ N_{ij} + 1}{N_i N_j + 1}, \tag{3.5}$$

where N is the total number of sentences in the corpus, N_i and N_j are the number of sentences containing word w_i and w_j respectively, and N_{ij} is the number of sentences containing both words. The addition of 1 in the numerator and denominator is for regularization.

Calculating Sentence Similarities from Lexical Networks

The PMI matrix calculated using the formula 3.5, gives a $n \times n$ matrix (network of nodes), where n is the total number of words in the given corpus. To compute a sentence level network from this PMI matrix respective weights for the word pairs are used. Suppose PMI is the normalized version of the PMI matrix for a corpus. If sentence a contains m unique words and sentence b contains n unique words, the sentence similarity between the two sentences is calculated using the formula,

$$W_{sent_{a,b}} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} PMI_{i,j}}{mn} \tag{3.6}$$

3.5.3 General Methods

Text Pre-Processing

The entire process first starts with cleaning text files. All the books used in this thesis have been taken from the Gutenberg Project. The first four books mentioned in Table 3.1 don't have multiple translations. While cleaning those four books, sections like the preface, table of contents, chapter headings, page numbers were removed as a part of the prepossessing. This cleaning is done using various regex expressions. In case of any outliers, or missing cases, few sentences are removed manually. Similarly, for cleaning the rest of the books (books of multiple translations) in Table 3.1, similar steps are followed. Prefaces, chapter headings, original lines from the book, figures and verse numbers (for poems), roman numerals were all removed. Once the books and translations are cleaned, they are split into individual sentences such that, each sentence appears on a new line. This process is called *tokenization*.

For the generation of the lexical network (PMI matrix), stopwords are also removed from the corpus. Due to the high frequency of stopwords, their removal helps with the determination of word co-occurrences of other words in a better manner as stopwords tend to occur with a lot of words. In cases, where sentences only contained stopwords for a given corpus, their removal decreases the size of the original corpus. These missing sentences are interpolated with the global mean of its SSM.

Sentence Similarity Analysis

These tokenized sentences are then fed into the sentence encoders to generate the sentence embeddings. Pairwise cosine similarities are calculated between all pairs of sentences in the document with n sentences, giving an $n \times n$ *sentence similarity matrix* (SSM) that captures the global pattern of meaning across the whole document. The lexical network model gives sentence similarities directly. For easier comparison along the eight methods, all

Books and their authors	Sentences	Word tokens
A Christmas Carol	1942	29116
Metamorphosis	795	22373
Heart of Darkness	2430	39061
The Prophet	647	12360
The Iliad (translated by Samuel Butler)	4192	153580
The Iliad (translated by George Chapman)	3974	169844
The Iliad (translated by Andrew Lang, Walter Leaf and Ernest Meyers)	3501	138611
The Iliad (translated by Alexander Pope)	5334	151891
The Odyssey (translated by Butcher & Lang)	3723	135589
The Odyssey (translated by Samuel Butler)	3139	117938
The Odyssey (translated by William Cowper)	4952	115358
The Odyssey (translated by Alexander Pope)	3950	112559
The Aeneid (translated by John Dryden)	4360	112387
The Aeneid (translated by Rolfe Humphries)	3493	85978
The Aeneid (translated by J. W. Mackail)	3659	98612
The Meditations (translated by Meric Casaubon)	1996	57048
The Meditations (translated by George Chrystal)	1981	40974

Table 3.3: Sentences and word tokens in each book

the generated SSMs are either normalized between 0 and 1, or standardized by converting the values to z-scores.

Visualizing the entire SSM at a single glance can be challenging for larger corpora which can contain thousands of sentences. Finding and visualizing the minor details become difficult in such cases. This is overcome by visualizing smaller (e.g., 200×200) patches along the diagonal of the SSM separately. Such plots are referred to as the *sectional heatmaps* in the thesis. An addition to these plots, the entire skeletal structure of the SSM is also visualized. This is done by thresholding the value of SSM at the mean. This thresholded plot highlights areas and structures of very high similarity in the text. These plots are referred as the *thresholded SSMs* in the thesis.

The SSMs allow the visualization of the entire corpus at a single glance. However, text is inherently sequential, and it is especially interesting to look at the time-series of similarities between successive sentences. To visualize this temporal structure, a sequential time-series for a given corpus is created. The plot can be taken from the first super diagonal of the SSM. These time-series are also normalized between 0 and 1 for comparison, though the normalizing factor for the time-series may be different than that for the SSM of the same document. Once a time-series from a given SSM is generated, it is plotted as a $1 \times n$ heatmap.

To obtain a numerical value of pairwise similarity between the time-series generated by the different embedding methods, a Pearson correlation coefficient is calculated:

$$r = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sqrt{\Sigma(x - \bar{x})^2 \Sigma(y - \bar{y})^2}} \tag{3.7}$$

where x is the first time-series and y is the second time-series. Similarly, a pairwise correlation coefficient value is obtained from the SSMs, measuring the similarity between them.

3.6 Study I: Comparison of Methods for Semantic Structure

The encoders used in this thesis have different objective functions, architectures and datasets. These encoders have produced good results on their respective benchmarks and downstream tasks, but the intrinsic validity of embeddings produced by them is unclear. When these encoders are used to embed the same text, they generate different embeddings. This first study simply compares the patterns of sentence similarity produced by all encoders on the same set of documents

It is motivated by the following question: *How mutually consistent are the different representations produced for each document?* This is motivated by the idea that every document has a fixed meaning, and good sentence encoders must produce similar semantic structure representations for it. If none of the representations are mutually consistent, that opens up a question of which one is picking up the true meaning. If most are consistent, that increases confidence that those encoders are actually representing the "true" underlying meaning of the document. And if one or two representations are very different than the rest, it can be asked whether they are just less correct, or if they are picking up on a different aspect of meaning.

While this study does not answer all these questions, it provides a systematic and quantitative comparison of the similarity and difference between the representations produced by different encoders.

The comparison between the encoders is done using all the documents, and two types of data:

1. The sentence similarity matrix (SSM) produced by all seven sentence encoders and the lexical network method for each document. Each SSM is normalized to be between 0 and 1 to remove the effect of range bias in different representations.

2. The time-series of similarity between successive sentences for each of the eight methods on each document. Each time-series is normalized to be between 0 and 1 to remove the effect of range bias in different representations.

For both of these, similarity is computed between pairs of representations using Pearson's correlation coefficient. The resulting 8×8 matrix is then shown as a *similarity heatmap* with the values noted in each cell. Thus, with M documents, the study produces M similarity heatmaps for the SSMs and M for the time-series. It should be noted that, while the time-series for any case comes from the first superdiagonal of the SSM for the same case, the normalizations for the two are different because each uses its own minimum and maximum values.

3.7 Study II: Evaluation of Relative Semantic Validity

While Study I simply compares the sentence similarity representations generated by each sentence encoding method, Study II attempts to assess their ability to capture "true" meaning. Thus, Study I relies on looking at different encoding methods while keeping the documents constant, while Study II evaluates individual methods across document pairs that are semantically identical but use different words. while keeping the embedding methods constant. The first study gives insight about how different encoders capture the semantic essence of a corpus in different dimensions while the second study evaluates the quality of each individual method.

Two methods are used to do the assessment of the methods: 1) Comparing the extreme sentence similarity and dissimilarity values produced by the eight methods with scores produced by several human raters; and 2) comparing the semantic consistency between the sentence similarity patterns of multiple translations of non-English documents.

3.7.1 Using Human Raters

The sentence embeddings which are generated by sentence encoders are designed to be *universal*, i.e., applicable to any NLP task. They are typically evaluated based on their performance on multiple tasks such as text classification, text generation and segmentation. A significant amount of NLP research is driven by this requirement to achieve better scores on these benchmarks, which in the recent years has led to the rise of bigger and bigger language models. However, these evaluations only provide an indirect test of encoding model's quality. There is no "ground truth" for embeddings per se, only for the application tasks where they are used. The present study tries to remedy this through the use of human raters.

The SSM generated for a given document D_r by a method Q_s provides assignments of semantic similarity for every pair of sentences in the document. At the most basic level, this just displays how meaning is structured in the documents – e.g., which parts are similar to each other, or where the topic under discussion changes. But the SSM also makes it possible to see which sentence pairs in D_r are considered semantically similar by method Q_s and which are not. In this part of the study, sets of most similar and least similar sentence pairs in the representation are taken, and human raters are asked to evaluate their similarity independently. The human ratings are then matched with those from model Q_s. The more consistent a model's ratings are with the human raters, the closer its assessment of semantic similarity is considered to the ground truth.

The procedure for this study is as follows:

1. For each document D_r and each method Q_s, the normalized SSM, S^{rs} is standardized by turning the value of each cell, S^{rs}_{ij} into a z-score:

$$Z^{rs}_{ij} = \frac{S^{rs}_{ij} - \mu^{rs}}{\sigma^{rs}} \tag{3.8}$$

   where μ^{rs} is the sample mean of S^{rs} and σ^{rs}its sample standard deviation calculated as

$$\sigma^{rs} = \sqrt{\frac{\sum_{i,j} (S_{ij}^{rs} - \mu^{rs})^2}{N-1}}$$

where $N = n_s^2$ with n_s denoting the number of sentences in the document.

2. From the standardized SSM Z^{rs}, the k most negative values are determined, and the set of the corresponding sentence pairs, L^{rs} is designated the *least similar sentence pairs* (LSSP) for document D_r as inferred by method Q_s. Similarly, the set of most positive *non-diagonal* entries in Z^{rs} are used to get the set M^{rs} of the k *most significant sentence pairs* (MSSP). In the present study, $k = 10$, so each method and document yields 20 pairs. With four documents (A Christmas Carol, Heart of Darkness, Metamorphosis, The Prophet), and eight methods, a total of $20 \times 4 \times 8 = 640$ sentence pairs is obtained. This is designated the *sentence pair probe set* (SPPS)

3. The sentence pairs in SPPS are compiled in an Excel spreadsheet with each sentence pair occupying two adjacent columns in a distinct row. All other data for each sentence pair (model, source document, similarity value, z-score, etc.) is also included, each in its own column.

4. An *evaluation copy* of the SPPS Excel spreadsheet is generated for each human rater by removing all information except the text of the sentence pairs, and all the rows are shuffled randomly and independently for each rater. Each rater is then given their own evaluation copy and asked to rate each sentence pair for semantic similarity on a 1-to-5 integer Likert Scale [33].

5. Once the evaluations are received from all the raters, each evaluation copy is sorted so the sentence pairs in all of them are in the same order. This order is hierarchical by similarity type (least/most), method, and source document, respectively: The first 320 pairs are the LSSP and next 32 the MSSP; within each of these, the pairs from each method are grouped together; and within each method's group, the pairs for the same

source document a grouped together. The resulting data produces a (rater)×(sentence pair) matrix called the *pair ratings matrix* (PRM), which is displayed as a heatmap and used for further analysis. The score of rater u on pair v in the PRM is denoted by ζ_{uv}.

6. The raters are validated for consistency using the method of *intraclass correlation coefficient* (consistency of two-way random raters) [29]. The raters used have an ICC of 0.90.

7. For each method, the means and standard deviations of each individual rater's scores on the LSSP pairs and MSSP pairs are found over all four documents (i.e., 40 scores for each rater). The histograms of these two sets of scores for each rater are also obtained. These two things show how each rater scored LSSP and MMSP sentence pairs.

   To get a quality score for each method, Q_s, based on the human ratings, the mean LSSP and MSSP scores across all raters are calculated separately:

$$\rho_L^s = \frac{1}{320 n_r} \sum_u \sum_{v \in LSSP} \zeta_{uv} \tag{3.9}$$

$$\rho_M^s = \frac{1}{320 n_r} \sum_u \sum_{v \in MSSP} \zeta_{uv} \tag{3.10}$$

   where u indexes the raters and n_r is the number of raters. The semantic quality score for method Q_s is then given by:

$$G^s = \rho_M^s - \rho_L^s \tag{3.11}$$

   , i.e., the greater the separation between the scores raters assigned to the LSSP and MSSP produced by a method, the closer the method is to human assessments.

The method described above looks only at the cases of most similar and most dissimilar sentence pairs, i.e., extreme cases. This was done for two reasons: First, to get the clearest possible evaluation, and second, to put no more than a reasonable load on the raters.

3.7.2 Using Multiple Translations

An interesting and novel method for comparing semantic representation methods is to use multiple translations of the same non-English documents. This approach complements the method using human raters because it provides an intrinsic comparison rather than one dependent on human opinions. At the same time, it is more indirect, and captures only some aspects of representation quality. Most importantly, it focuses on the *meaning* of text rather than the exact words used because, while each translation uses its own words, it is trying to convey the same meaning.

Thus, the foundation of this analysis is on two principles:

1. **Principle 1:** All unabridged and high-quality translations of the same text convey the same meaning, even though they use different words.

2. **Principle 2:** Two texts with different words but the same meaning should produce the same (or similar) representations.

Thus, a semantic representation method applied to translations that satisfy Principle 1 should produce representations that are similar. Each representation can be seen as an "estimate" of the *latent ground truth*, i.e., the true underlying meaning of the original text. By comparing the consistency between the semantic representations that a method produces for several translations of the text, one can assess how well the method has captured this ground truth, and thus its ability to represent meaning independent of the exact words being used. The main limitation of the approach is that each translator brings different biases to their work, and it is impossible to qualify how well they have captured the true meaning of

the original. However, since all methods are evaluated on the *same* set of translations, the method can justifiably be used for comparison between them.

In Study II, we have compared four different translations of The Iliad and The Odyssey, three translations of The Aeneid, and two translations of the Meditations of Marcus Aurelius. The details are given in Table 3.1. Embeddings for these translations are produced in the same way as described in Sections 3.5.3 and 3.6.

One complication is that different translations of the same document can have different numbers of sentences, making a direct comparison of their semantic structure more difficult. Thus, a challenge is to first map all translations of the document to modified structural representations of the same size in a principled way and then compare them. In particular, the sentence similarity time-series for the translations must first be equalized in length. This task of comparing temporal sequences of various lengths has been widely explored in the fields of signal processing and electronics [27], [57], [49], etc. In this thesis, we use a technique called *dynamic time warping* (DTW) as a method to compare these embeddings of translations.

Dynamic Time Warping (DTW)

Dynamic time warping was introduced in 1978 [49] as a way to compare temporal sequences of different lengths. The algorithm prescribed in the original paper is of $O(n \cdot m)$ complexity, where n in the length of the first time-series and m is the length of the second time-series. The pseudo-code of the original algorithm is given below [18].

The algorithm finds an optimal match between the time steps of both the time-series based on a cost function (distance metric). The distance metric used in the original was the **absolute difference between the two time steps**. Following the algorithm, tracing the smallest value from $DTW[n, m]$ (top right corner in the DTW matrix), the warping path is retrieved. The projection of this warping path on either axes give the warped time-series for the time-series s and t. The length of the warped series cannot be controlled. The degree to

Require: $s \leftarrow array[1..n], t \leftarrow array[1..m]$
  $DTW \leftarrow array[0..n, 0..m]$
  for $i \leftarrow 0$ to n **do**
    for $j \leftarrow 0$ to m **do**
      $DTW[i, j] \leftarrow infinity$
    end for
  end for
  $DTW[0, 0] \leftarrow 0$
  for $i \leftarrow 1$ to n **do**
    for $j \leftarrow 1$ to m **do**
      $cost \leftarrow d(s[i], t[j])$
      $DTW[i, j] \leftarrow cost + min(DTW[i-1, j], DTW[i, j-1], DTW[i-1, j-1])$
    end for
  end forreturn $DTW[n, m]$

which a time-series series gets warped depends on the distance metric and the actual values in the time-series. The algorithm also has a subtle caveat, the warping procedure can only take 2 time-series at a time. So this would mean that when multiple translations are being compared, only 2 translations of a book can be compared at once. Thus, in case of *The Iliad* which has 4 translations, $\binom{4}{2} = 6$ pairs are evaluated. Another point to note is that the time warping procedure is commutative.

The purpose of time-warping is to make the two time-series as consistent as possible, but this cannot be done perfectly and some discrepancy remains. This is one indication of how inconsistent the two time-series were to begin with. Another measure of this is the amount of warping needed to bring them to optimal consistency. The more inconsistent they are, the more they have to be stretched for them to fit with each other. The latter is defined as the *warp coefficient*, and measured as the percentage change between the length of the original time-series and warped time-series. As two time-series are warped, two warp coefficients are obtained and the mean warp coefficient of the DTW process is calculated as:

$$\begin{aligned} \omega_c^1 &= \frac{W_l - O_l^1}{O_l^1} \\ \omega_c^2 &= \frac{W_l - O_l^2}{O_l^2} \end{aligned} \tag{3.12}$$

$$\Omega_c = \frac{\omega_c^1 + \omega_c^2}{2} \tag{3.13}$$

where where ω_c^1 is the warp coefficient of the first time-series, W_l is the length of the (new) warped time-series and O_l^1, O_l^2 are the original lengths of the first and second time-series, respectively. The average warp coefficient Ω_c is the mean of W_c^1 and W_c^2. In an ideal case, i.e., if the time-series were identical, the value of the warp coefficient would be 0.

Once the time-series are warped, the calculation of their Pearson correlation coefficient becomes possible. This provides a measure of the success of the warping process. Based on the value of the average warp coefficient Ω_c and the correlation coefficient, a new composite metric Q is devised which is used as a quality metric to get a score for each pair of translations based on each semantic representation method.

$$Q = \frac{z_{s,t}}{\Omega_c} \tag{3.14}$$

where $z_{s,t}$ is correlation coefficient between the two time-series and Ω_c is the average warp coefficient for that case.

Chapter 4

Results and Discussion

4.1 Results from Study I

4.1.1 SSMs for the Four Literary Books

From the procedures described in 3.6, plots to visualize the global semantic structure (SSMs) and sequential variation in the form of time-series are obtained.

Figures 4.1, 4.3, 4.2, and 4.4 plot the SSM heatmaps for all four books inferred by the eight methods. The SSMs may look somewhat uninformative at first glance but a closer look reveals a subtle checkered pattern for each corpus. The darker colors in the SSMs indicate low similarity or relatedness. Brighter colors indicate a comparatively higher degree of similarity. The diagonal is a bright line because the cosine similarity between the sentence and itself is 1. Bright squares seen along the diagonal represent sequences of semantically coherent sentences, i.e., semantic segments in the document.

For each book, the semantic structure inferred by the methods show distinct similarities but not a perfect match. Although their numeric values are different, the methods agree and disagree on the broad pattern of sentence similarity As discussed earlier, this indicates that all the methods are capturing the actual intrinsic meaning of the text to a large degree.

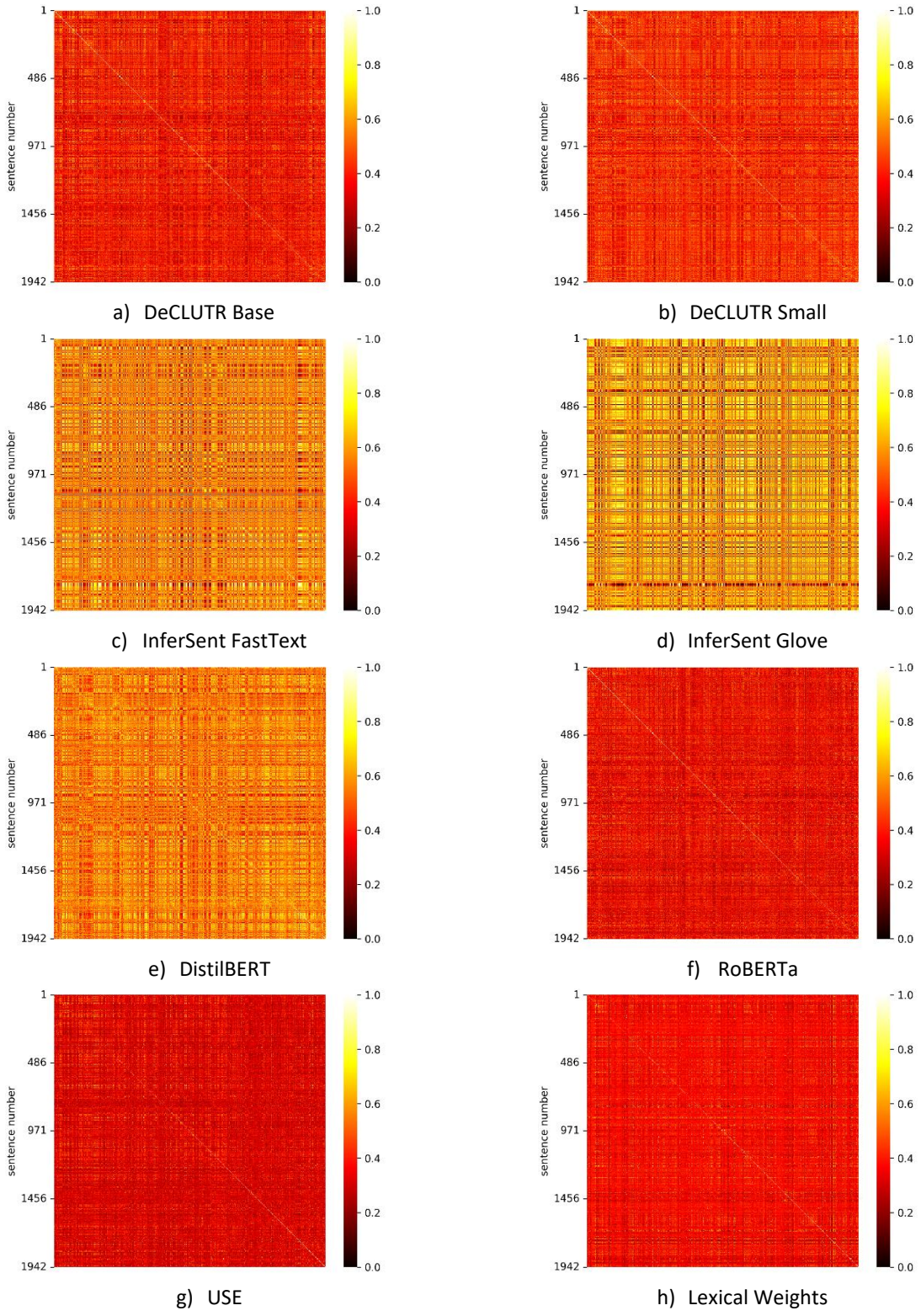


Figure 4.1: Normalized SSMs for *A Christmas Carol*

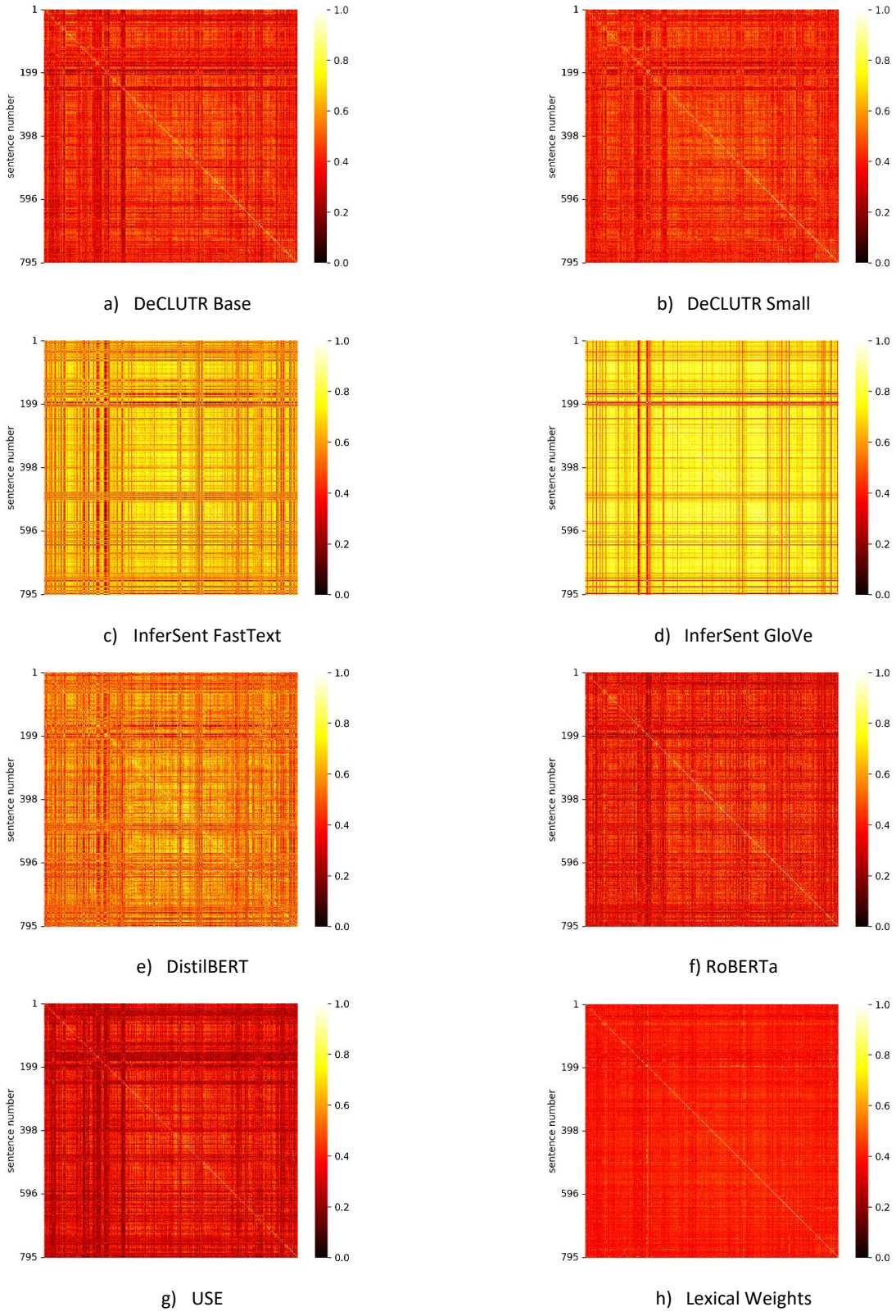


Figure 4.2: Normalized SSMs for *Metamorphosis*

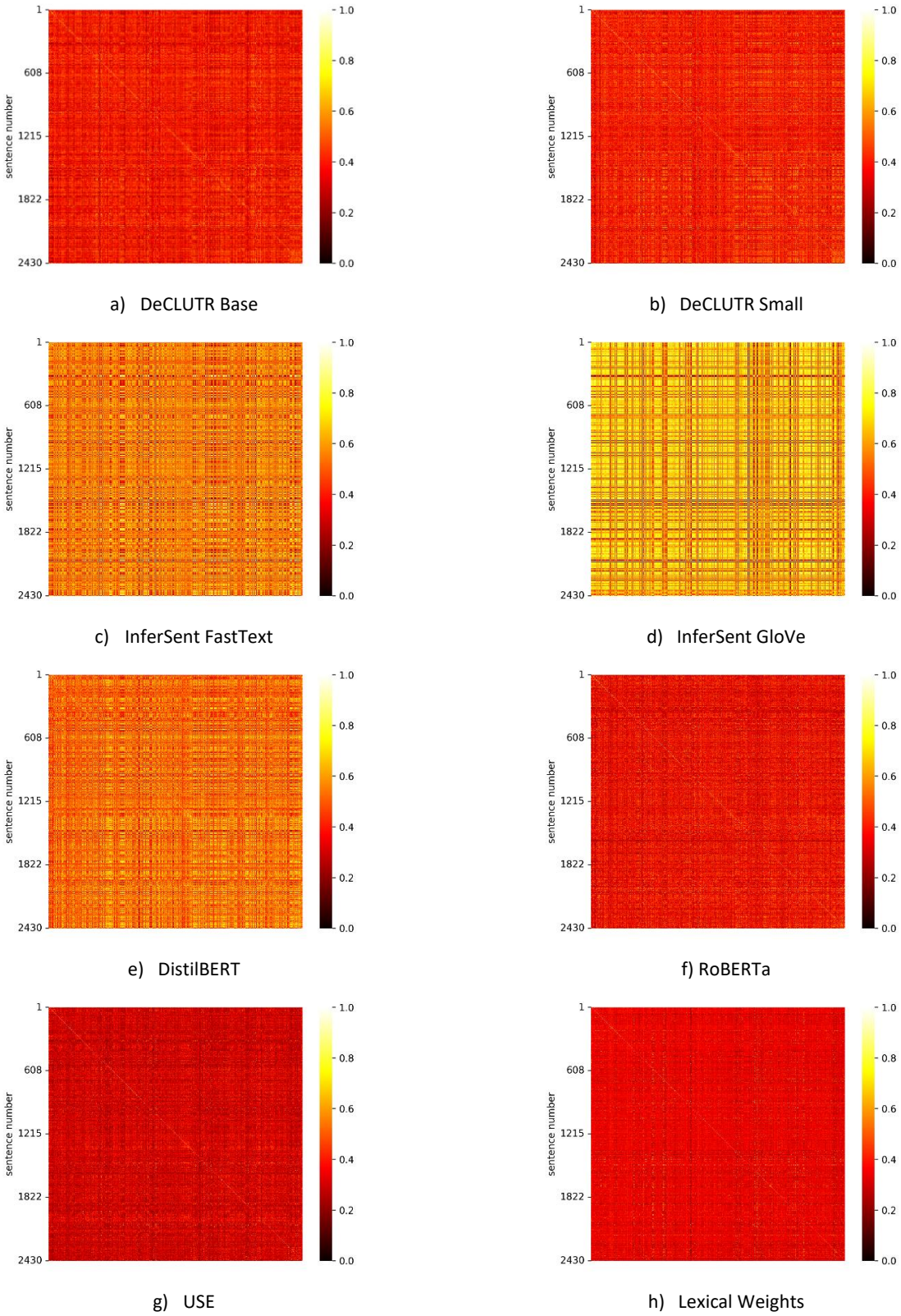


Figure 4.3: Normalized SSMs for *Heart of Darkness*

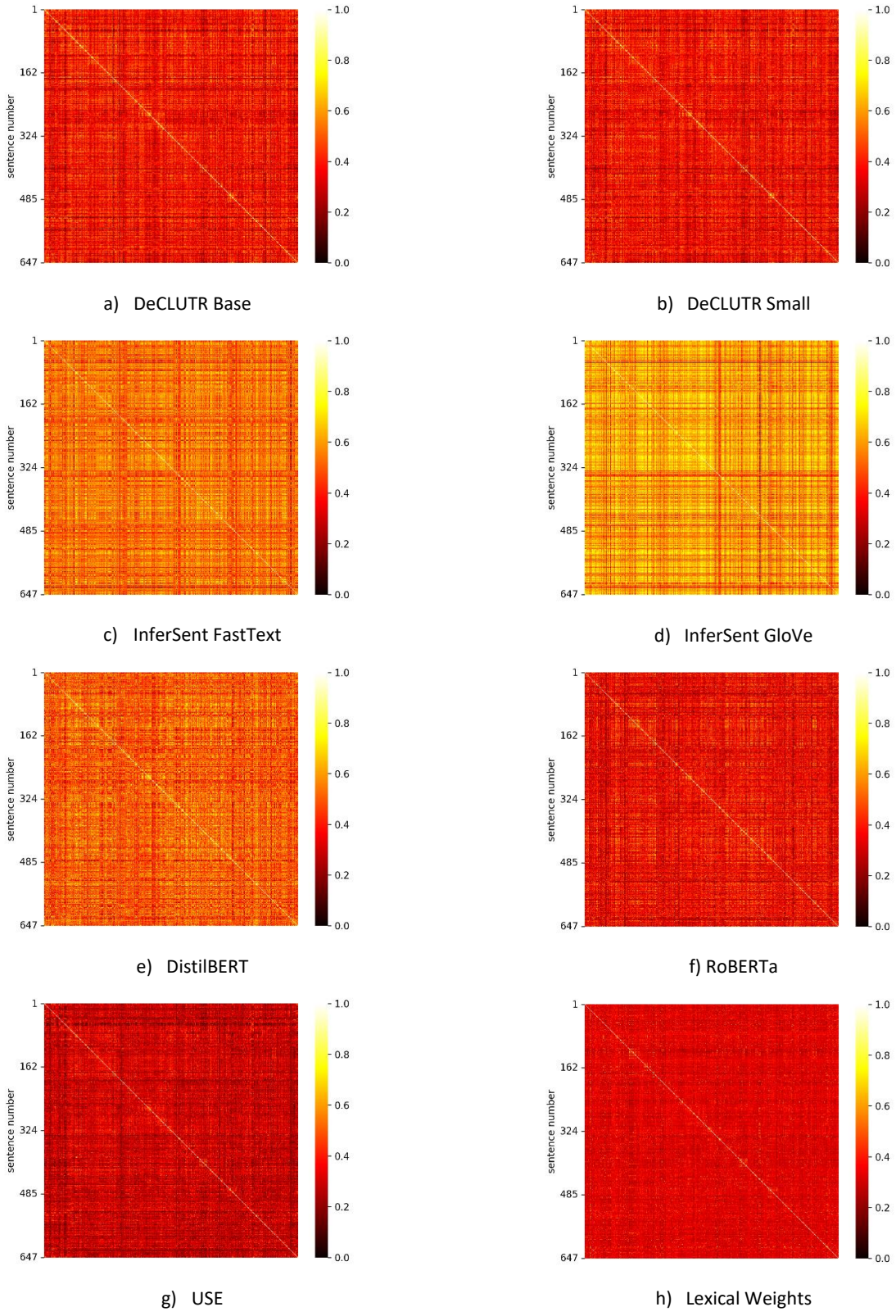


Figure 4.4: Normalized SSMs for *The Prophet*

It is noticeable that the SSMs produced by the methods for each book have different dominant colors after normalization. This is because each method produces its own distribution of sentence similarities but all the heatmaps use the same colormap. The histograms of sentence similarity distributions are shown in Figures 4.5 - 4.8.

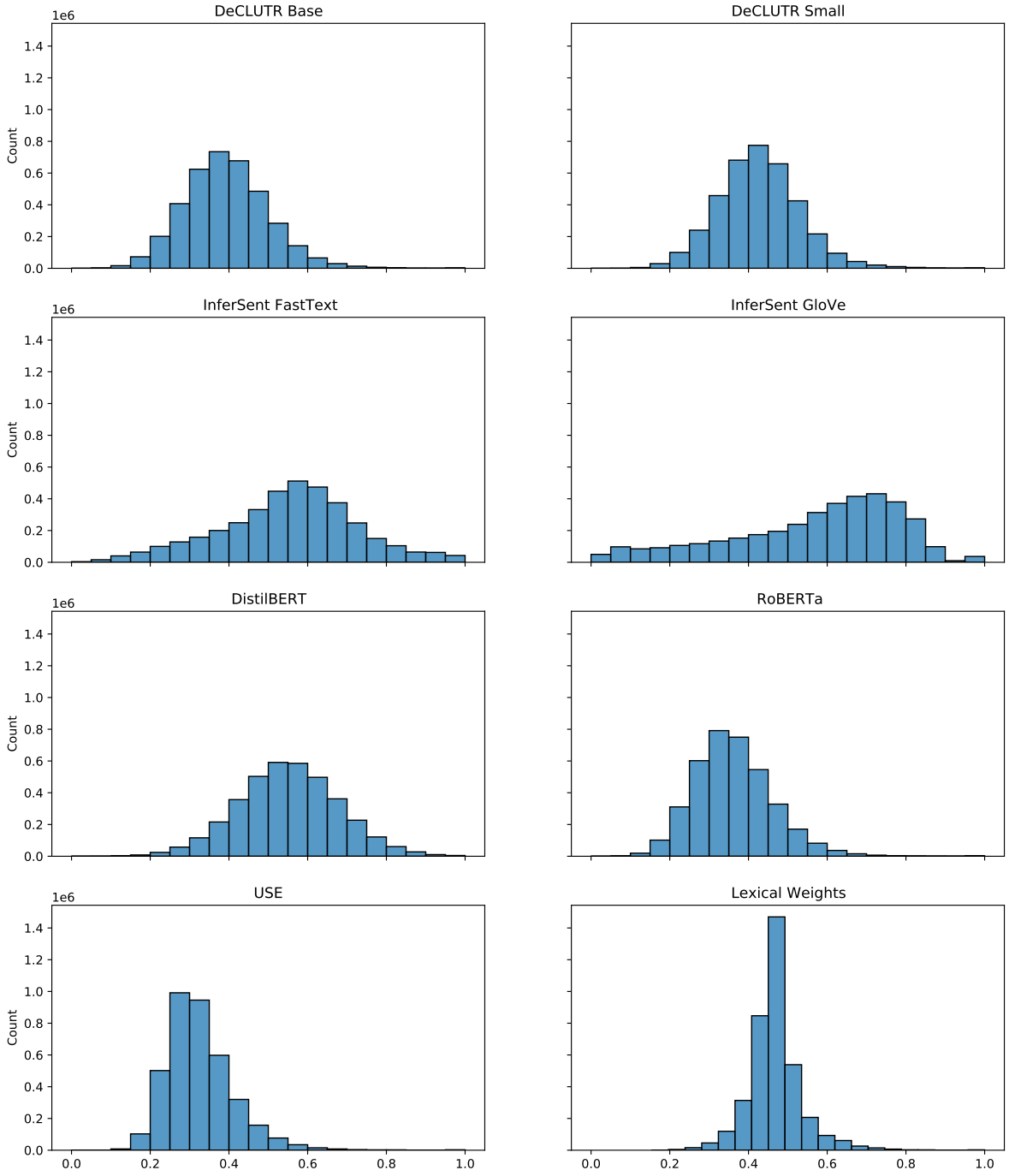


Figure 4.5: Histograms of sentence similarity distributions for *A Christmas Carol*

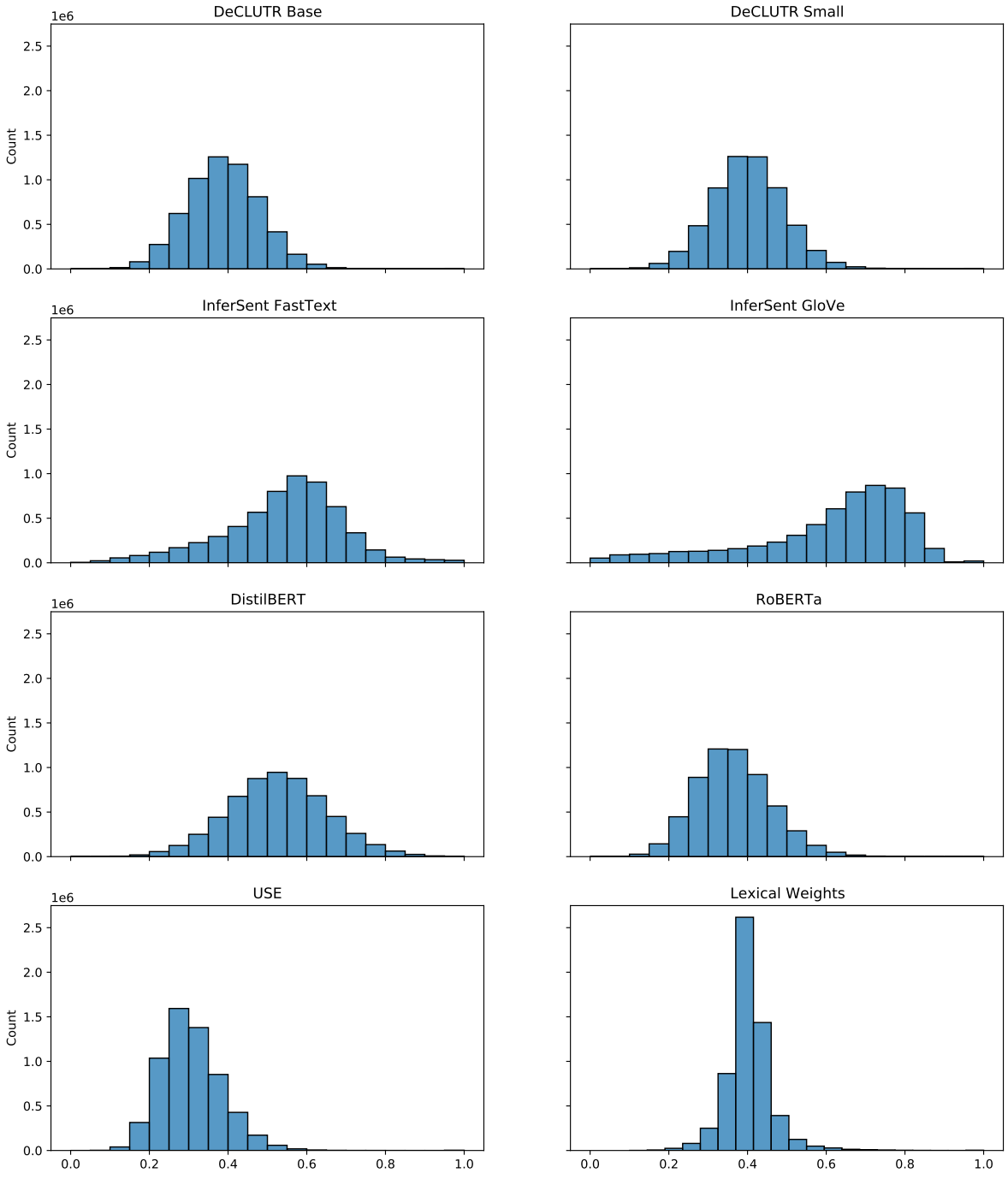


Figure 4.6: Histograms of sentence similarity distributions for *Heart of Darkness*

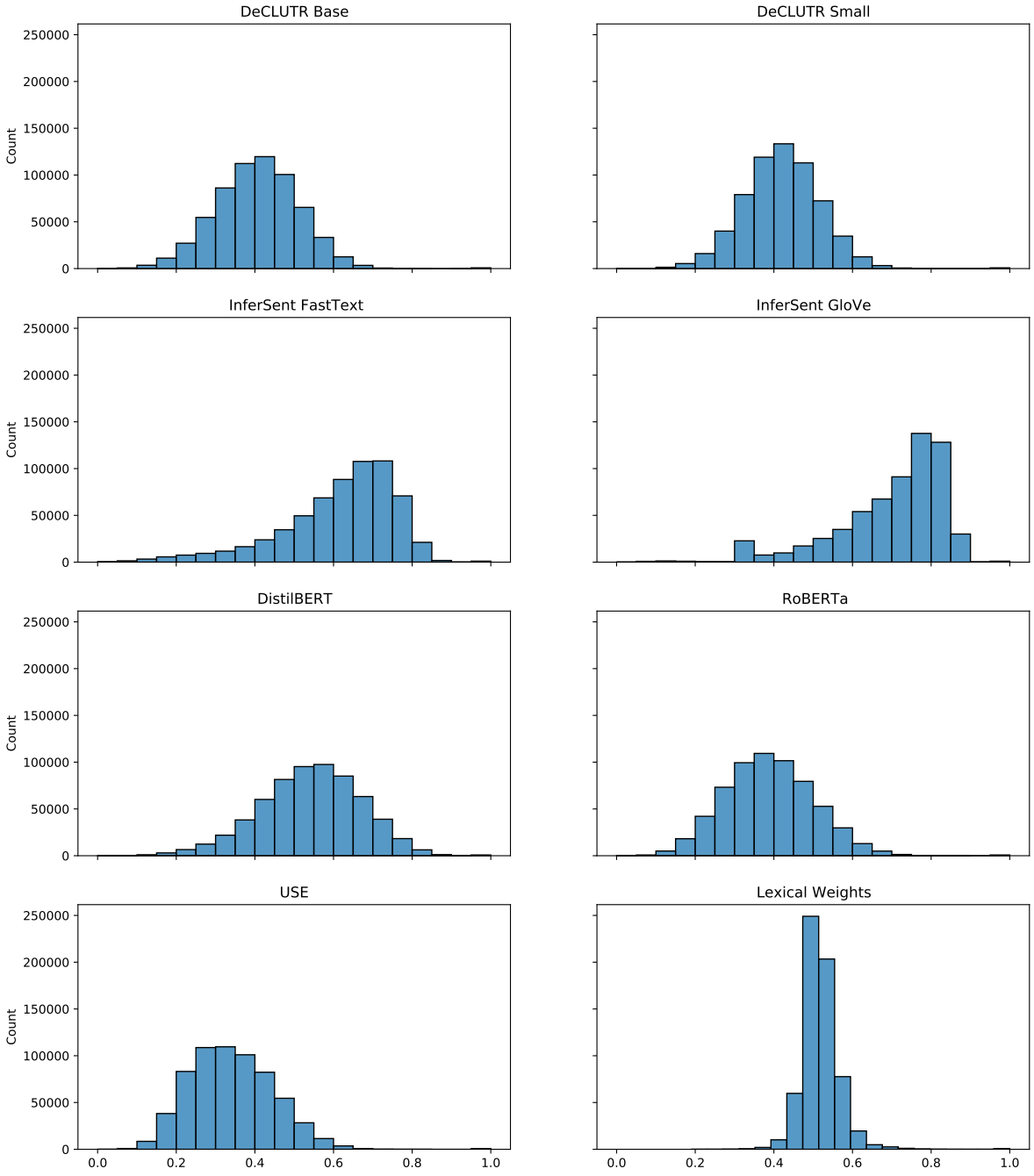


Figure 4.7: Histograms of sentence similarity distributions for *Metamorphosis*

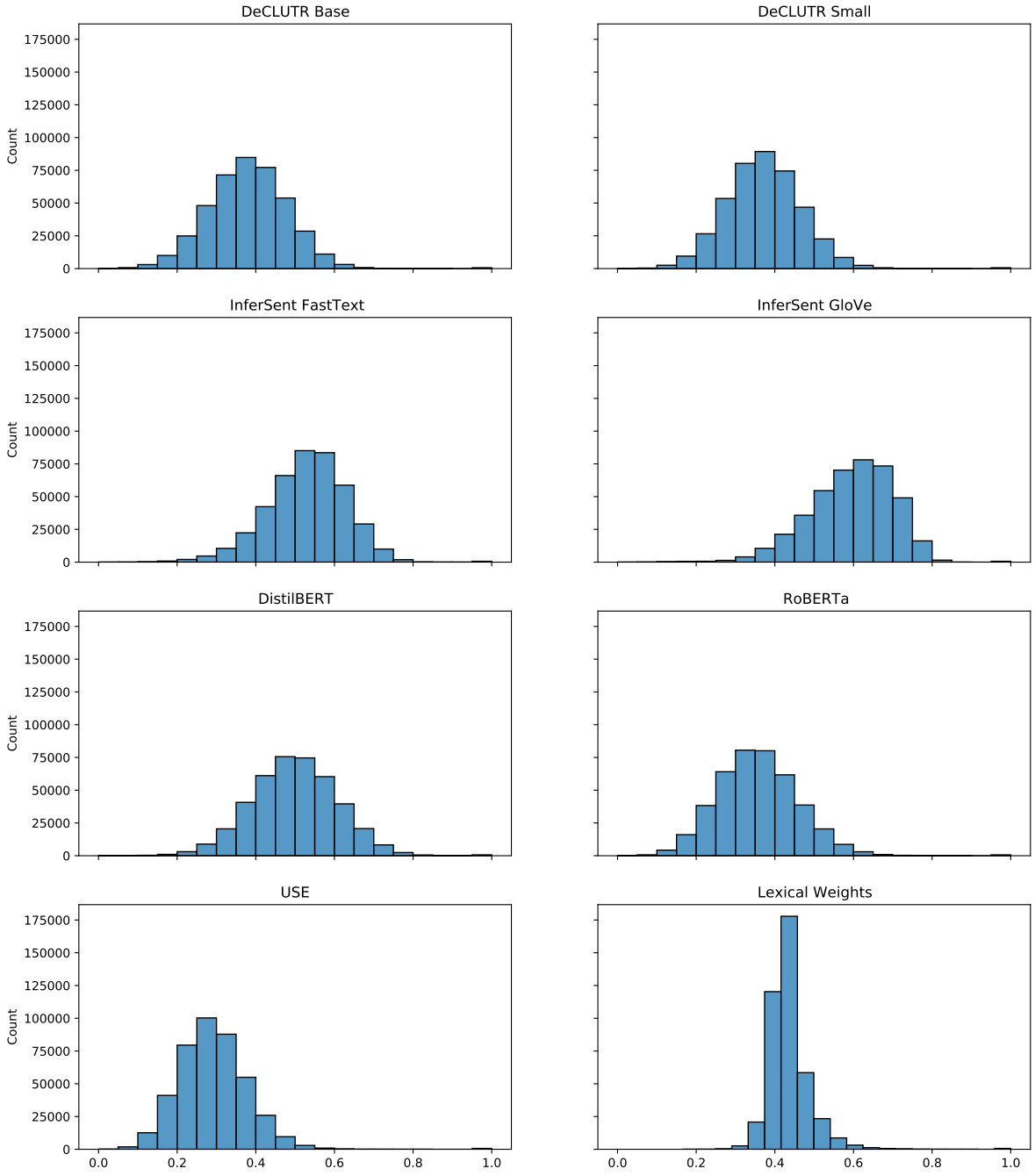


Figure 4.8: Histograms of sentence similarity distributions for *The Prophet*

4.1.2 Threshold Plots for the Four Literary Texts

SSMs are able to capture the overall semantic structure of the corpus but it is difficult to see the structure very clearly. To do that, it is useful to turn the SSM heatmaps into

binary maps that only show whether sentences are below or above some level of similarity. A systematic way of doing this is to threshold each SSM by a specific z-score, e.g., 1 standard deviation above the mean (z-score = 1). This blacks out all sentence similarity values below the threshold, revealing the global skeletal structure of high similarity in the document.

Figures 4.9, 4.10, 4.11 and 4.12 show normalized SSMs for all books thresholded at their respective mean values (z-score = 0). The black regions in the figures are regions of below average similarity and the white regions those of above average similarity. The higher the z-score threshold, the more skeletal the map gets by picking up patterns of greater similarity.

These thresholded maps allow a quicker visual evaluation of the semantic structure being inferred by each method, as well as the gross similarities and differences in the structures inferred by various models. For example, Figure 4.9 (c) shows the presence of a region of very variable similarity between sentences 900 and 1000 picked up by DistilBERT. A less definitive pickup of that feature can be seen in (c) InferSent FastText, (f) RoBERTa and (g) USE. The other methods do not seem to pick up this feature visibly. In Figure 4.11, in contrast, all eight methods – even Lexical Weights – pick up a similar banding structure. This is also the case – albeit to a lesser degree – in Figures 4.10 and 4.12.

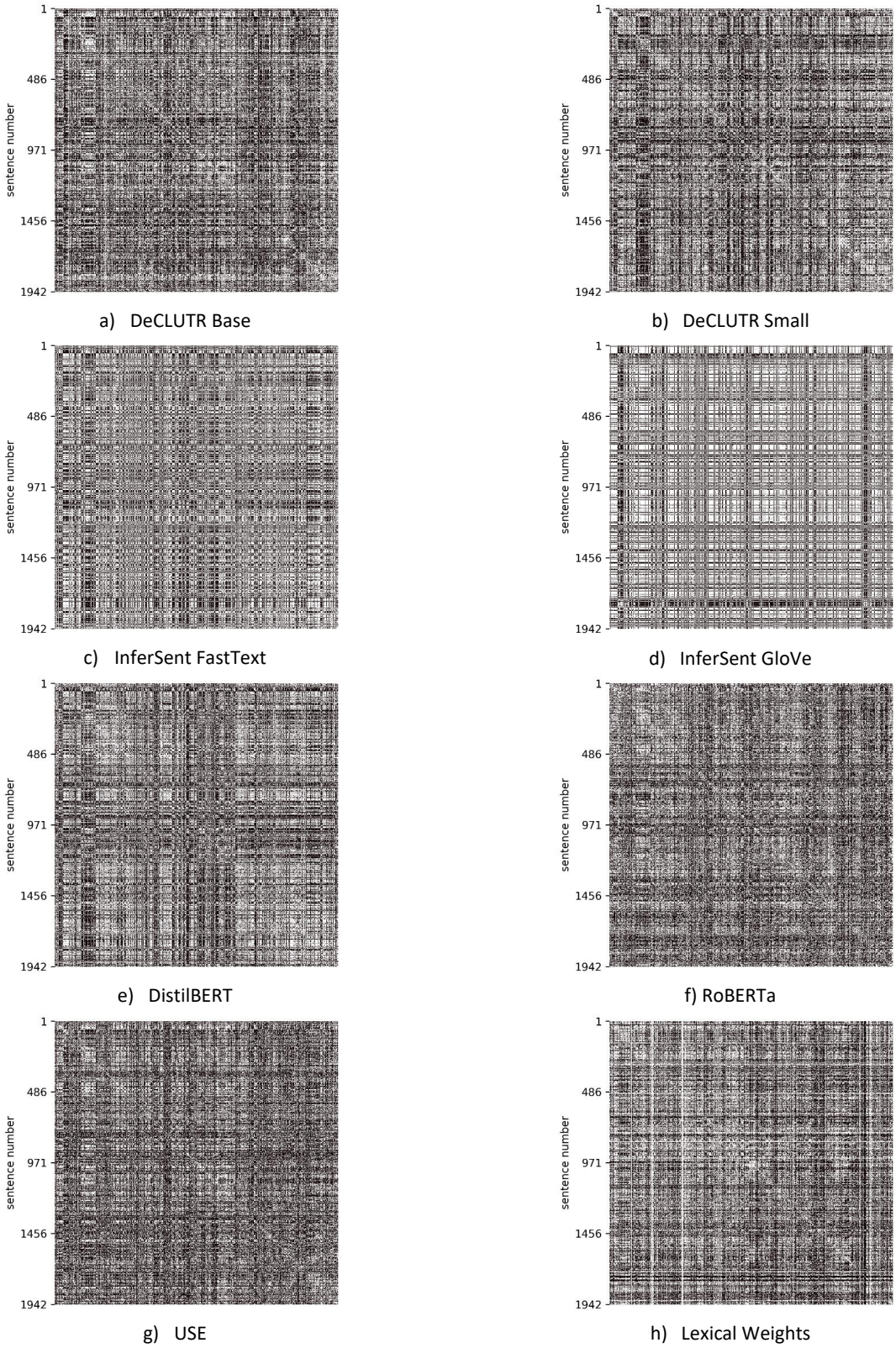


Figure 4.9: Threshold SSMs for *A Christmas Carol*

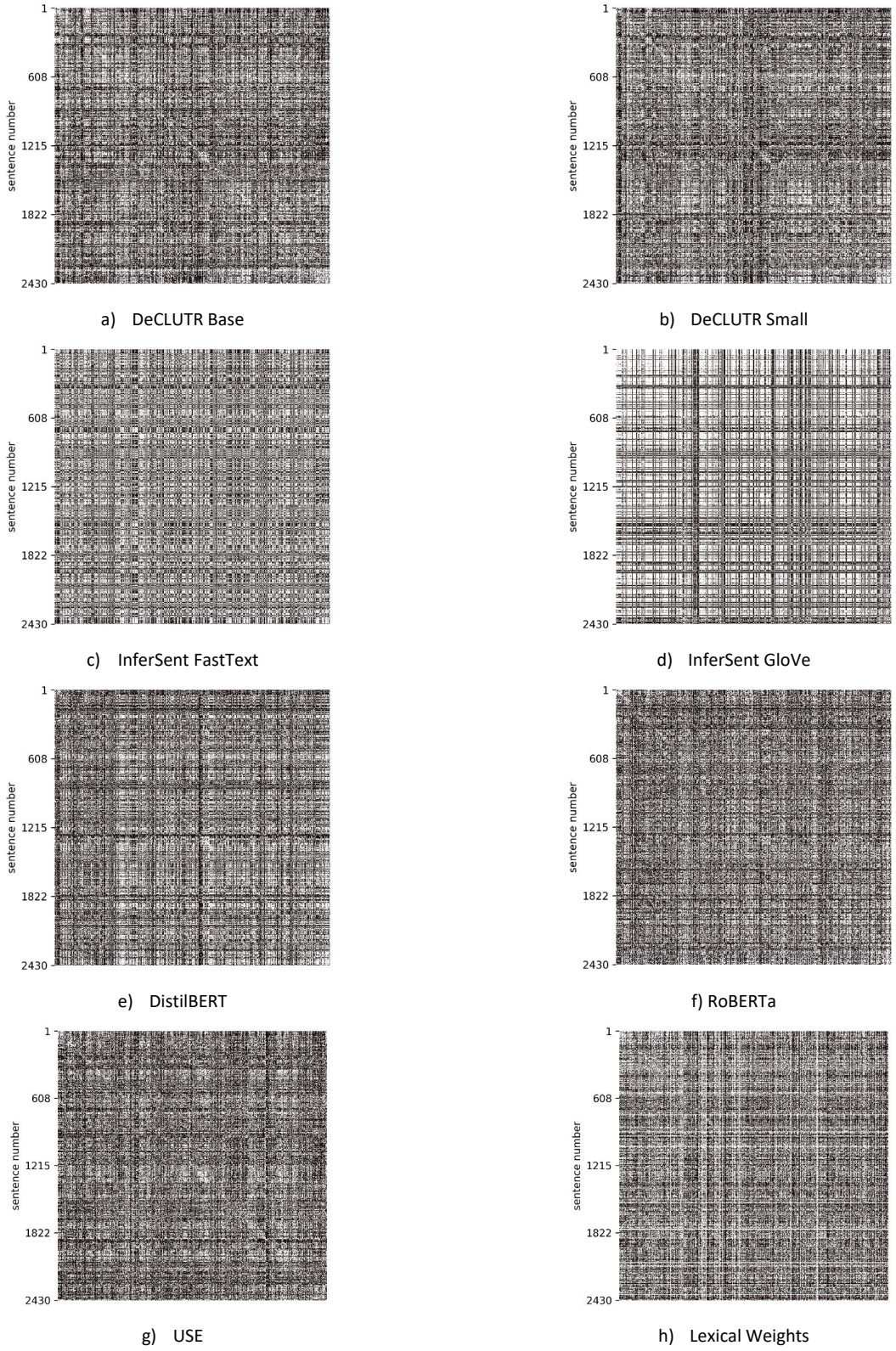


Figure 4.10: Threshold SSMs for *Heart of Darkness*

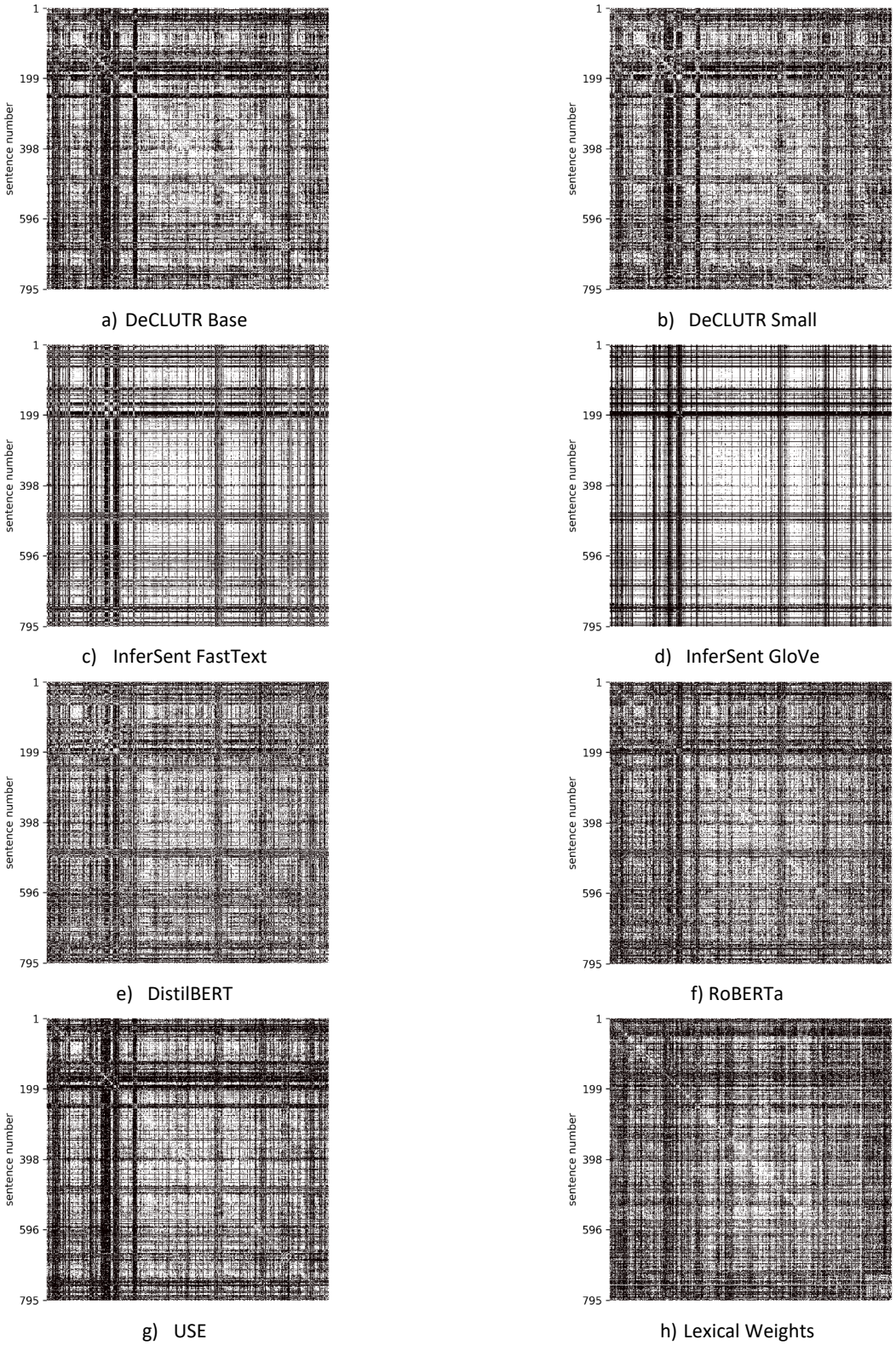


Figure 4.11: Threshold SSMs for *Metamorphosis*

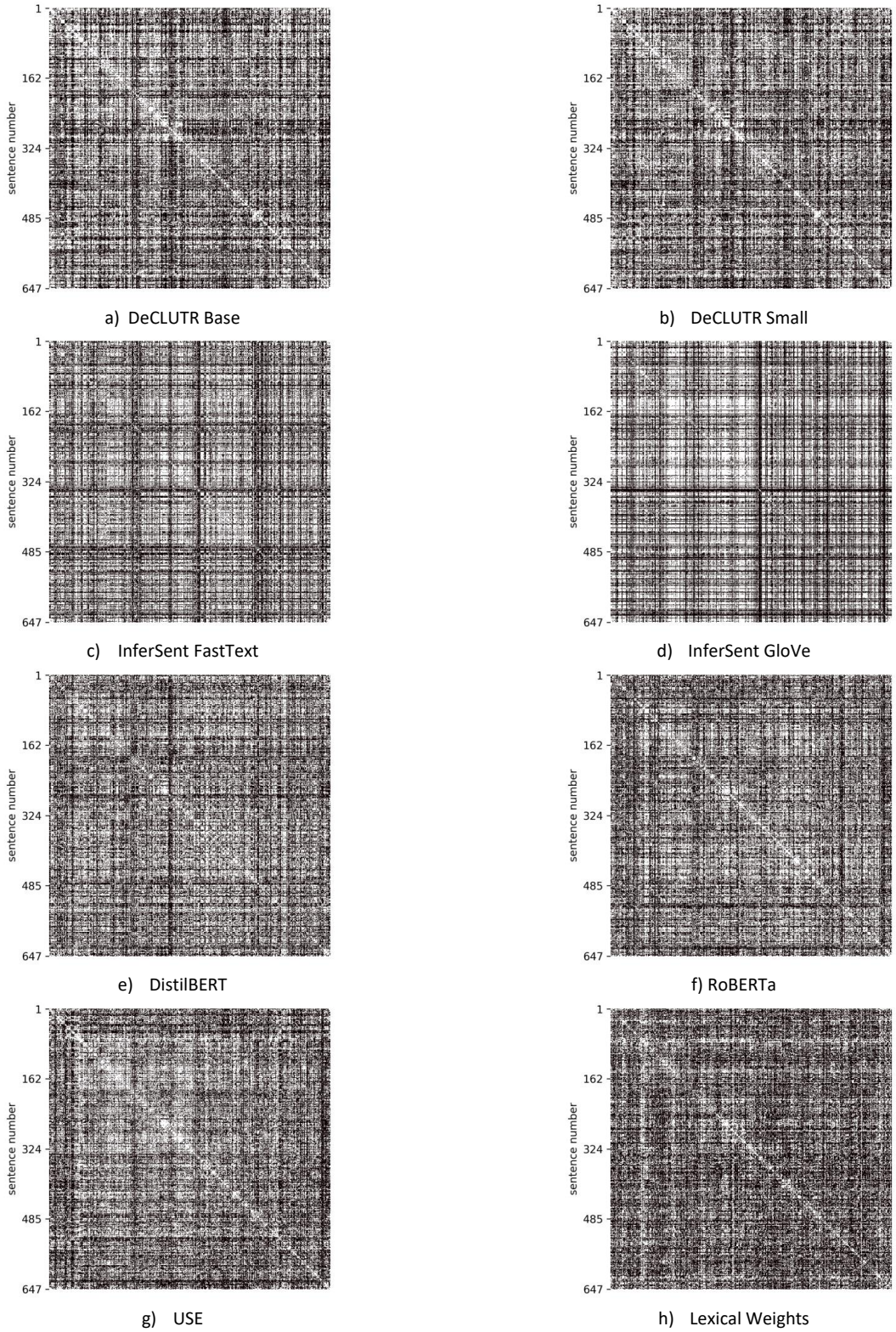


Figure 4.12: Threshold SSMs for *The Prophet*

4.1.3 SSMs and Threshold Plots for Translations

The SSMs for the translations follow patterns to those of the novels. These SSMs are huge for certain translations. For instance, the translation of the Iliad by Alexander Pope contains 5334 sentences, so its SSM would be of the size $5334 \times 5,334$. Visualizing that SSM is possible, but important details and other salient features would not be easily visible. These same principles apply to the threshold plots. As a result, these plots are not included in this thesis.

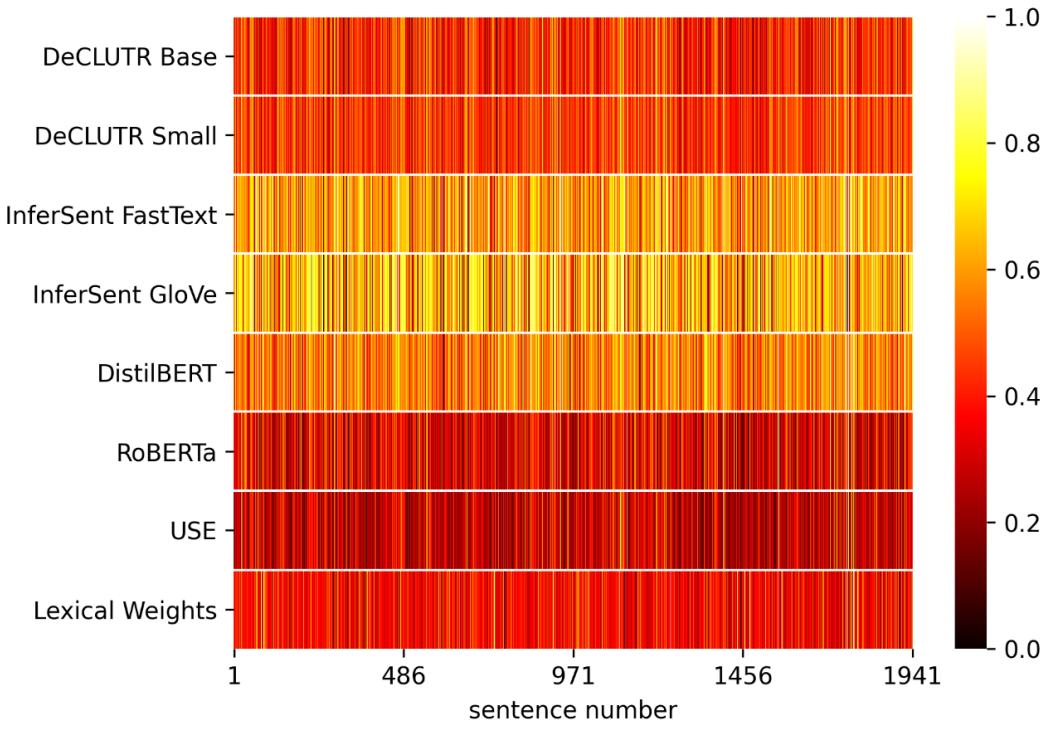
4.1.4 Time-series for All Books

The sentence similarity time-series are plotted as strip heatmaps using color to code similarity value. This makes it possible to compare across methods more easily than with SSMs by placing all the strips one above the other.

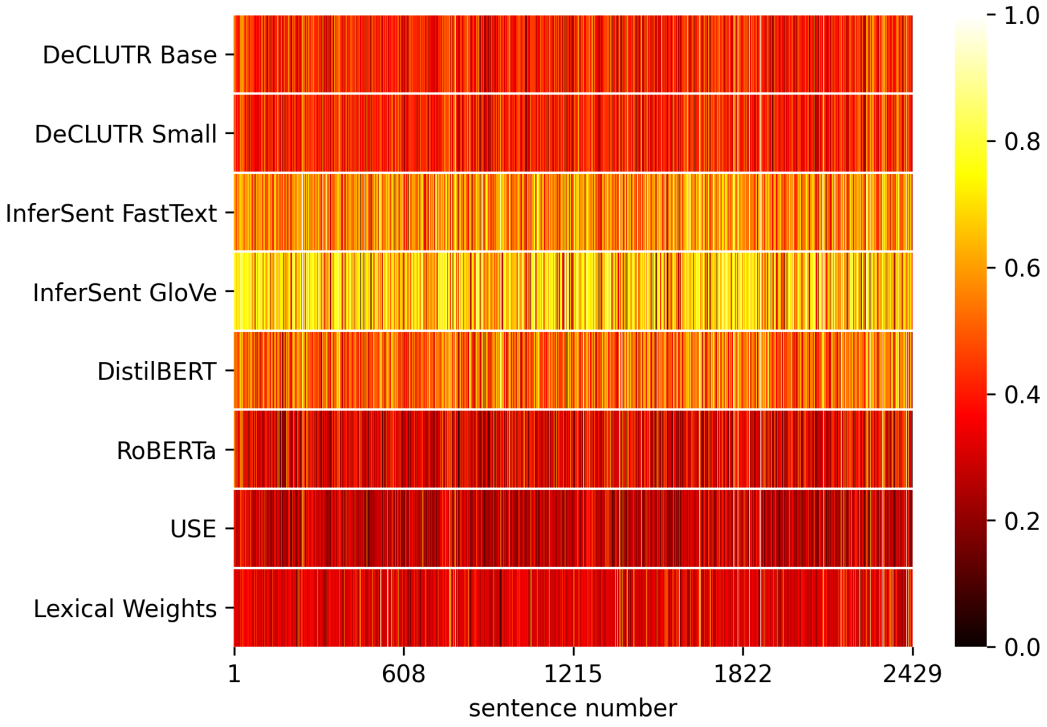
Figures 4.13 and 4.14 show the time-series plots for the four literary books, and Figures 4.15 - 4.20 for the translations. The following observations can be made individually for each case:

- Both time-series for DeCLUTR Base and DeCLUTR Small are extremely similar, which is validated by their correlation coefficients in 4.23
- The time-series for InferSent FastText and InferSent GloVe look quite similar, but also show notable differences.
- The time-series for DistilBERT and RoBERTa are actually quite similar, though they look different because the distribution of similarity values is very different in the two cases.
- The USE has significant resemblance with all embedding based methods, but somewhat lower with DistilBERT and the two InferSent methods.

- The Lexical Weights time-series does have some resemblance with the others in terms of gross features, but much than that between the other methods.

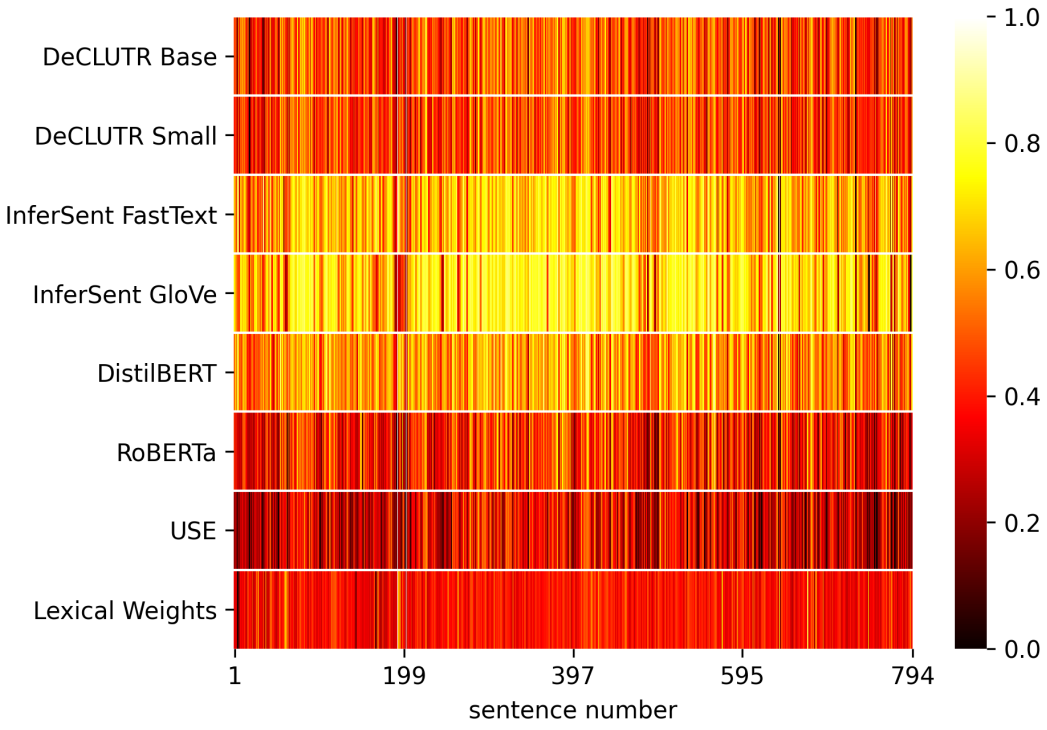


a) A Christmas Carol

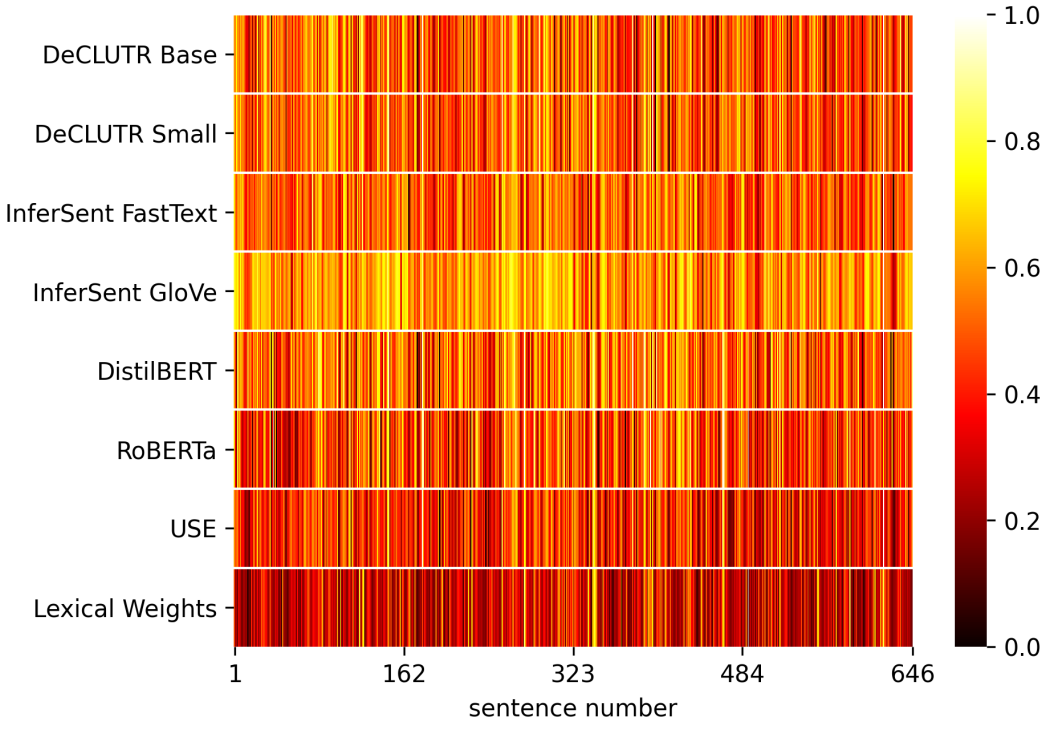


b) Heart Of Darkness

Figure 4.13: time-series for *A Christmas Carol* and *Heart of Darkness*

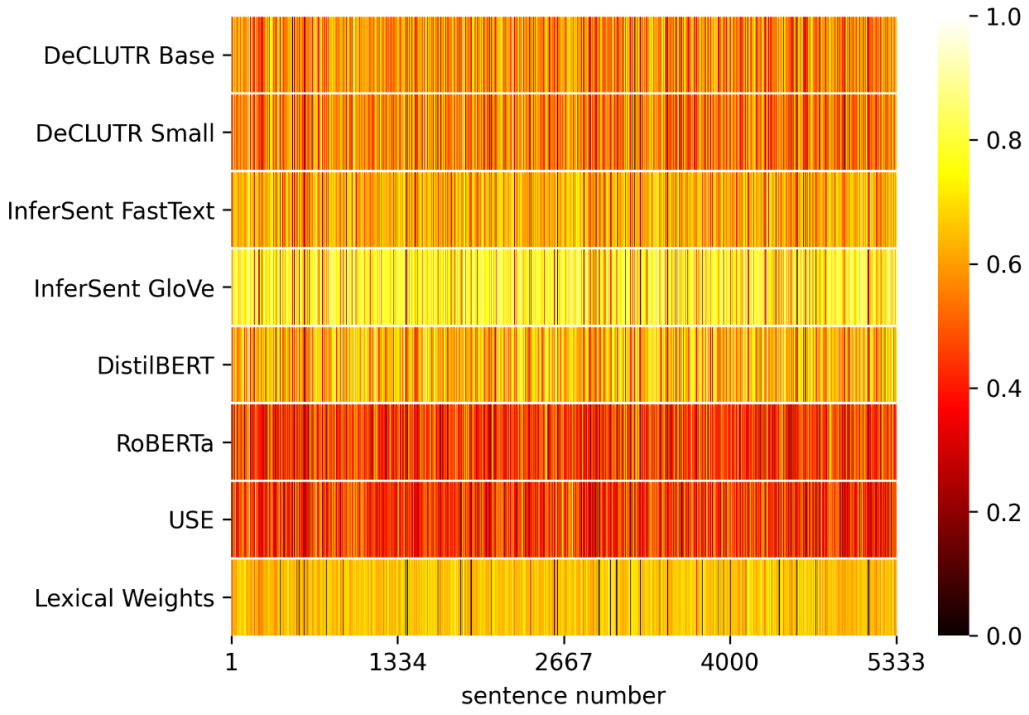


a) Metamorphisis

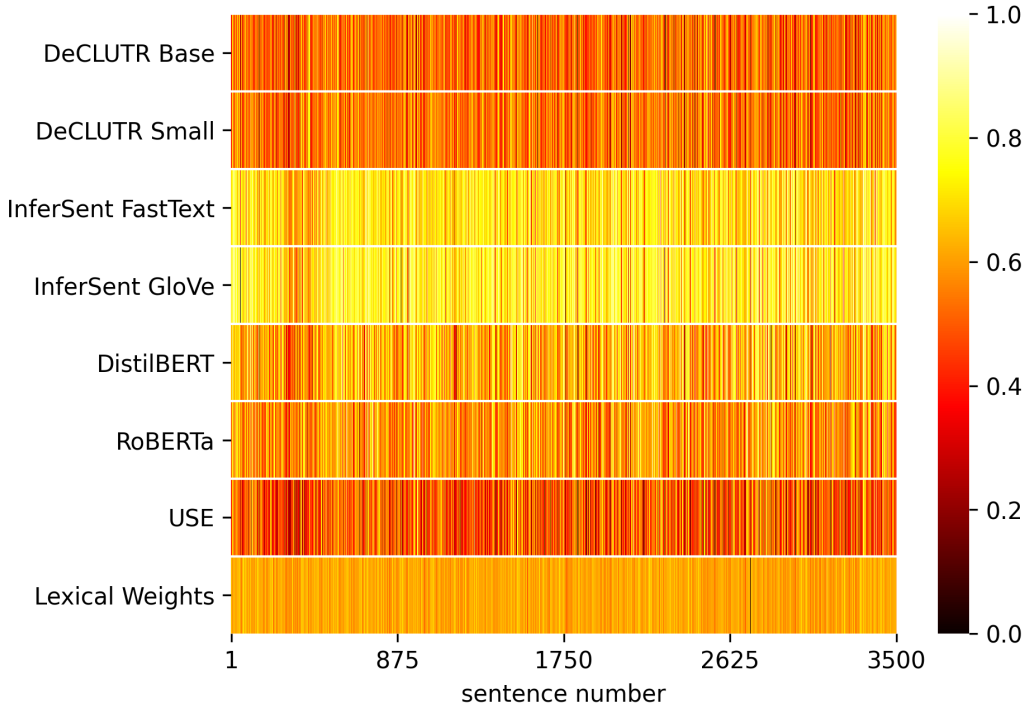


b) The Prophet

Figure 4.14: time-series for *Metamorphosis* and *The Prophet*

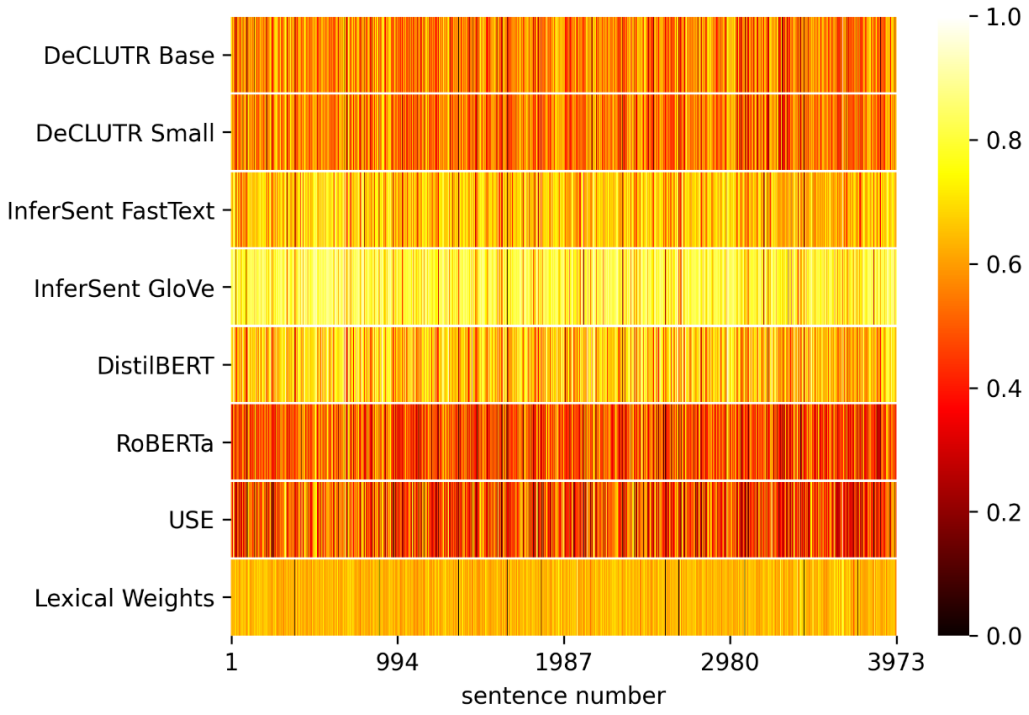


a) Alexander Pope

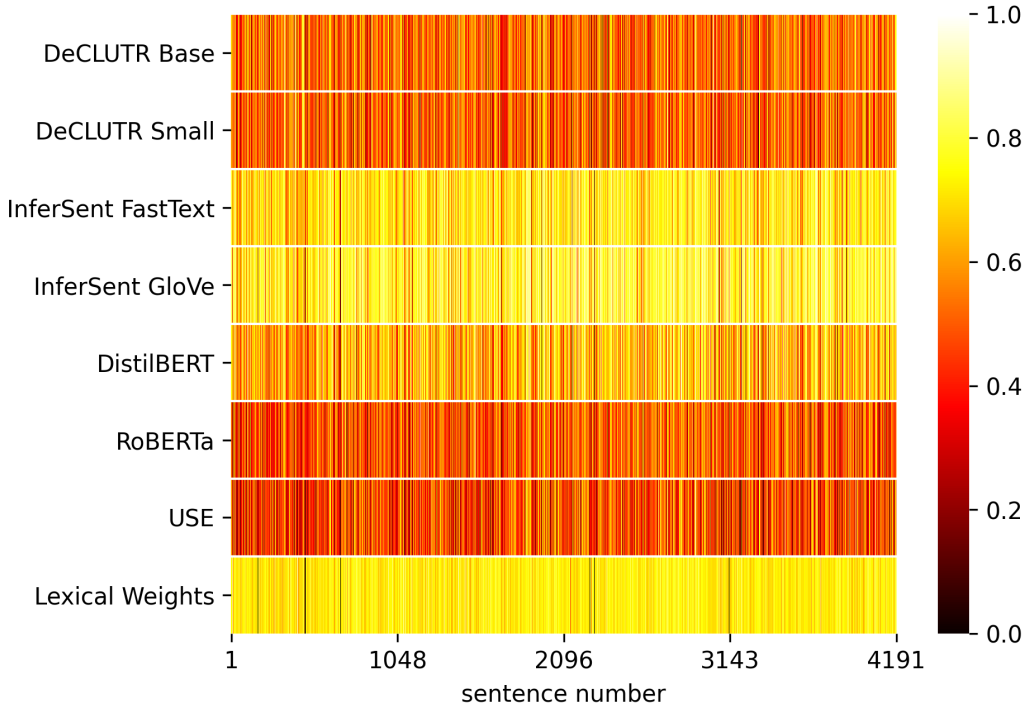


b) Lang et al

Figure 4.15: time-series plots for *The Iliad* by Alexander Pope and Lang et al

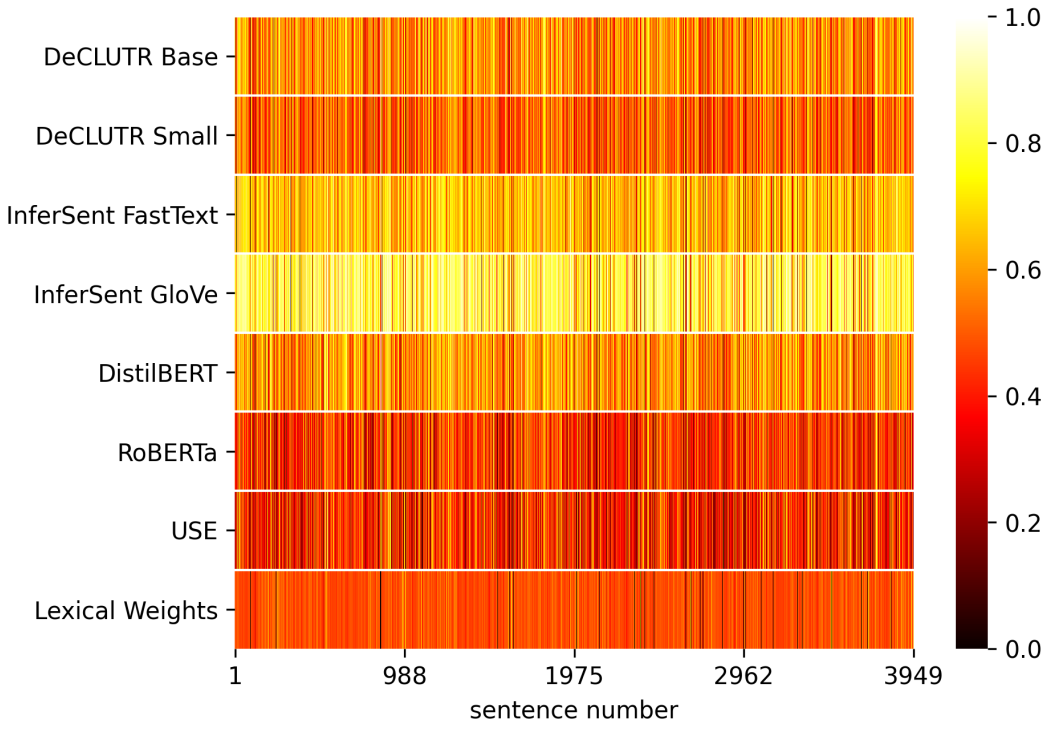


a) George Chapman

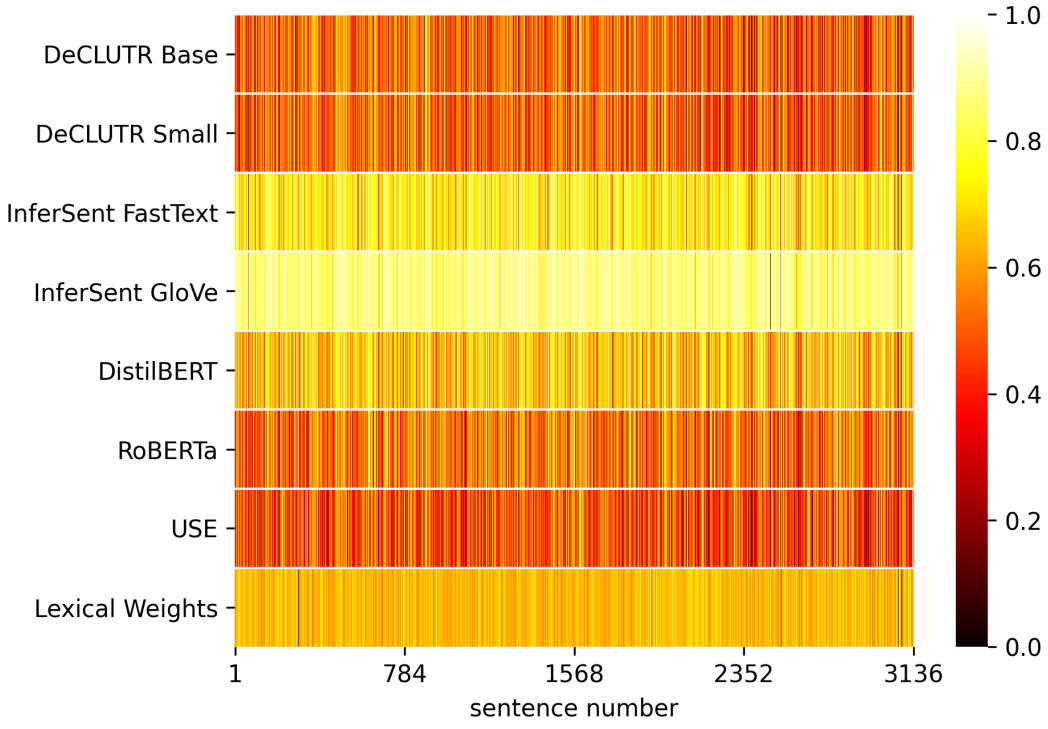


b) Samuel Butler

Figure 4.16: time-series plots for *The Iliad* by George Chapman and Samuel Butler

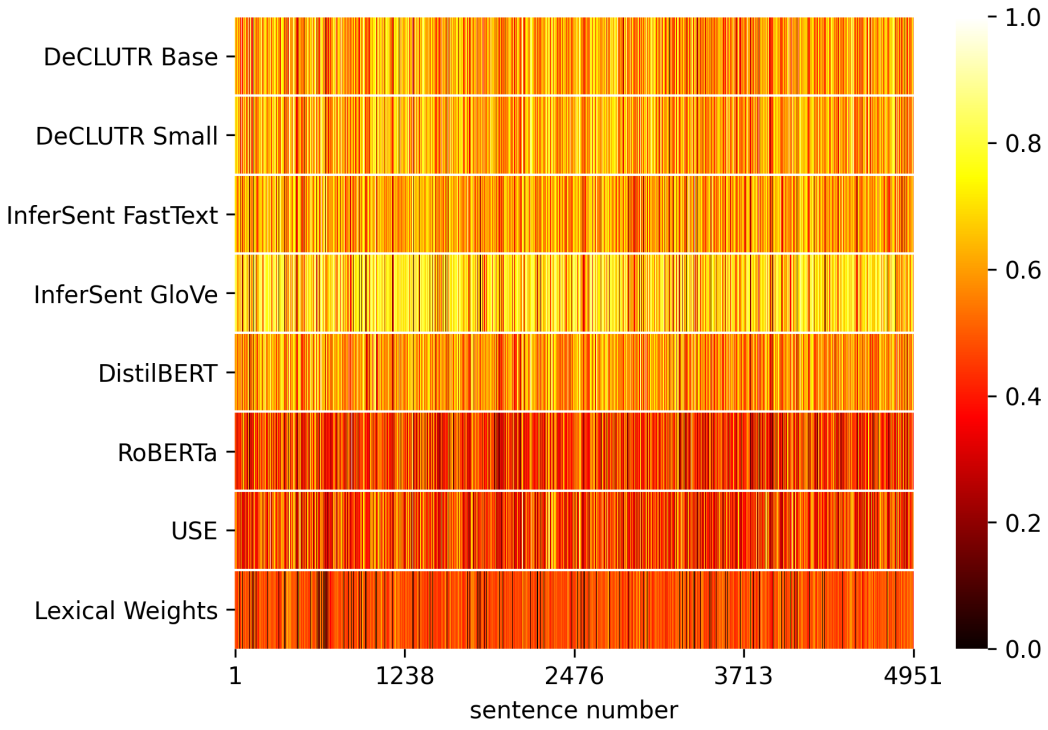


a) Alexander Pope

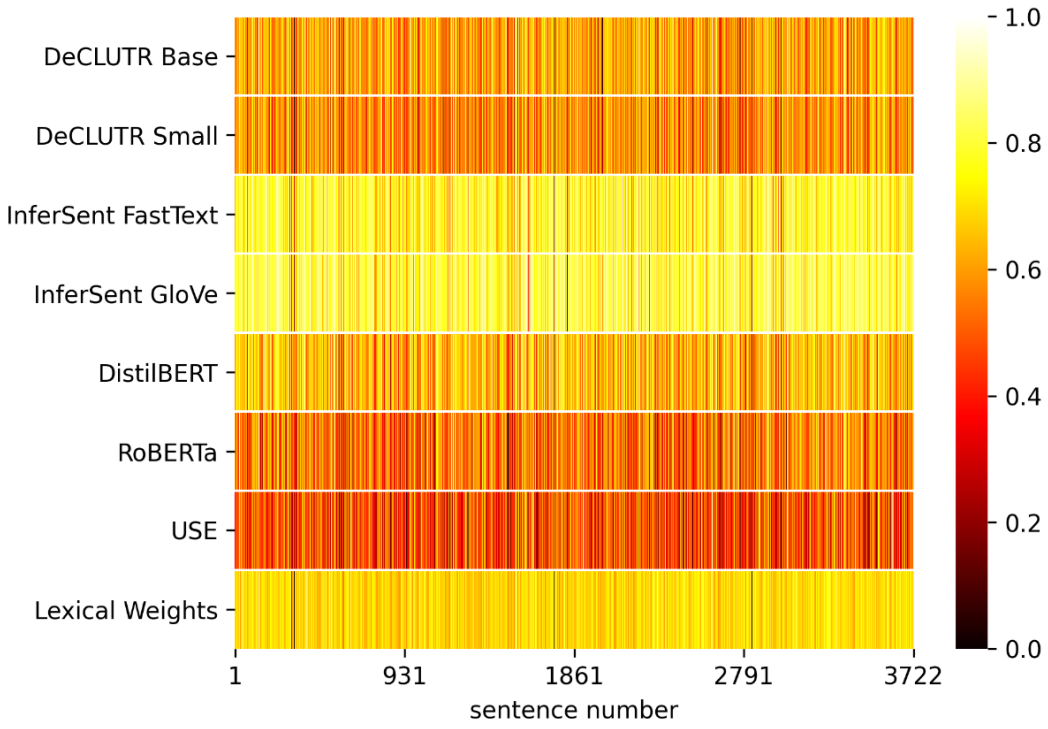


b) Samuel Butler

Figure 4.17: time-series plots for *The Odyssey* by Alexander Pope and Samuel Butler

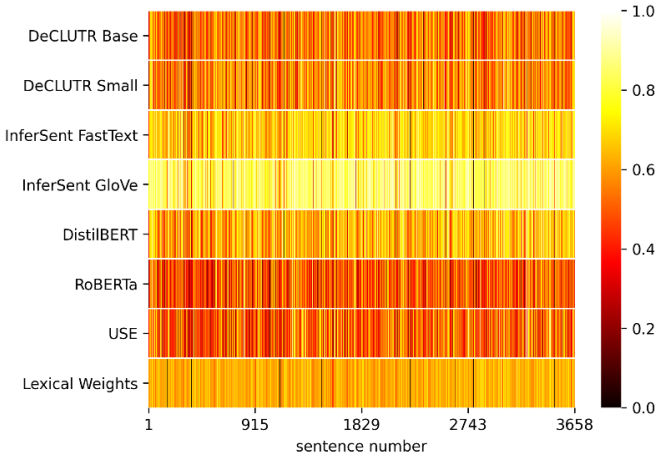


a) William Cowper

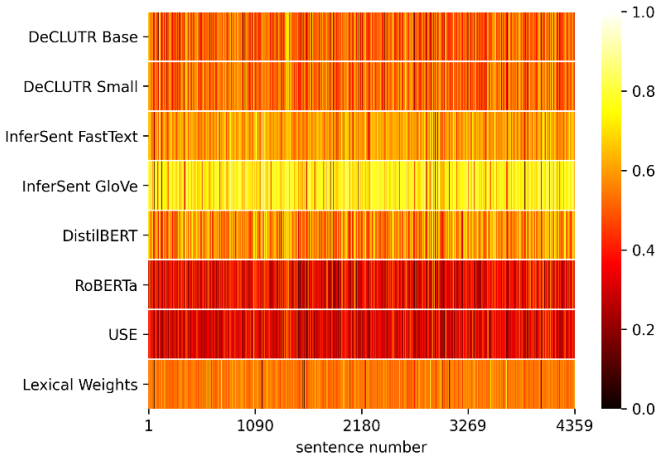


b) Butcher and Lang

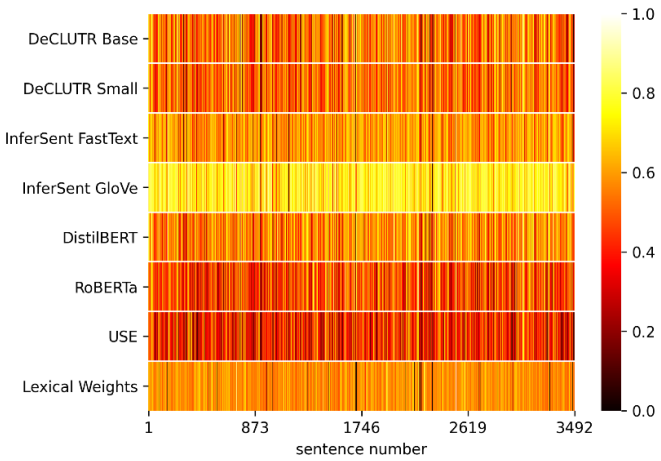
Figure 4.18: time-series plots for *The Odyssey* by William Cowper and Butcher and Lang



a) J. W. Mackail

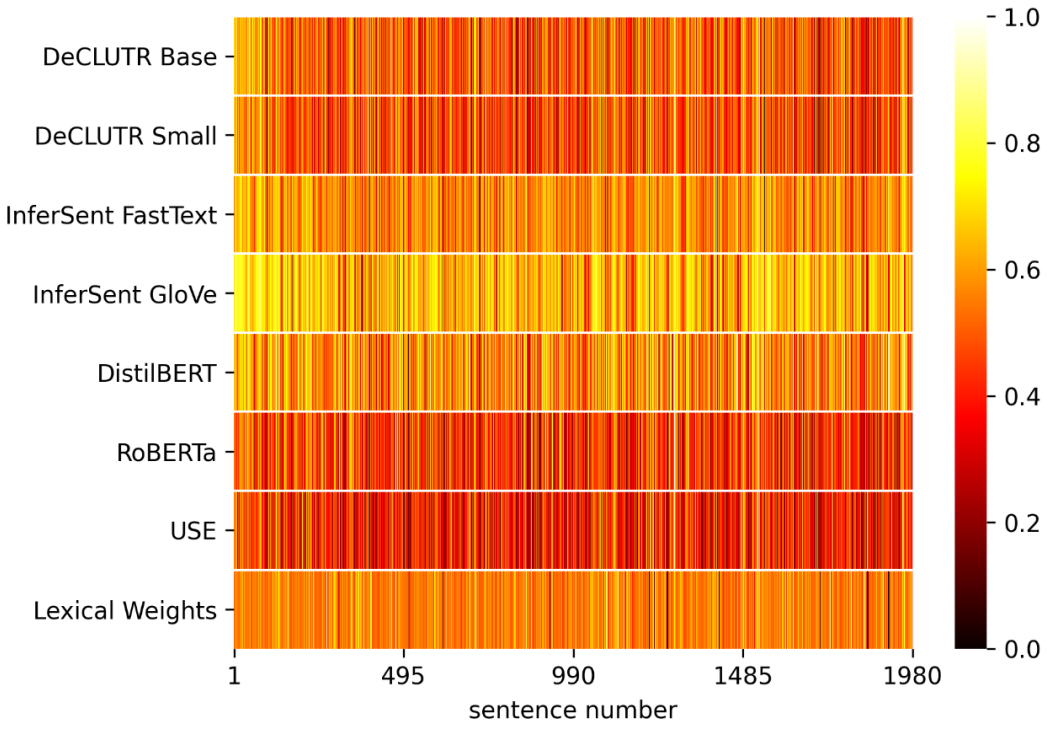


b) John Dryden

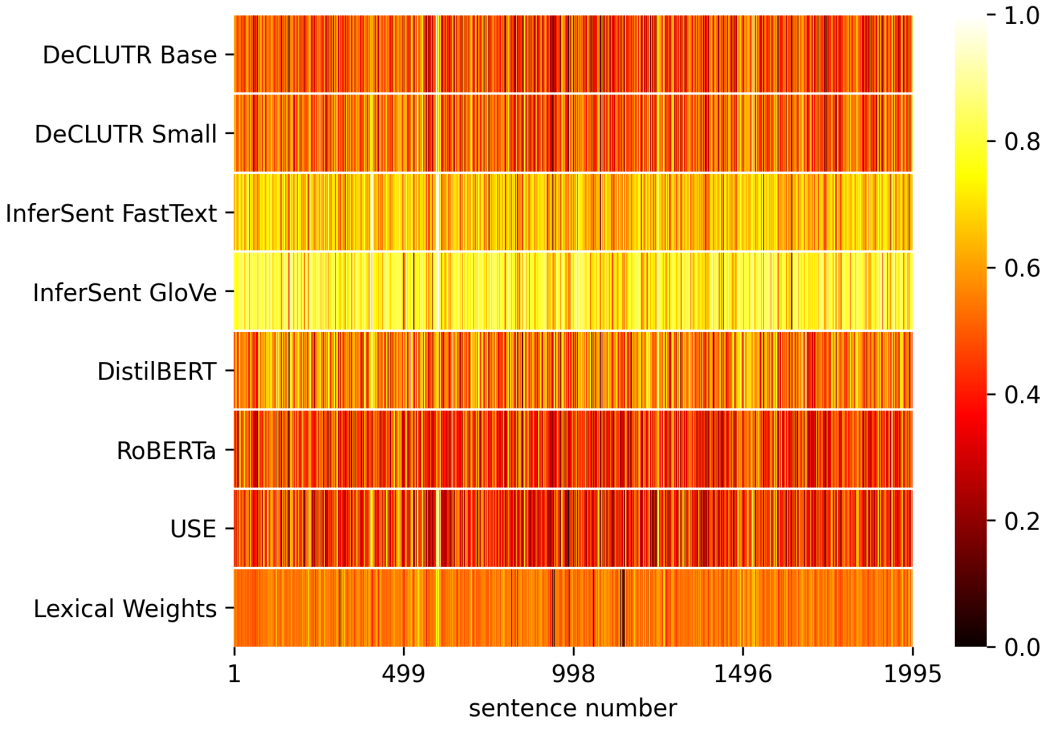


c) Rolfe Humphries

Figure 4.19: time-series plots for *The Aeneid*



a) George Chrystal



b) Meric Casaubon

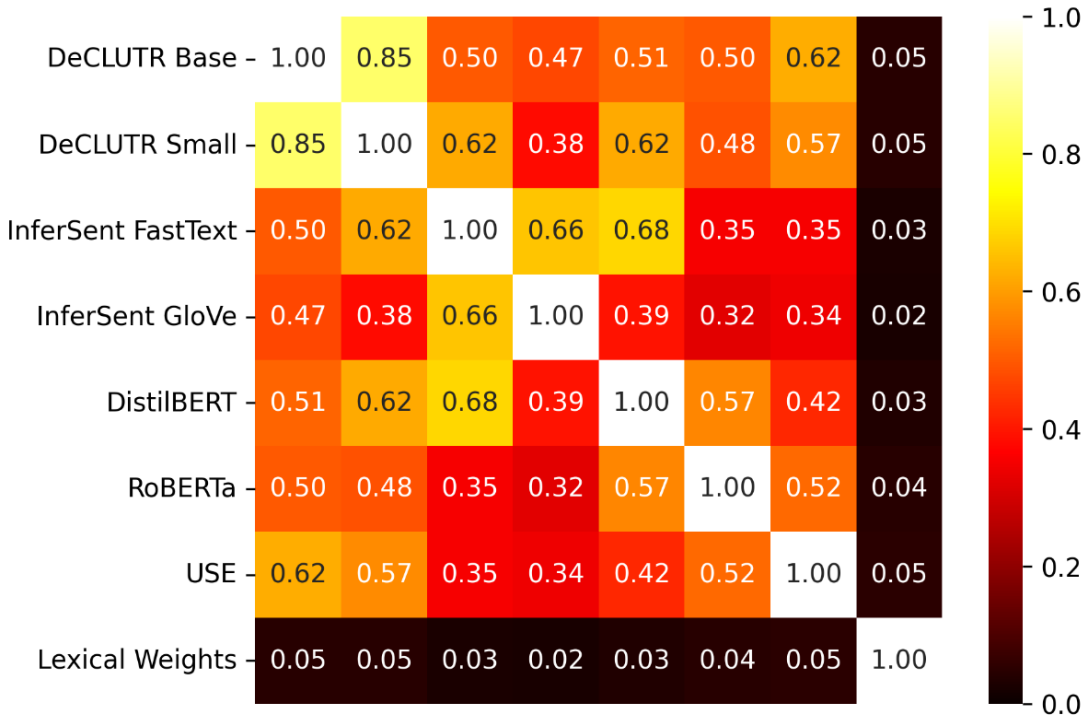
Figure 4.20: time-series plots for *The Meditations*

4.1.5 Correlation Plots for SSMs

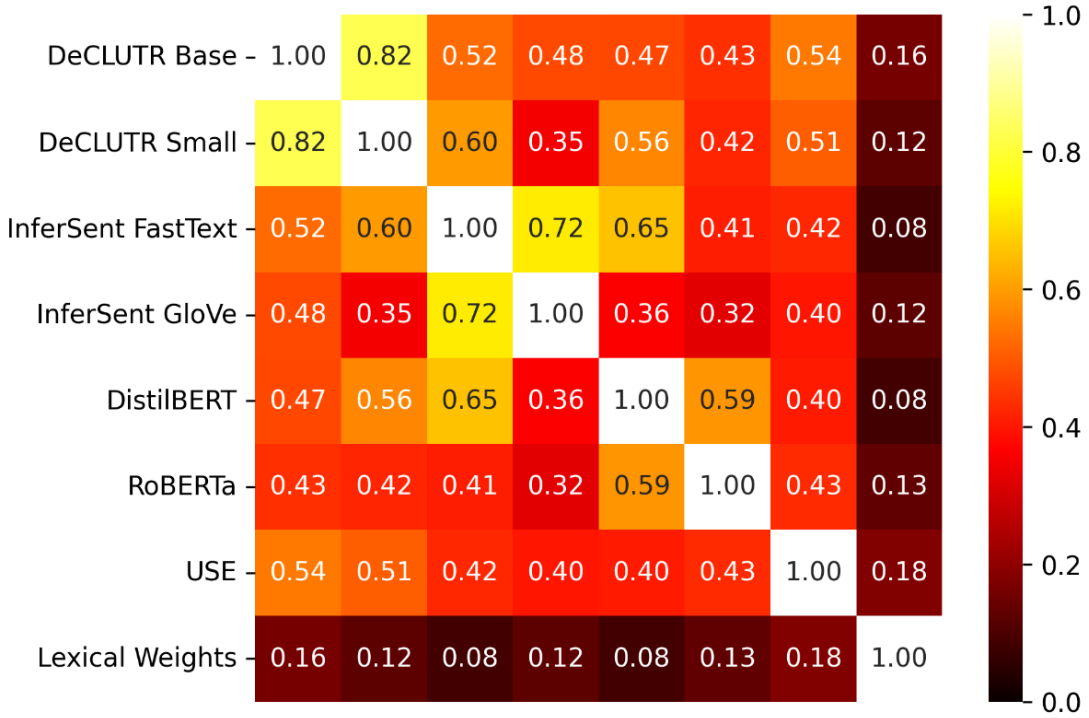
The SSMs for a given corpus look similar because of their overall structure. To get a numeric value of their similarity, Pearson correlation coefficients were calculated between the SSMs for each method on the same corpus as described in Chapter 3. Figures 4.21 and 4.22 show the SSM correlation plots for the four lterary texts. The correlation plots have been arranged so that methods with similar architectures get grouped together. It is evident all the four correlation plots in that:

- The encoders using DeCLUTR have very high correlation coefficients.
- The two encoders using InferSent have fairly high correlation coefficients.
- DistilBERT and RoBERTa have high correlation with each other.
- USE has the highest correlation with DeCLUTR – perhaps due to architectural similarities between their underlying neural networks.
- The Lexical Weights method has low correlation with all the other methods.

A very peculiar thing is noticeable in Figure 4.21 (a), where the Lexical Weights method has almost no correlation with the others. This is probably due to the dialog-oriented style of *A Christmas Carol*, with many short – even single-word – sentences in every conversation. Since PMI is based only on the co-occurrence of words *within* a sentence, short, generic sentences lead to poor semantic inference – especially after stop-word removal. InferSent uses bidirectional recurrent neural networks to keep track of the context between the current sentence and the sentences previously encountered whereas other methods use the attention mechanism in its original (or modified) for this purpose. The Lexical Weight method has significantly higher correlation with the other methods of the other three corpora.

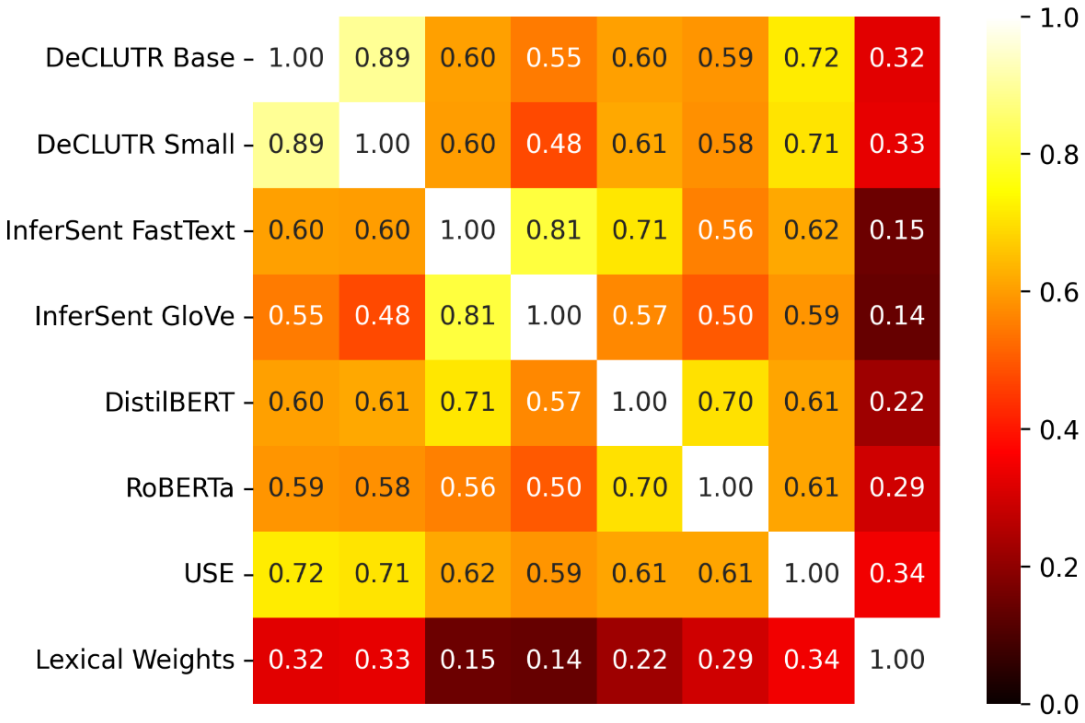


a) A Christmas Carol

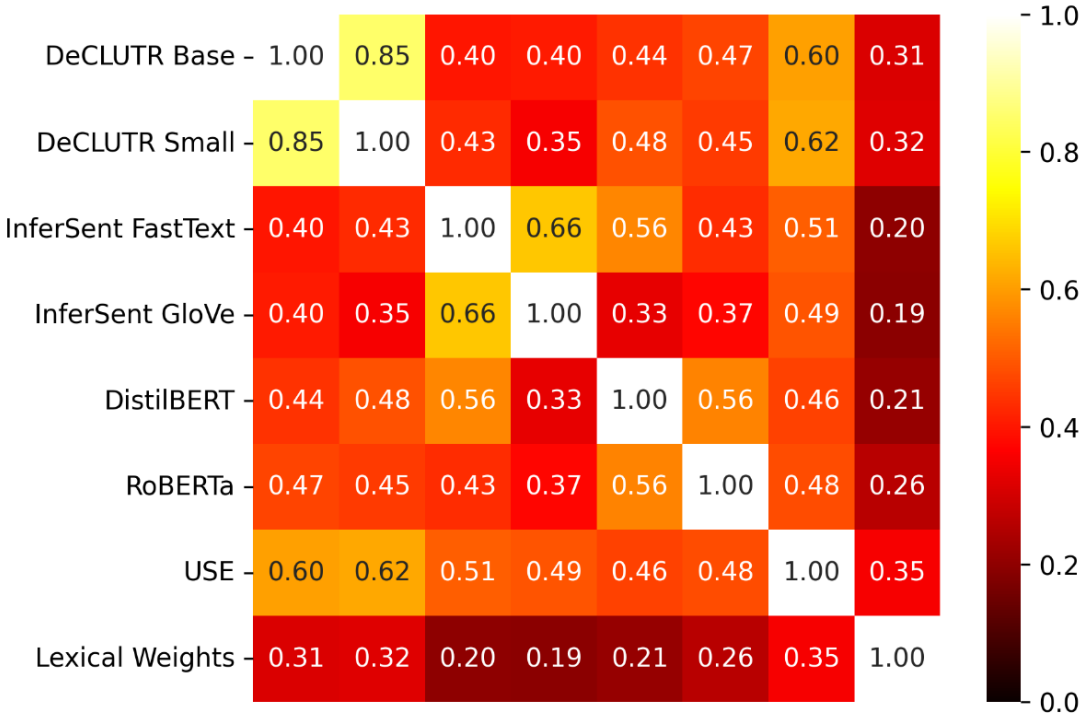


b) Heart of Darkness

Figure 4.21: Correlation plots of SSMs for *A Christmas Carol* and *Heart of Darkness*



a) Metamorphosis



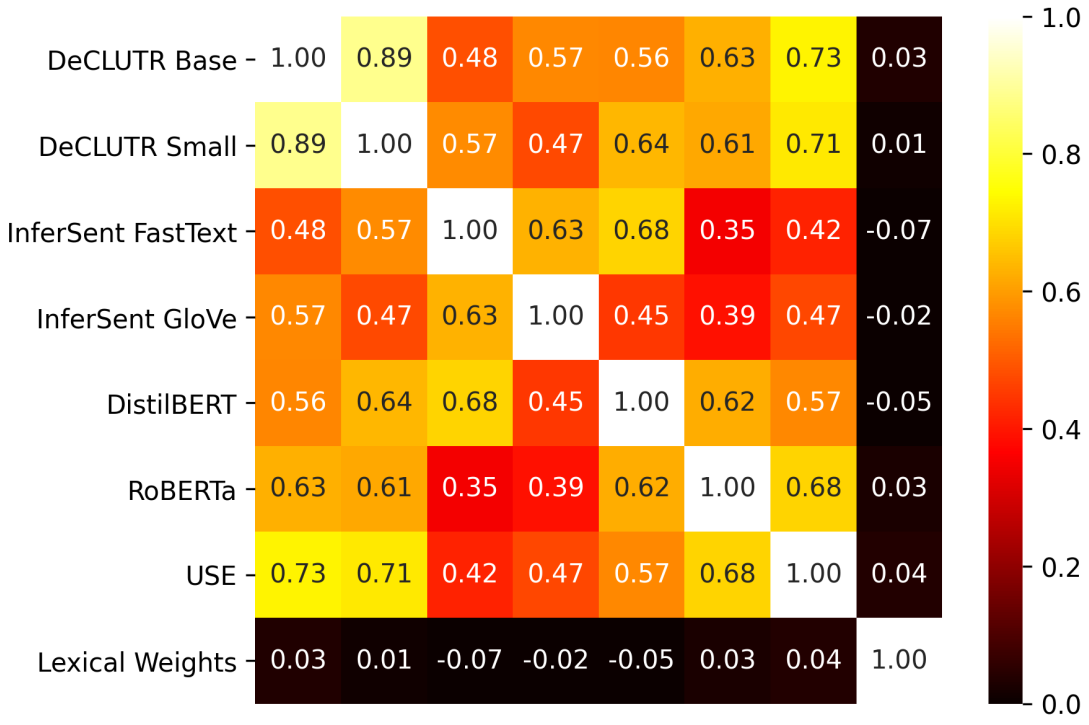
b) The Prophet

Figure 4.22: Correlation plots of SSMs for *Metamorphosis* and *The Prophet*

4.1.6 Correlation Plots of the Time-Series for the Four Texts

The correlation plots for the time-series from the four literary books are shown in Figures 4.23 and 4.24. They give an indication of which embedding methods are correlated with each other while they capture the temporal sequence. As expected, the correlation patterns are very similar – though not identical – to those for the SSMS, since the time-series are just the first super-diagonals of the SSMs. The main observations are as follows:

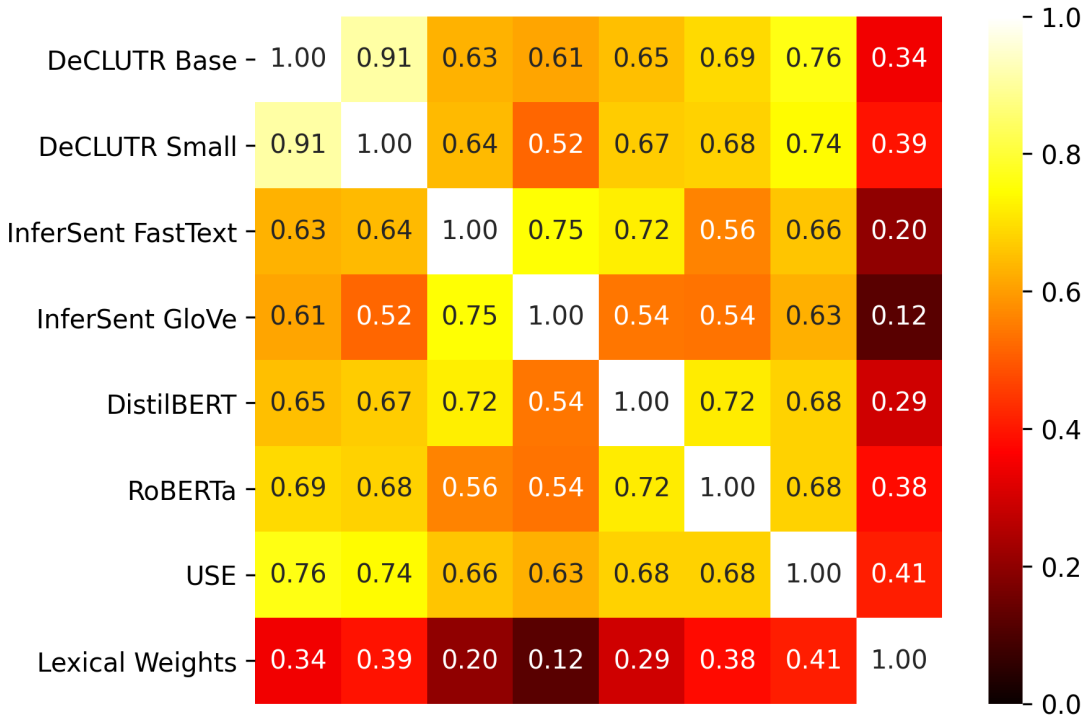
- There is very high correlation (0.89) between the time-series for the two DeCLUTR methods.
- There is high correlation (0.63) between the time-series of the two InferSent methods.
- There is high correlation (0.62) between DistilBERT and RoBERTa.
- There is no correation between the Lexical Weights method and the other methods in *A Christmas Carol*, but significant correlations ranging from 0.23 to 0.51 with methods other than the two InferSent cases.
- The Lexical Weights method has consistently low correlation with the two InferSent methods on all four corpora.



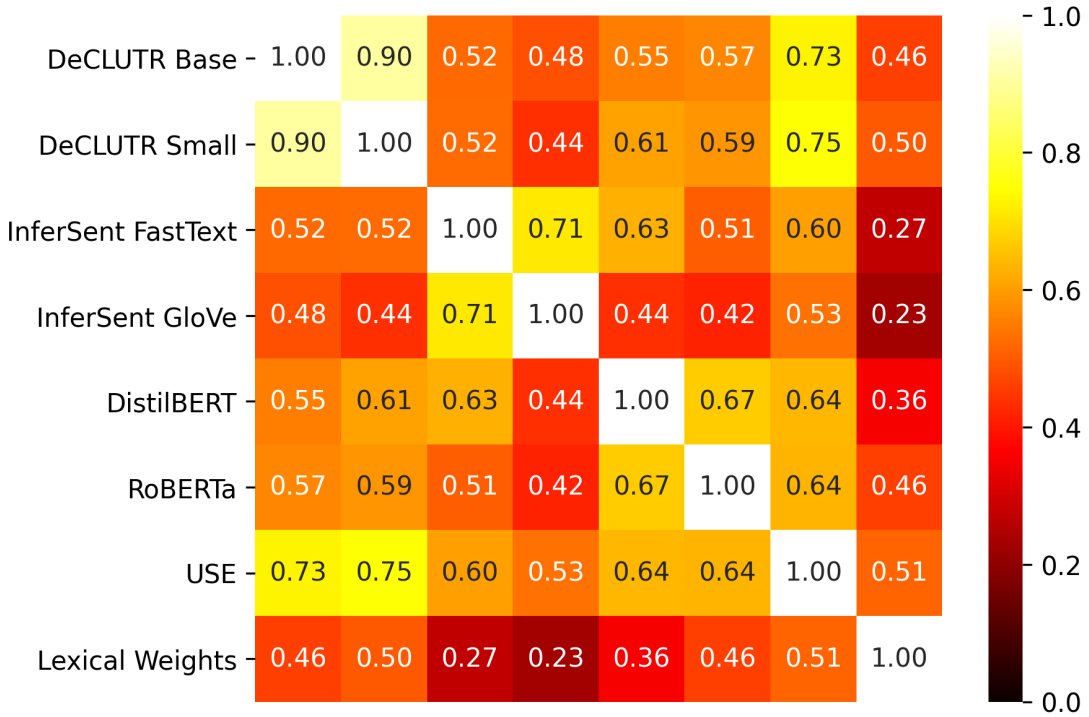
a) A Christmas Carol

b) Heart of Darkness

Figure 4.23: Correlation plots of time-series for *A Christmas Carol* and *Heart of Darkness*



a) Metamorphosis



b) The Prophet

Figure 4.24: Correlation Plots of time-series for *Metamorphosis* and *The Prophet*

4.1.7 Correlation Plots of the Time-Series for Translations

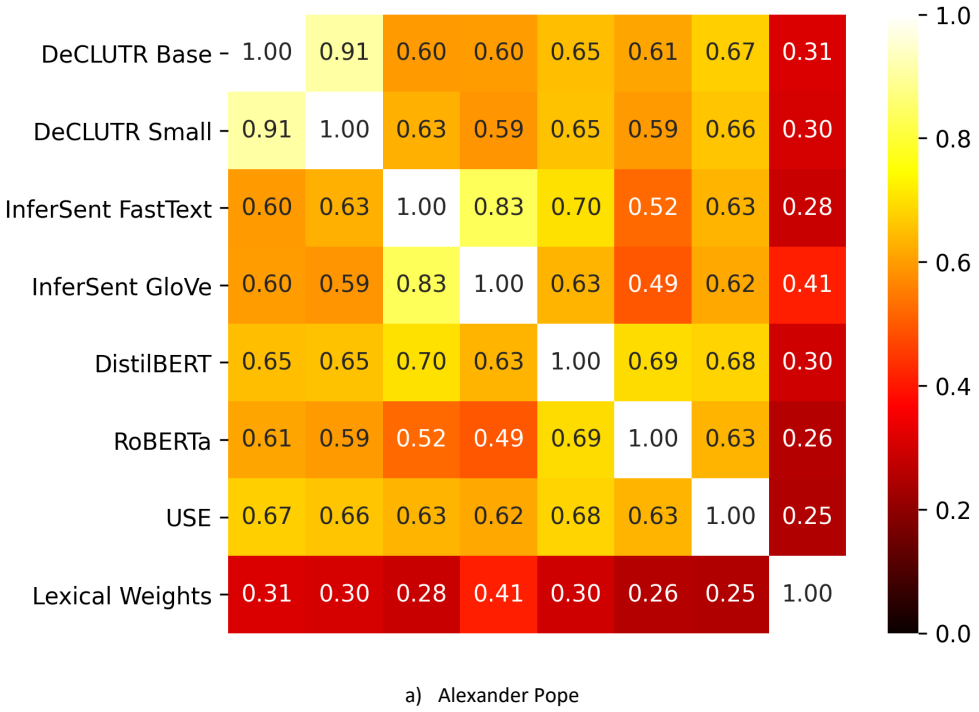
Correlation plots for *The Iliad* show similar patterns to those of the four literary books. As discussed previously, the translations are comparatively greater in length. As shown in Figure 4.25 (a), the translation by Alexander Pope has 5,334 sentences, which is the largest corpus analyzed in this thesis. The correlation plot for this corpus shows the interesting result that the Lexical Weights method performs is more correlated with the embedding-based models than in the other corpora, with the highest correlation ($\rho = 0.41$) with InferSent GloVe. These observations can most likely be explained by two factors:

1. While all other methods use text embeddings trained on large, generic train sets such as the Wikipedia or Google News corpora, the representations produced by the Lexical Weights method are obtained only from the corpus itself. While this can be an advantage because it makes the method self-sufficient and corpus-specific, it also means that inference may be poor in short corpora than cannot provide enough training data. The fact that the Pope Iliad is much longer than the other corpora may have improved the performance of the Lexical Weights method.

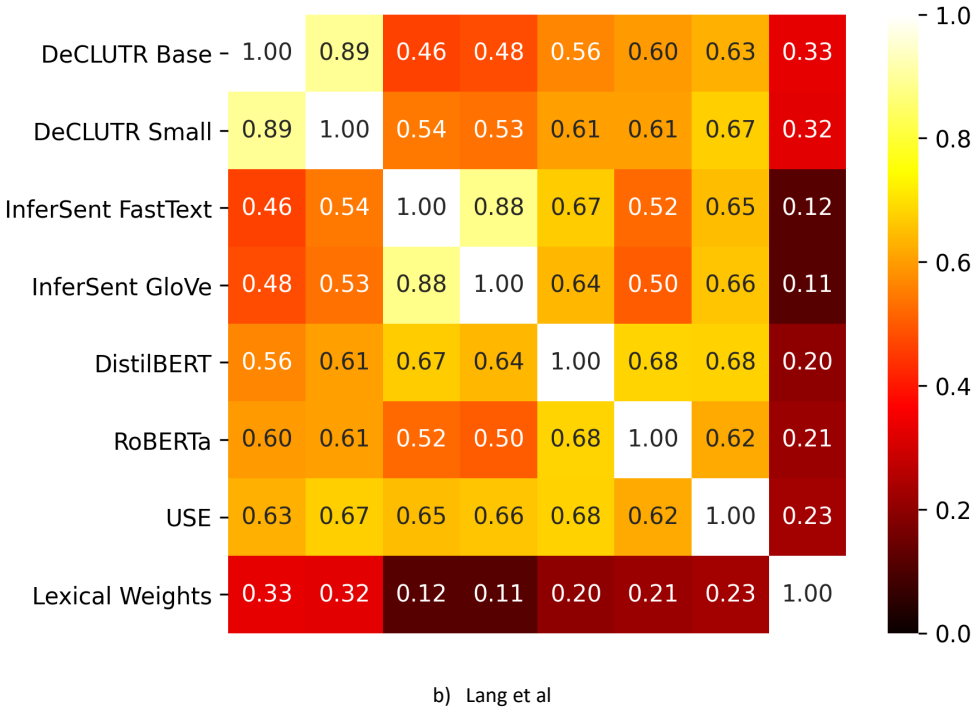
2. Infersent GLoVe is based on GloVe word embeddings [42], which are based on the co-occurrence statistics of words, just like the PMI-based weights in the Lexical Weights method. This affinity may be partly the reason for the relatively higher correlation between the two methods. This effect is seen in several other corpora as well, e.g., for the William Cowper translation of the Odyssey, the correlation value reaches 0.5.

In contrast, Figure 4.27 (b) for the Samuel Butler translation of the Odyssey should almost no correlation between The Lexical Weights method and the others. This is once again due to the conversational nature of the dataset. Samuel Butler's translation of The Odyssey is done in a style that involves much more dialogue between the various characters in the book.

Translations of The Aeneid and The Meditations show patterns similar to those observed for the previous books.



a) Alexander Pope



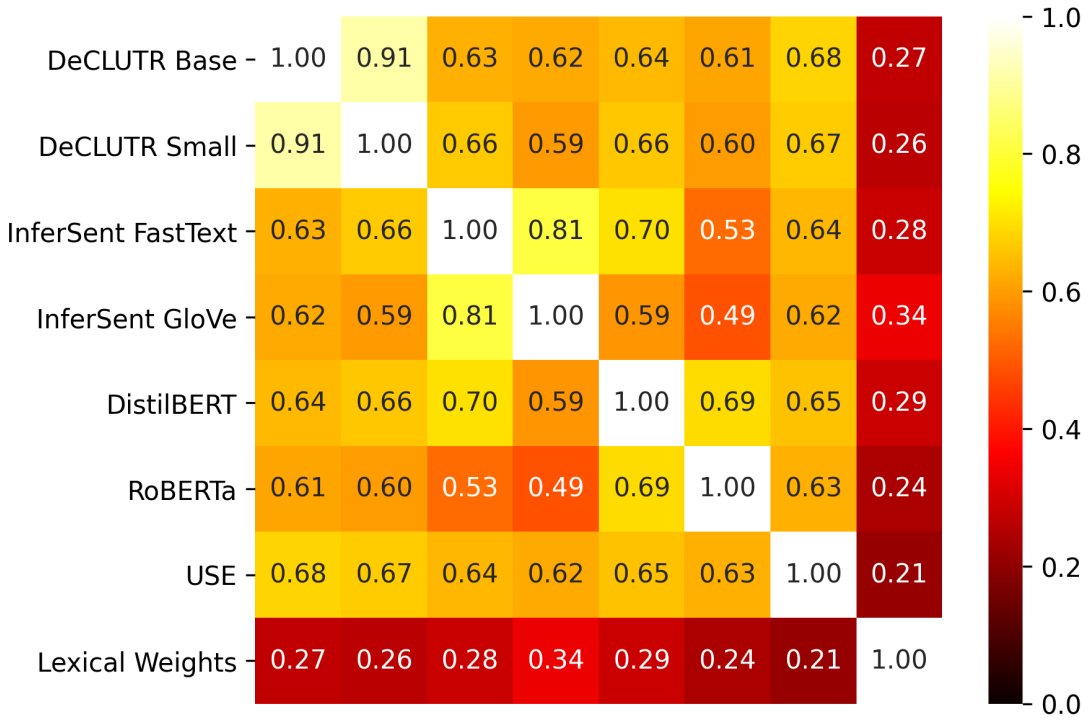
b) Lang et al

Figure 4.25: Correlation plots of time-series for *The Iliad* by Alexander Pope and Lang et al

a) George Chapman

b) Samuel Butler

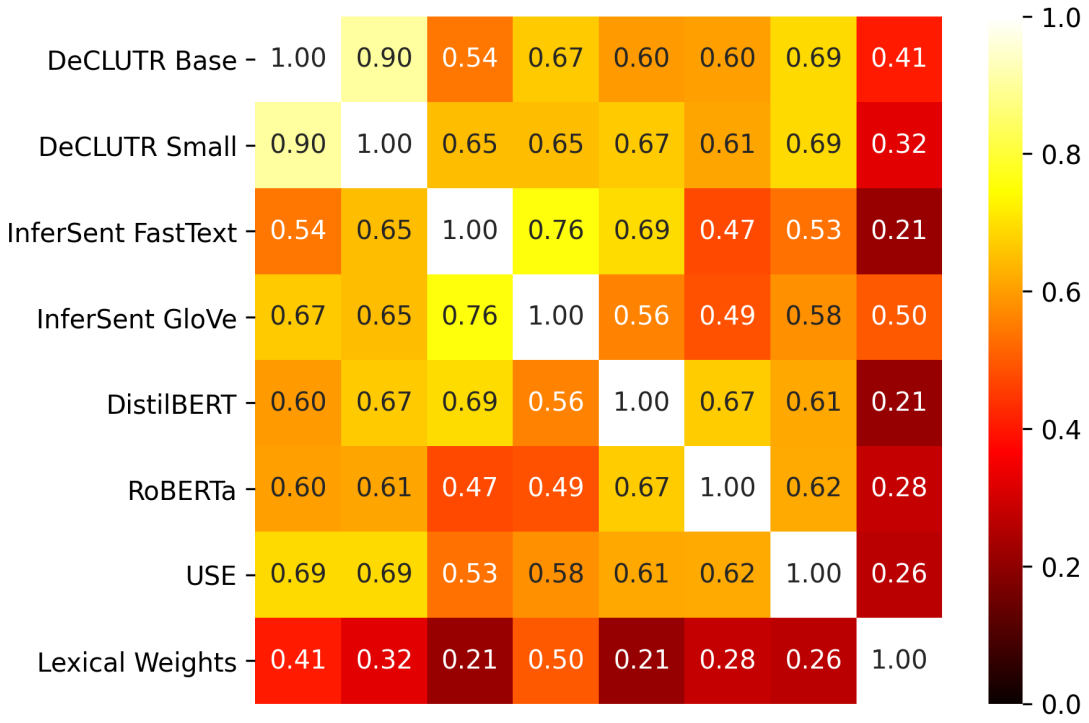
Figure 4.26: Correlation plots of time-series for *The Iliad* by George Chapman and Samuel Butler



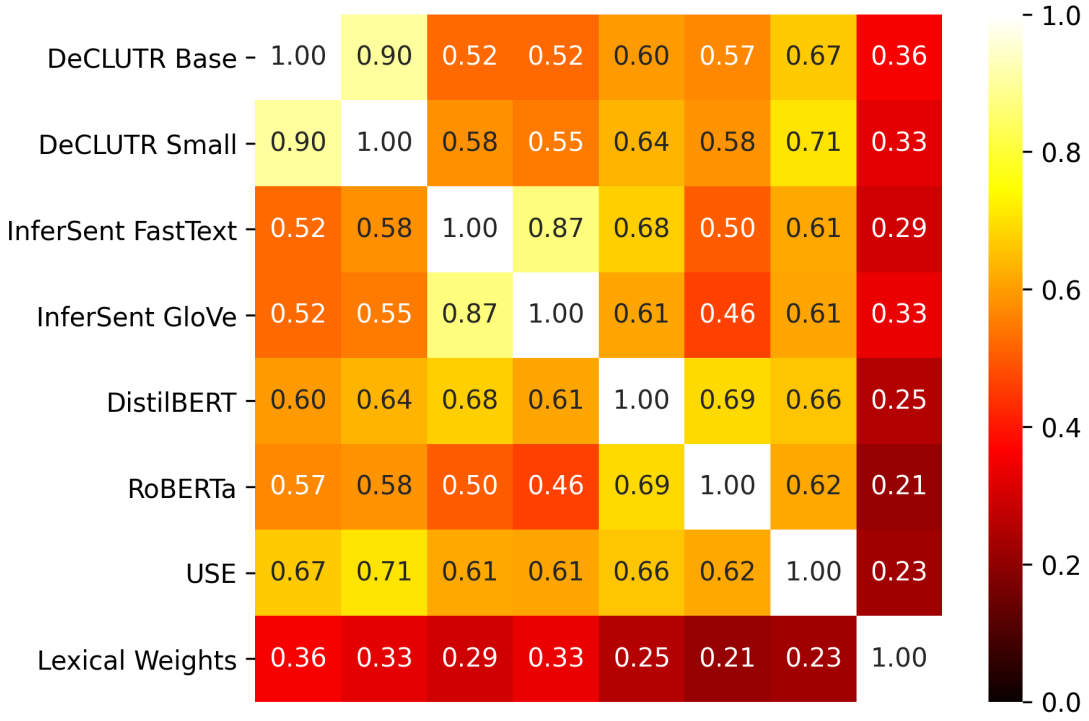
a) Alexander Pope

b) Samuel Butler

Figure 4.27: Correlation plots of time-series for *The Odyssey* by Alexander Pope and Samuel Butler

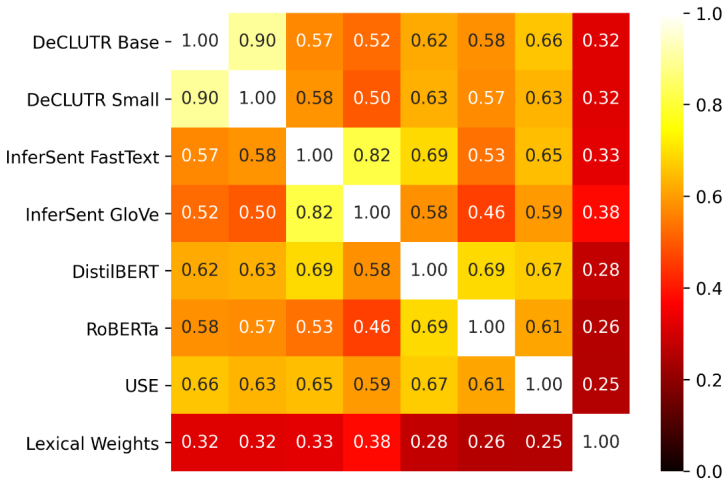


a) William Cowper

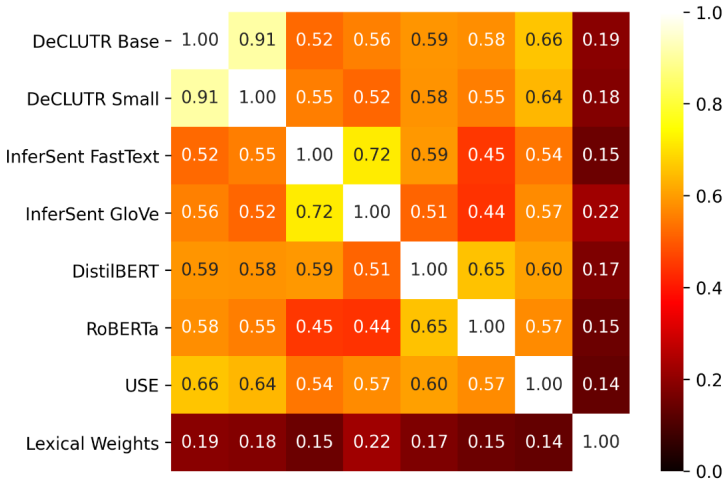


b) Butcher and Lang

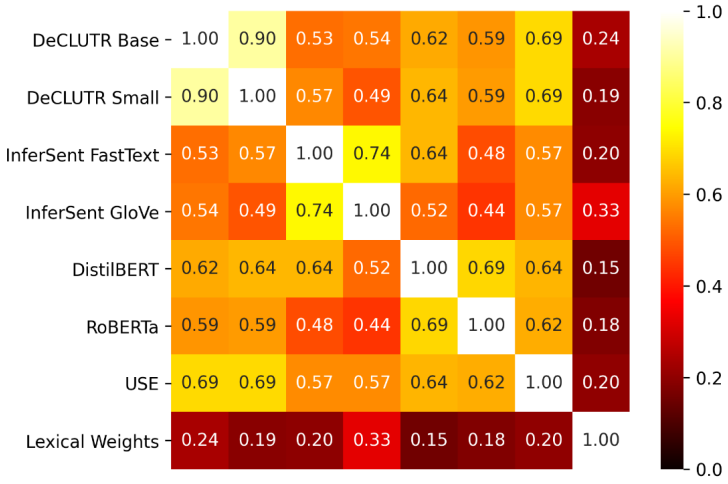
Figure 4.28: Correlation plots of time-series for *The Odyssey* by William Cowper and Butcher and Lang



a) J. W. Mackail

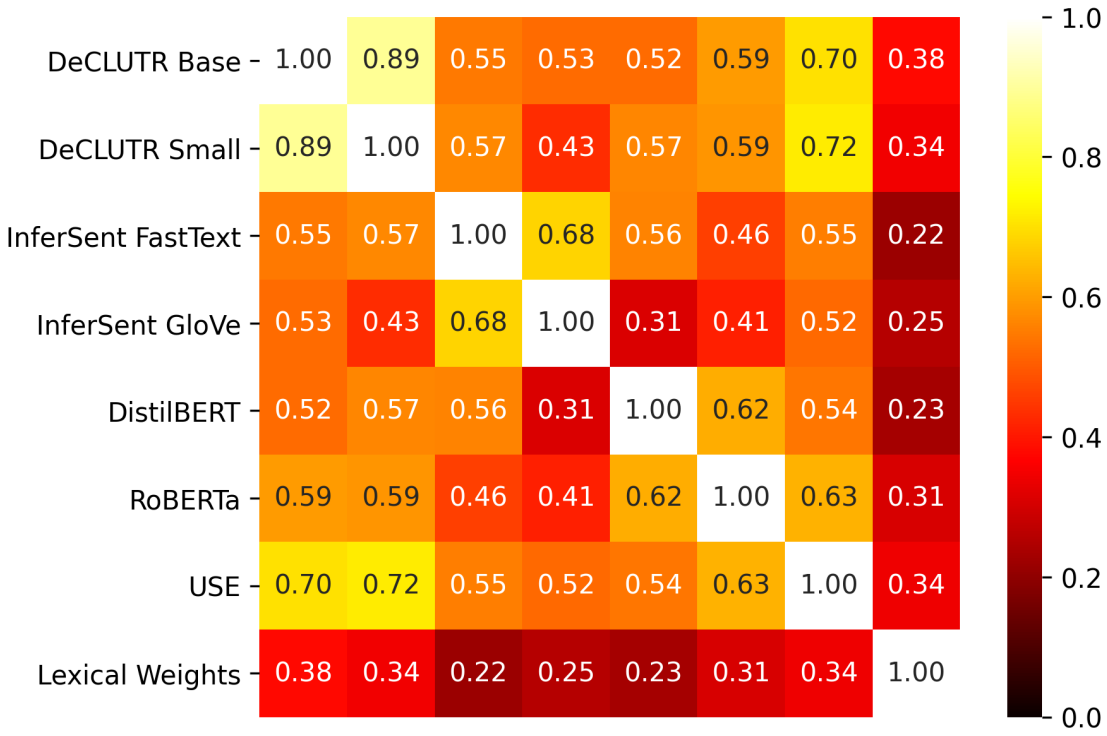


b) John Dryden

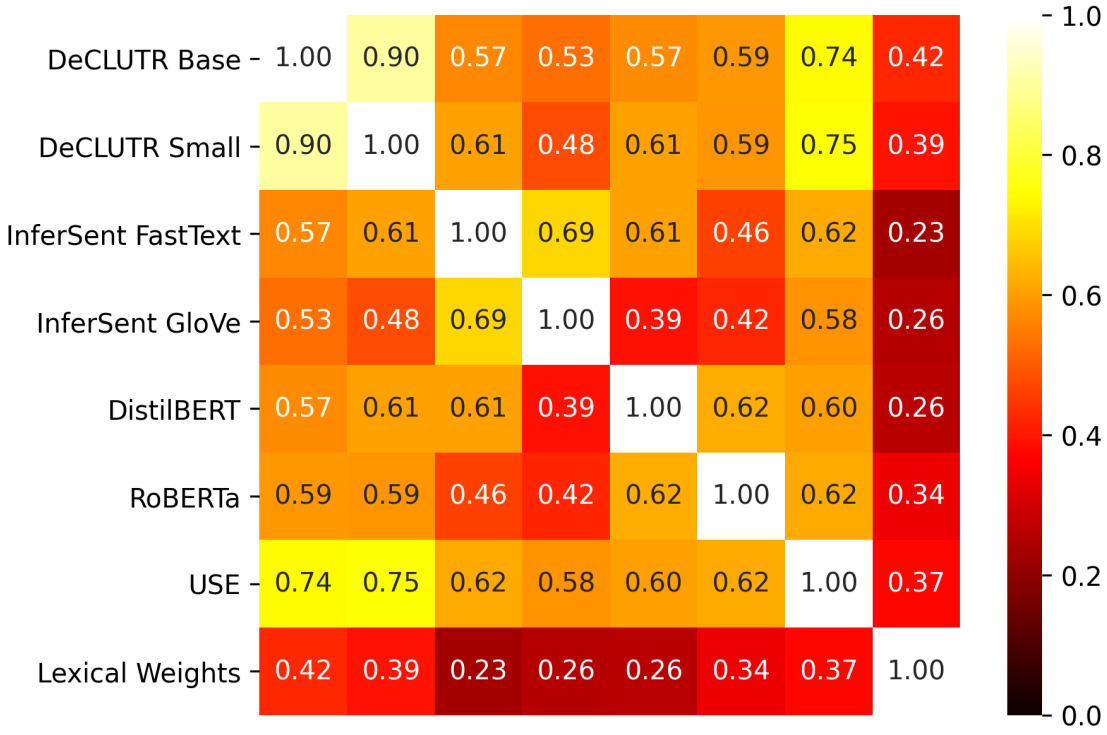


c) Rolfe Humphries

Figure 4.29: Correlation plots of time-series for *The Aeneid*



a) George Chrystal



b) Meric Casaubon

Figure 4.30: Correlation plots of time-series for *The Meditations*

4.1.8 Global Correlation Coefficient Plot of SSMs

The global correlation coefficient matrix is calculated by taking the mean of all the correlation coefficients of SSMs across all the 17 books analyzed.

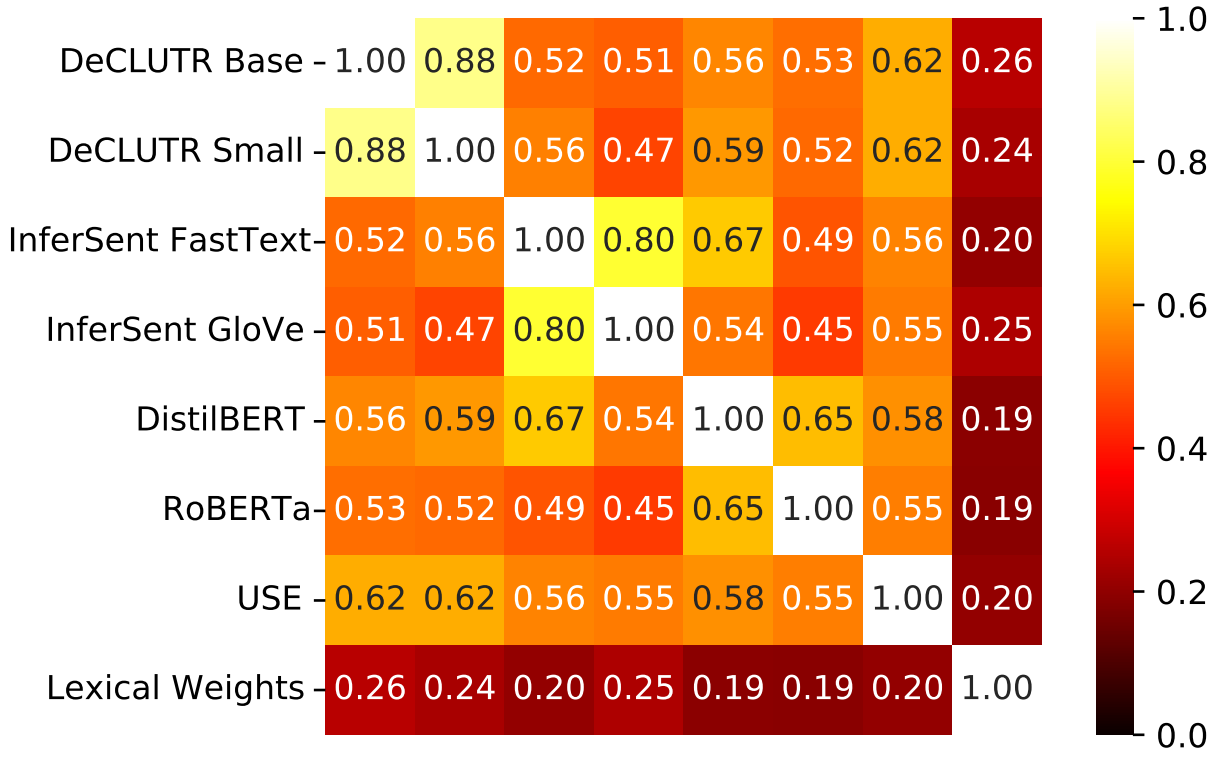


Figure 4.31: Correlation plot of the Global Mean of all the SSMs

4.1.9 Global Correlation Coefficient Plot of time-series

The global correlation coefficient matrix is calculated by taking the mean of all the correlation coefficients of time-series across all the 17 books analyzed.

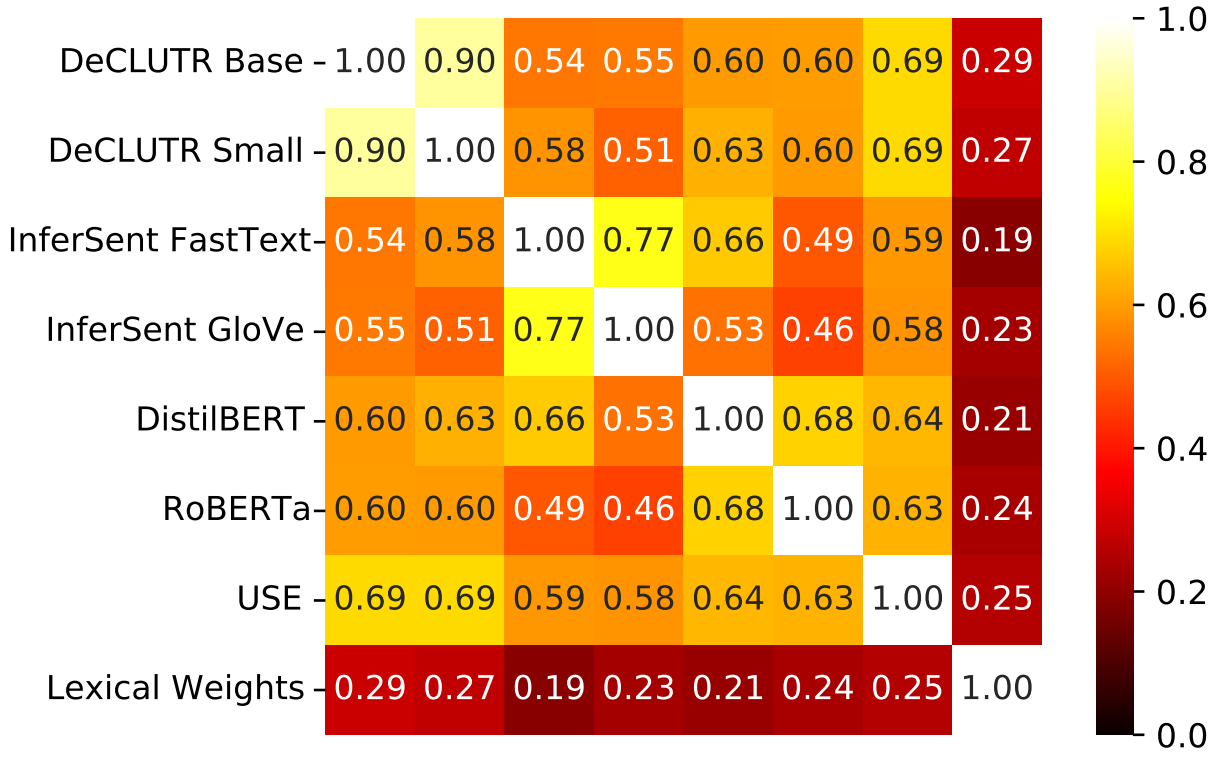


Figure 4.32: Correlation plot of the Global Mean of all the time-series

4.2 Results from Study II

In this section, the results from the procedures mentioned in Section 3.7 are visualized. Results are organized in the same order that they appear in Section 3.7.

4.2.1 Results from the Analysis of Human Ratings

Some interesting insights are obtained when the human raters rate the SPPS. Figure 4.33 shows the distribution of scores given by each individual rater to the sentence pairs in the LSSP and MSSP sets. An interesting characteristic emerges from the histograms: For the 320 LSSP pairs, most human raters were able to rate the vast majority as having low similarity, but the similarity values were much less correlated with those produced by the semantic models for the 320 MSSP pairs.

Collecting the ratings by all raters by method clarifies things more. Figure 4.34 shows the mean human ratings for the LSSP and MSSP for each of the eight methods. The taller the beige bar and the lower the blue, the more consistent the method is with human ratings. This is clarified further in 4.35, where each bar gives the difference between the MMSP and LSSP ratings for each metho. It can be seen that RoBERTa has the greatest difference, with the two DeCLUTR methods, DistilBERT, and USE close behind and very close to each other. The two InferSent methods fare significantly worse, and the Lexical Weights metho has the lowest correspondence with human raters.

When the similarity ratings for individual sentence pairs are visualized in Figure 4.36, the human ratings for the LSSP across all methods are reliably low (left half of the heatmap). Figure 4.37 shows the same data with the values for each rater standardized to have 0 mean and unit variance, thus plotting the z-score values relative to each rater's own scores. For the MSSP, humans and the semantic models disagree much more, though human raters agree quite significantly with each other (intraclass correlation coefficient = 0.9). Nevertheless, on average, human raters do think that most MSSP pairs are more similar than LSSP pairs.

It is interesting to note that human raters tend to give lower similarity scores to the same MSSP pairs across all raters.

Since the data points in Figure 4.36 are grouped by encoding method and by data source, one can get further information by plotting the heatmap in other ways. Figure 4.38 averages the LSSP and MSSP ratings of each rater across all the 40 sentence pairs generated by the same encoder. Looking down the columns in the MSSP half, it is clear that human rating were least similar to InferSent and Lexical Weights rating for all raters. Then averaging each column over the rows gives the bar plot in Figure 4.39, showing the mean ratings human raters gave to the LSSP and MSSP pairs produced by each method.

Finally, in Figure 4.40, the results for each method and each rater are separated by the source tests, with the ratings for the 10 LSSP and 10 MSSP pairs from each book averaged separately. Thus, each method column in the previous heatmap is split into four columns, one each for *A Christmas Carol*, *Heart of Darkness*, *Metamorphosis* and *The Prophet* in order. Now looking at each small column, it becomes clear that, in most cases, high-similarity sentence pairs produced by *Metamorphosis* elicited the most disagreement from human raters. The only exception was the sentence pairs from the Lexical Weights method, where *The Prophet* produced the highest level of disagreement. In contrast, high-similarity sentence pairs generated from *A Christmas Carol* by all the embedding-based methods obtained the highest level of agreement from human raters. Again, Lexical Weights MSSP pairs are an exception, where *Heart of Darkness* fared best.

Finally, the columns of the heatmap in Figure 4.40 are averaged across all raters to get the mean ratings for each book with each encoding method. These are plotted in Figure 4.41, with the cource books color-coded, and then plotted separately for each book in Figure 4.42. This last figure makes clear the human ratings for *A Christmas Carol* are most consistent with those inferred by the models (except for Lexical Weights), while consistency is worst for *Metamorphosis*. This is especially interesting because, of the four books, the text for *Metamorphosis* - taken from a recent translation - is the most modern.

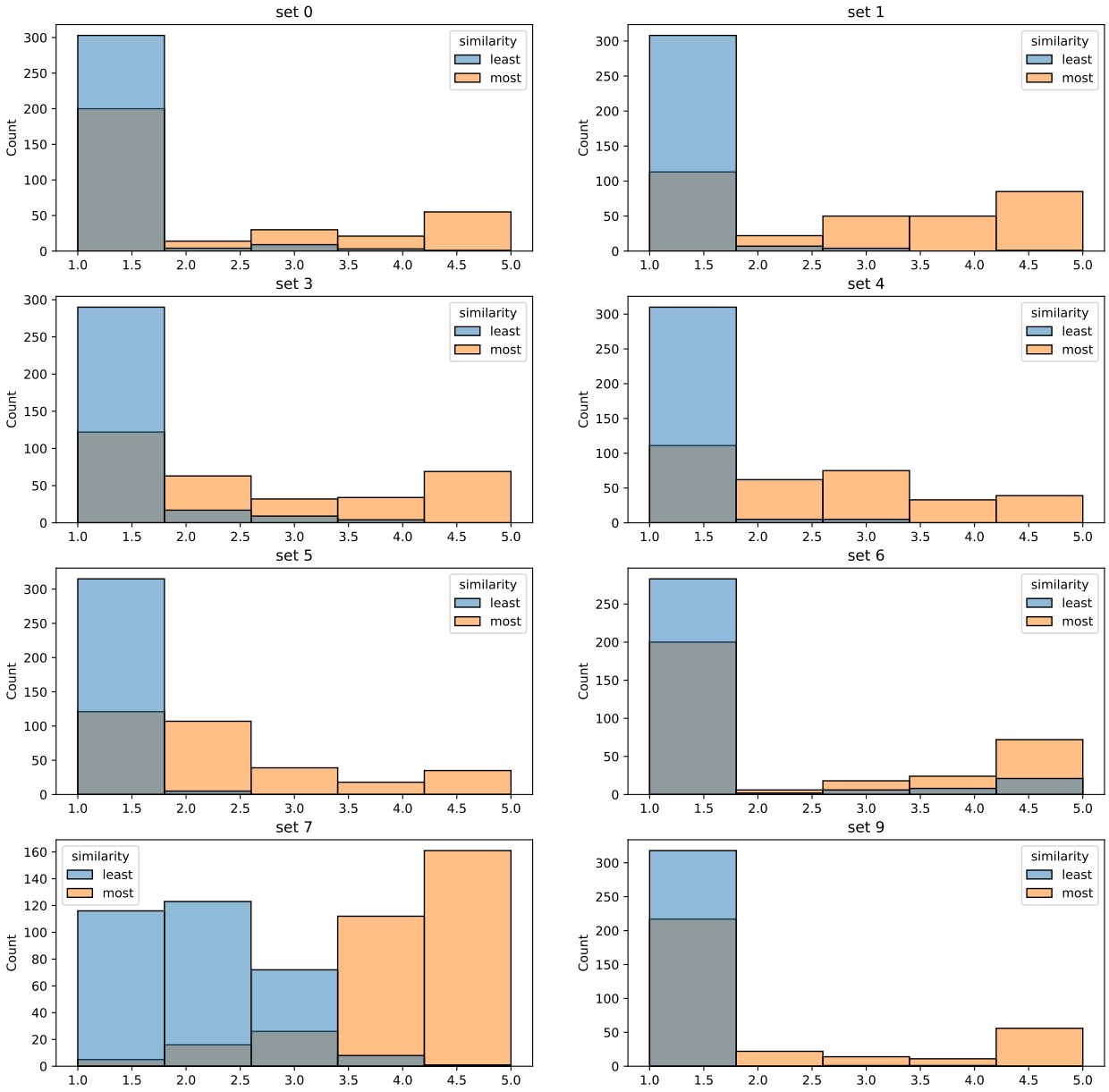


Figure 4.33: Histograms of ratings for individual raters. The blue bars are for the LSSP and the beige bars for the MSSP.

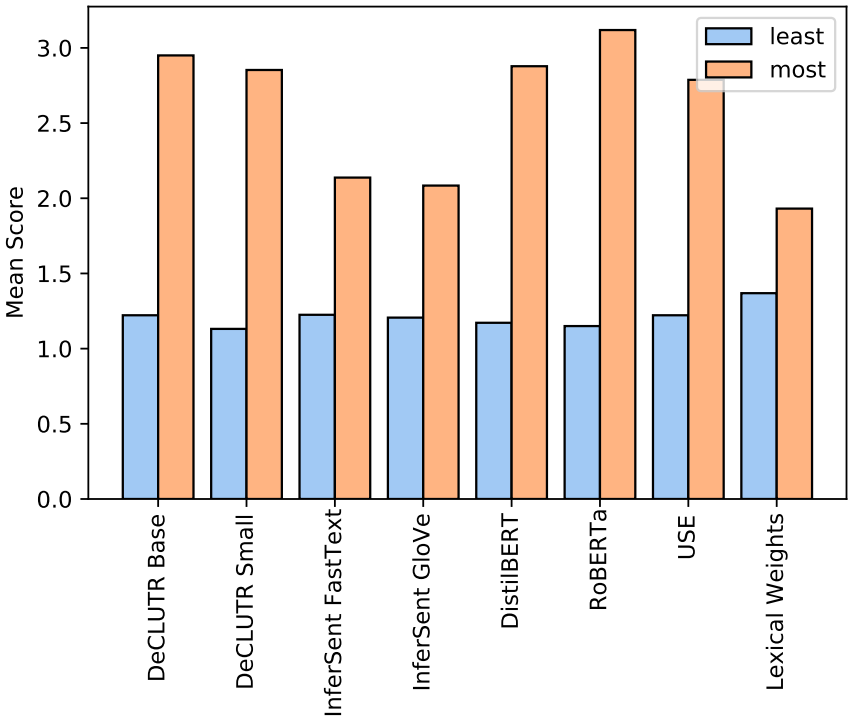


Figure 4.34: Mean ratings of all 8 human raters split by embedding methods

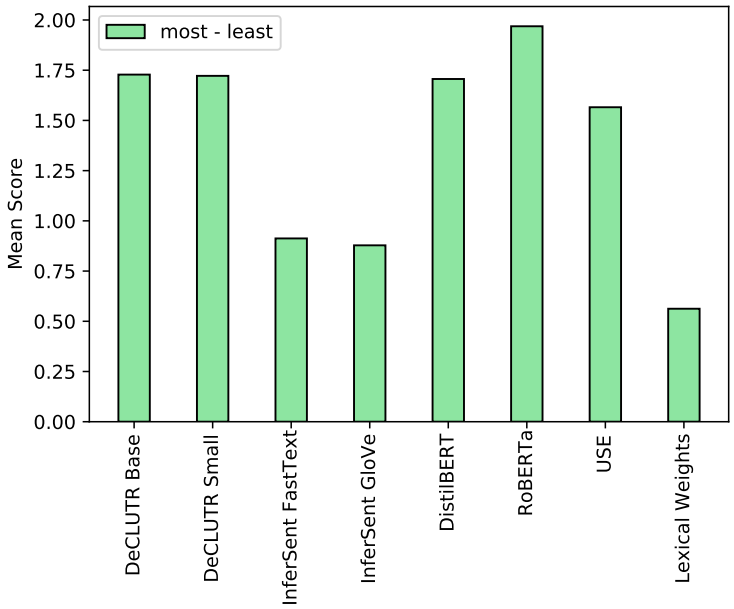


Figure 4.35: Difference of mean ratings of all 8 human raters split by embedding methods

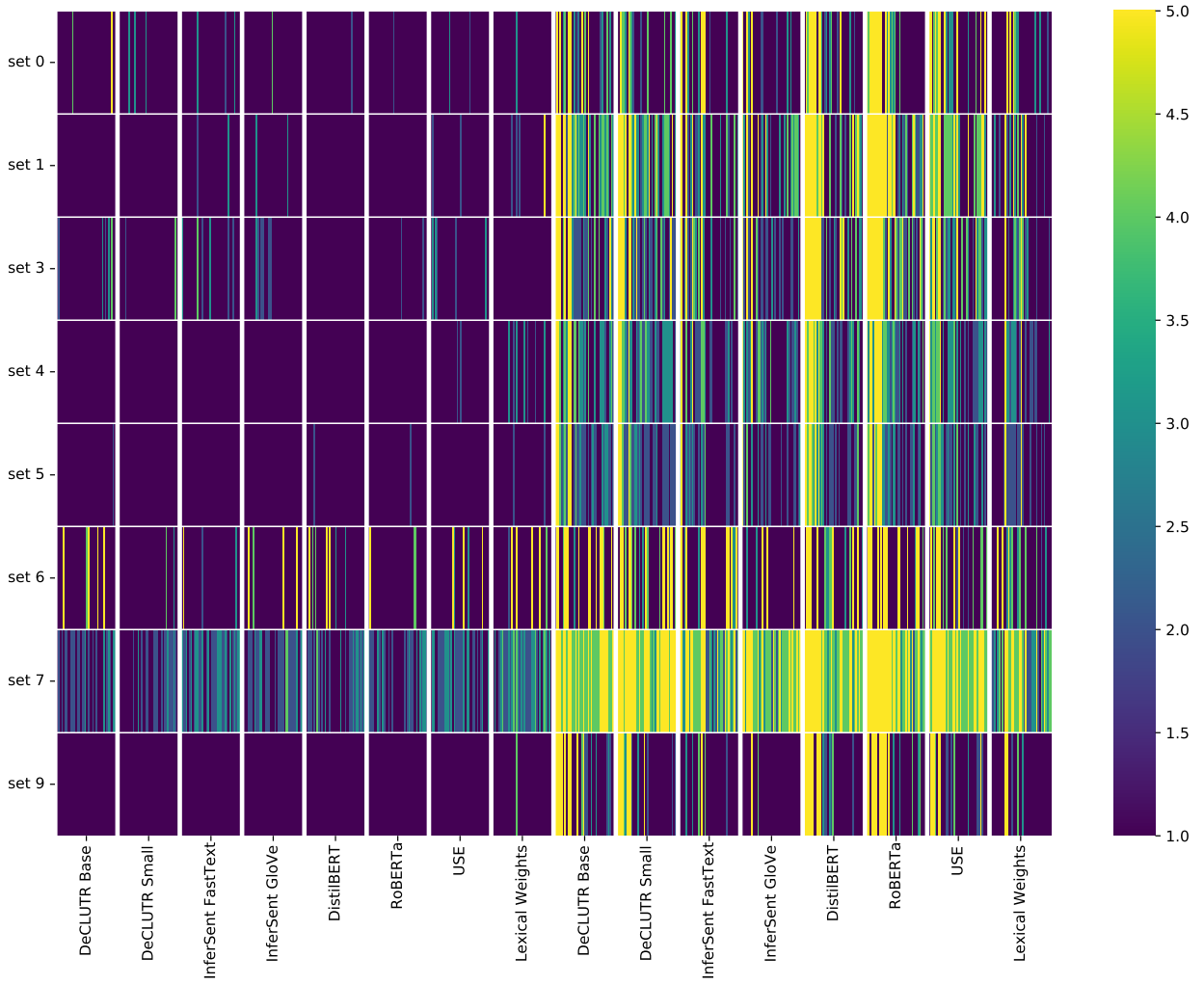


Figure 4.36: Heatmap of rating for individual sentence pairs by each rater. The left half of the figure shows sentence pairs from the LSSP, and the right half from MSSP. Each row represents a rater, and eavh wide column an encoding methos. Within each wide column, there are 40 sentence pairs –10 from each book. Ideally, all LSSP airs should have a rating near 1 and all MSSP pairs a rating near 5.

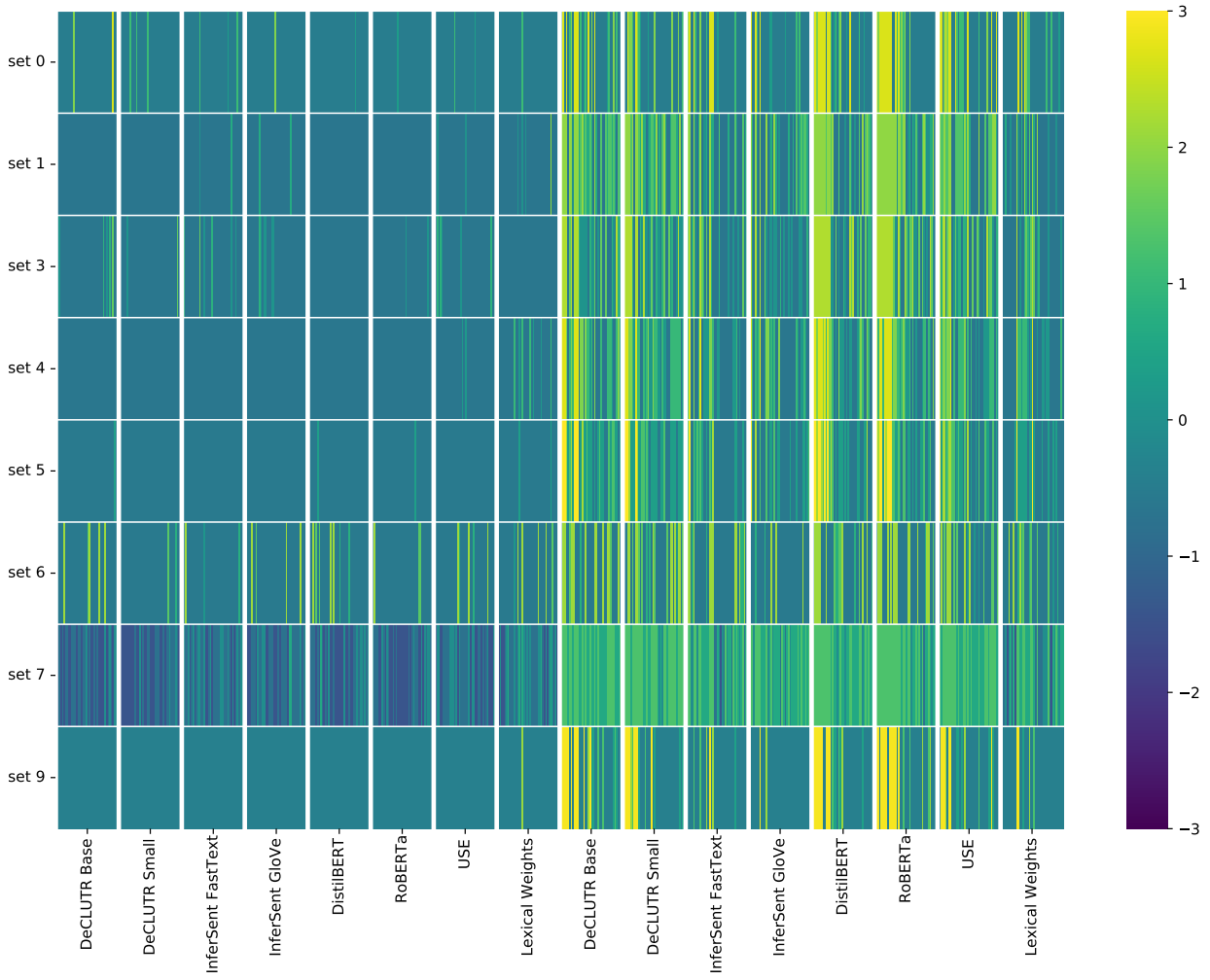


Figure 4.37: Heatmap of z-score ratings standardized for each rater.

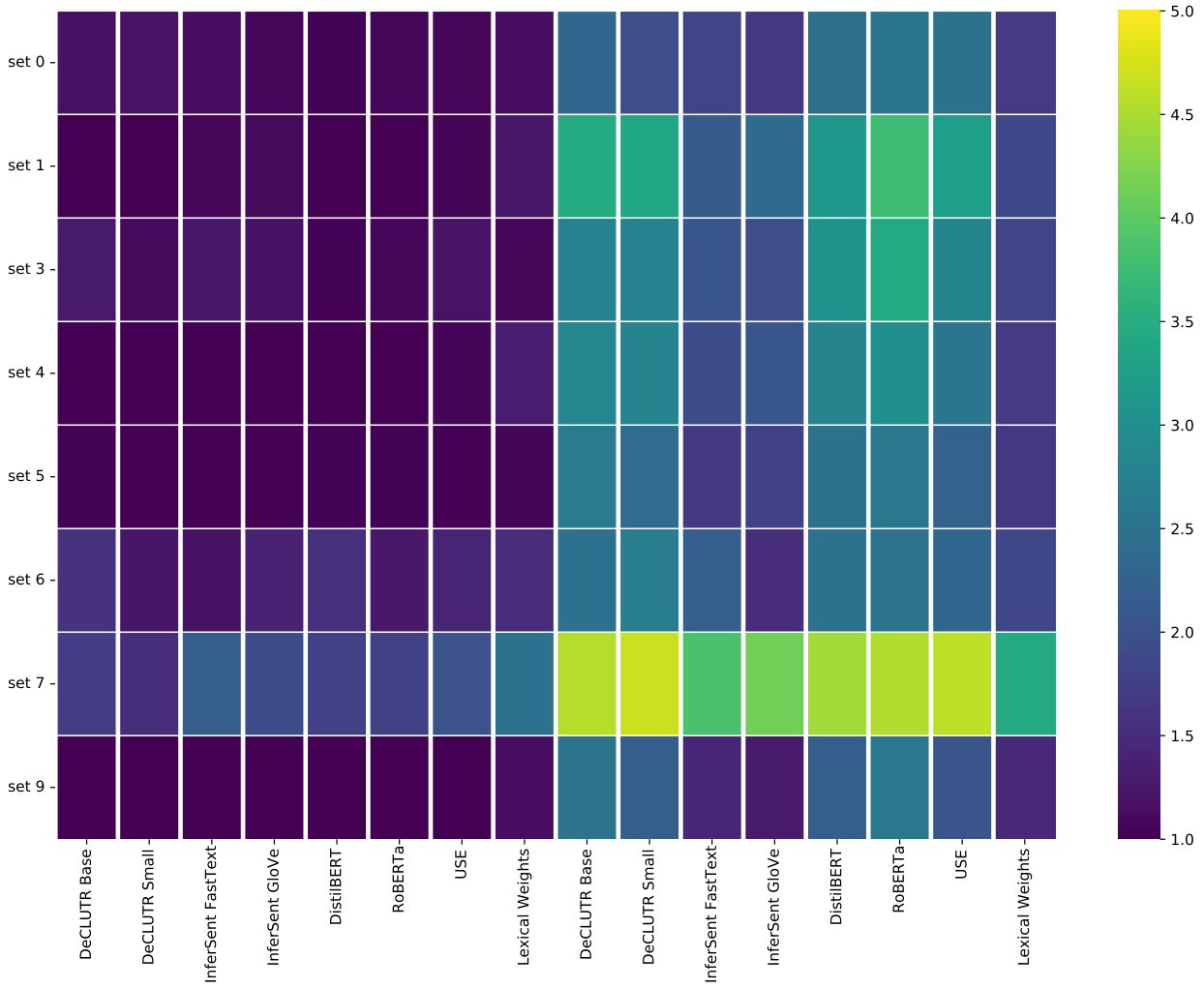


Figure 4.38: Heatmap of mean ratings split by method

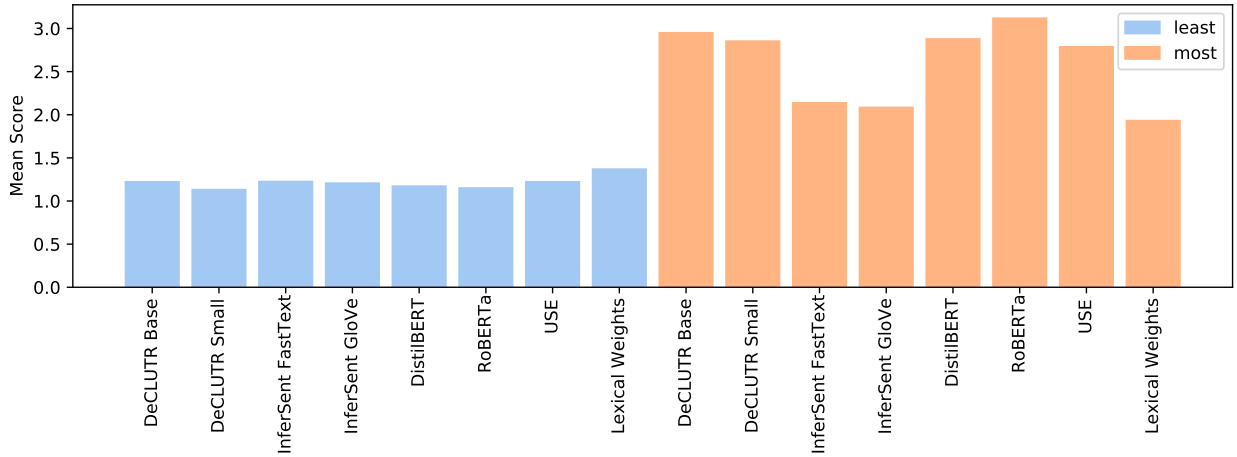


Figure 4.39: Barplot of mean ratings split by method

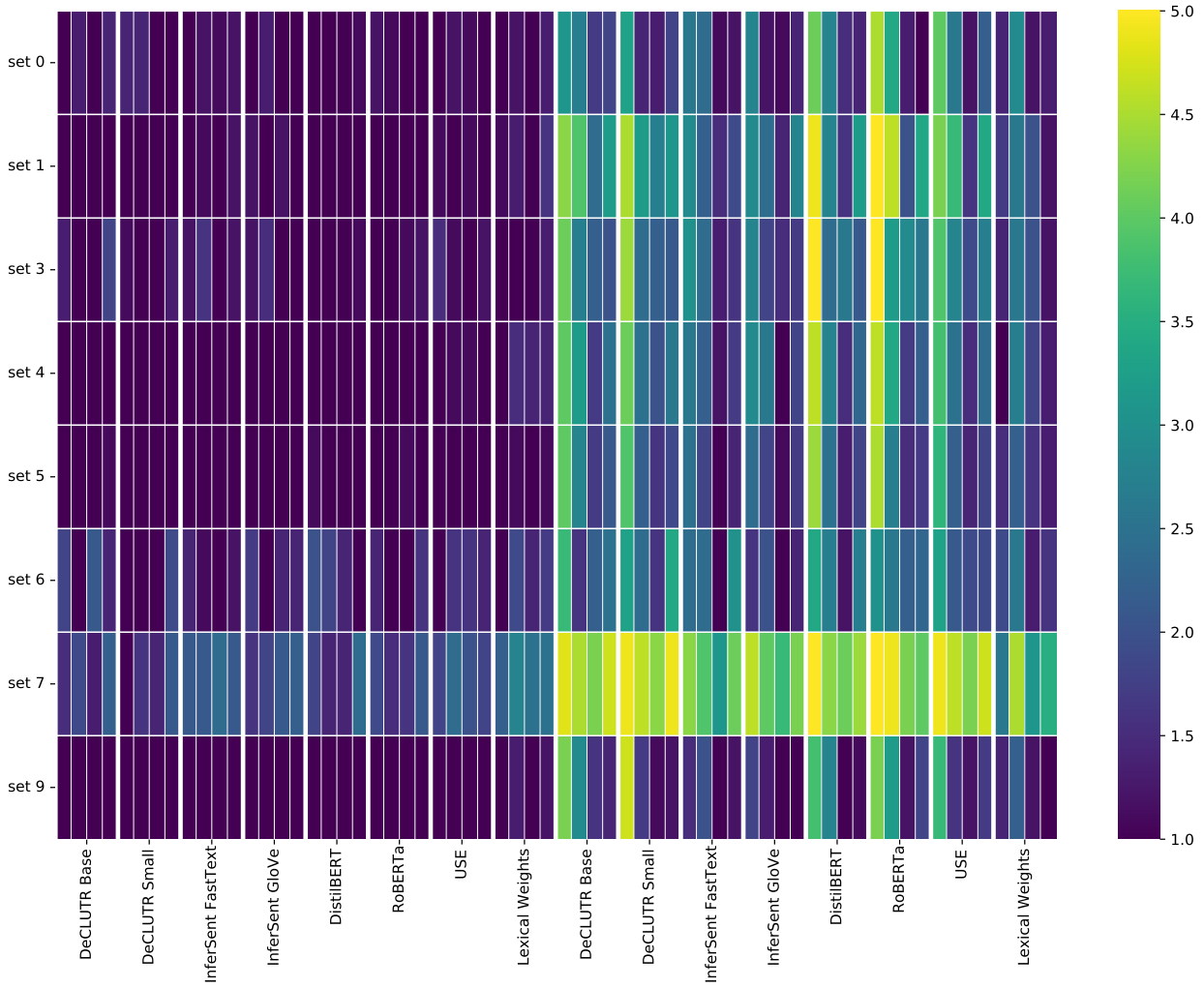


Figure 4.40: Heatmap of mean ratings split by individual books and method

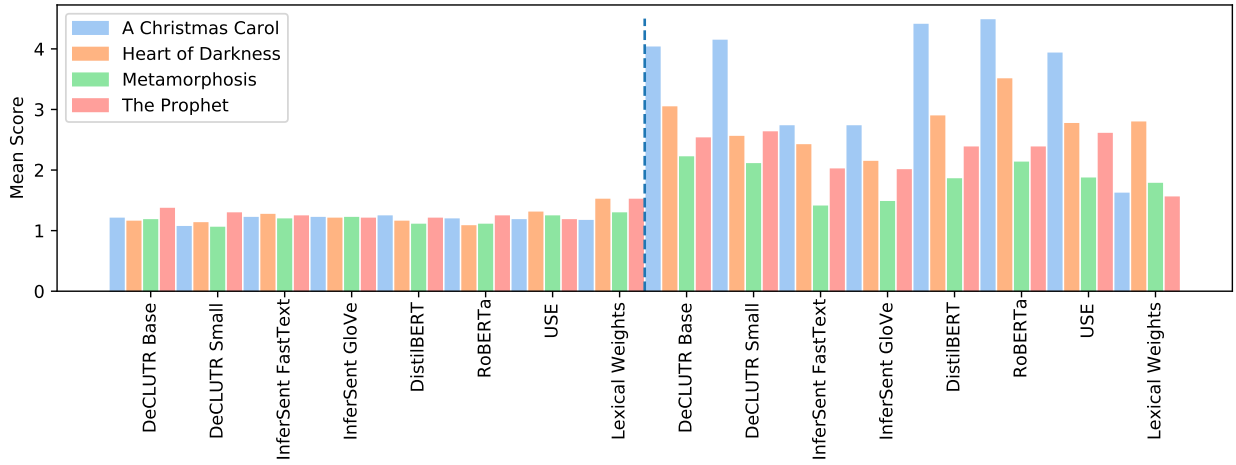


Figure 4.41: Barplot of mean ratings split by individual books and method

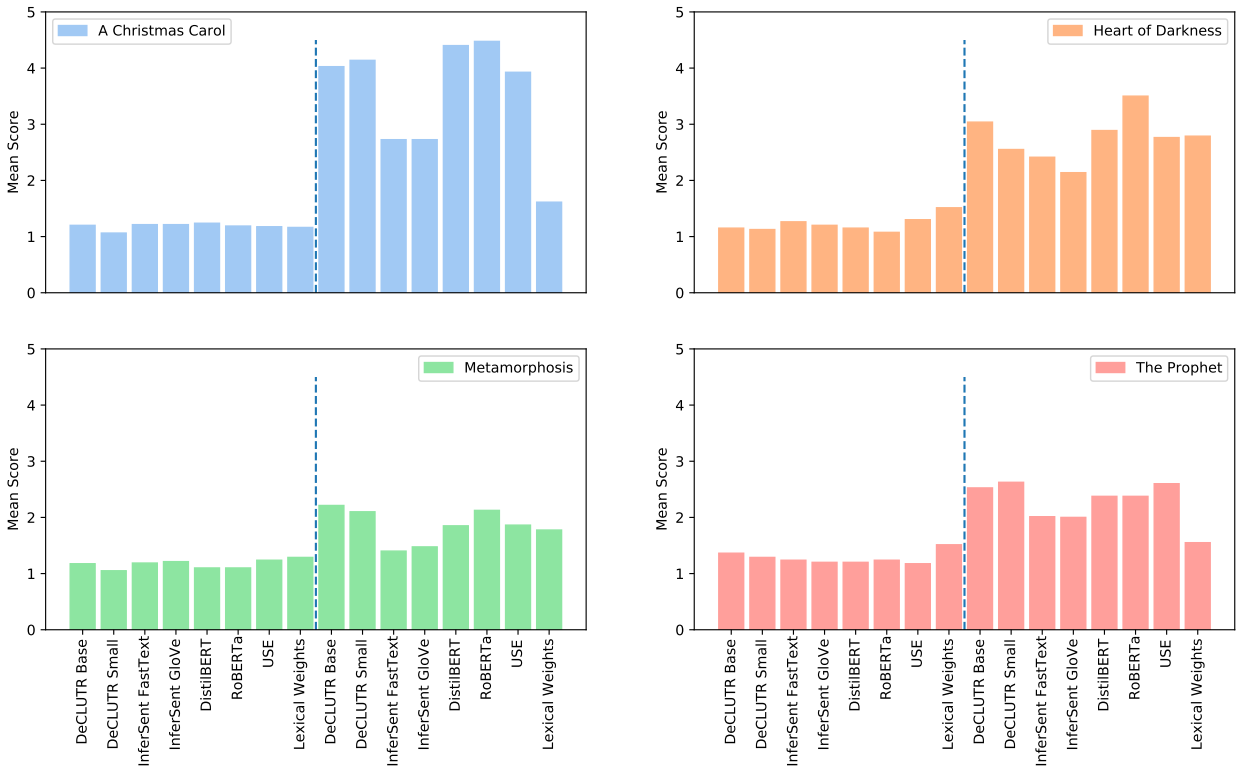


Figure 4.42: Barplot of mean ratings grouped by individual books

4.2.2 Results from DTW Analysis

Tables 4.1 - 4.8 show the results of applying the DTW-based analysis described in Section 3.7.2 to the time-series produced by each encoder. Since DTW is a pairwise operation which can only warp 2 time-series at once, only two translations of the same book can be compared at a time. The *Pairs* column in the tables shows which two translations are being compared (see Table 3.1 for the labels). In these tables W_c^1 is the warp coefficient obtained by calculating the percentage change in the length of the warped time-series with respect to the first time-series, W_c^2 the percentage change in the length of the warped time-series with respect to the second time-series,. $\bar{W}_c$ is the mean of W_c^1 and W_c^2, $Z_{z,t}$ is the correlation coefficient between the two warped time-series, and Q is the ratio of $Z_{z,t}$ and $\bar{W}_c$, which is used as a measure of semantic consistency. A high Q value indicates higher consistency.

	DeCLUTR Base				
Pairs	ω_c^1	ω_c^2	Ω_c	$z_{s,t}$	Q
A1 - A2	0.24	0.55	0.40	0.83	2.09
A1 - A3	0.26	0.51	0.39	0.83	2.15
A2 - A3	0.39	0.33	0.36	0.84	2.31
I1 - I2	0.33	0.41	0.37	0.83	2.24
I1 - I3	0.25	0.49	0.37	0.82	2.22
I1 - I4	0.58	0.24	0.41	0.83	2.04
I2 - I3	0.29	0.47	0.38	0.83	2.20
I2 - I4	0.65	0.23	0.44	0.83	1.90
I3 - I4	0.76	0.16	0.46	0.82	1.80
O1 - O2	0.25	0.48	0.37	0.83	2.25
O1 - O3	0.61	0.21	0.41	0.83	2.00
O1 - O4	0.42	0.34	0.38	0.83	2.21
O2 - O3	0.81	0.15	0.48	0.82	1.72
O2 - O4	0.55	0.23	0.39	0.83	2.13
O3 - O4	0.24	0.55	0.40	0.84	2.12
M1 - M2	0.34	0.35	0.34	0.84	2.43

Table 4.1: Warp coefficients for DeCLUTR Base

	DeCLUTR Small				
Pairs	ω_c^1	ω_c^2	Ω_c	$z_{s,t}$	Q
A1 - A2	0.23	0.54	0.38	0.82	2.14
A1 - A3	0.26	0.50	0.38	0.83	2.16
A2 - A3	0.38	0.32	0.35	0.83	2.36
I1 - I2	0.33	0.40	0.37	0.83	2.24
I1 - I3	0.25	0.50	0.38	0.83	2.20
I1 - I4	0.56	0.23	0.40	0.82	2.08
I2 - I3	0.30	0.47	0.39	0.83	2.15
I2 - I4	0.62	0.21	0.42	0.82	1.96
I3 - I4	0.77	0.16	0.47	0.82	1.75
O1 - O2	0.26	0.49	0.38	0.83	2.20
O1 - O3	0.61	0.21	0.41	0.82	2.00
O1 - O4	0.41	0.33	0.37	0.83	2.25
O2 - O3	0.82	0.15	0.49	0.81	1.67
O2 - O4	0.55	0.23	0.39	0.83	2.12
O3 - O4	0.24	0.56	0.40	0.83	2.06
M1 - M2	0.35	0.36	0.35	0.83	2.35

Table 4.2: Warp coefficients for DeCLUTR Small

	DistilBERT				
Pairs	ω_c^1	ω_c^2	Ω_c	$z_{s,t}$	Q
A1 - A2	0.25	0.56	0.41	0.82	2.02
A1 - A3	0.27	0.51	0.39	0.82	2.13
A2 - A3	0.4	0.34	0.37	0.83	2.24
I1 - I2	0.35	0.42	0.39	0.84	2.16
I1 - I3	0.28	0.53	0.4	0.84	2.08
I1 - I4	0.59	0.25	0.42	0.84	1.97
I2 - I3	0.3	0.48	0.39	0.84	2.13
I2 - I4	0.65	0.23	0.44	0.83	1.89
I3 - I4	0.8	0.18	0.49	0.82	1.67
O1 - O2	0.26	0.5	0.38	0.83	2.17
O1 - O3	0.61	0.21	0.41	0.82	2.01
O1 - O4	0.42	0.33	0.37	0.83	2.22
O2 - O3	0.83	0.16	0.49	0.8	1.63
O2 - O4	0.56	0.24	0.4	0.83	2.09
O3 - O4	0.24	0.56	0.4	0.82	2.03
M1 - M2	0.34	0.35	0.35	0.84	2.42

Table 4.3: Warp coefficients for DistilBERT

	InferSent FastText				
Pairs	ω_c^1	ω_c^2	Ω_c	$z_{s,t}$	Q
A1 - A2	0.27	0.58	0.43	0.8	1.88
A1 - A3	0.31	0.56	0.43	0.81	1.86
A2 - A3	0.44	0.38	0.41	0.82	2.01
I1 - I2	0.39	0.46	0.42	0.82	1.94
I1 - I3	0.29	0.55	0.42	0.83	1.97
I1 - I4	0.61	0.26	0.43	0.82	1.89
I2 - I3	0.33	0.51	0.42	0.83	1.99
I2 - I4	0.69	0.26	0.47	0.82	1.73
I3 - I4	0.83	0.2	0.52	0.81	1.58
O1 - O2	0.3	0.54	0.42	0.82	1.95
O1 - O3	0.66	0.25	0.45	0.79	1.75
O1 - O4	0.45	0.37	0.41	0.82	1.99
O2 - O3	0.86	0.18	0.52	0.78	1.51
O2 - O4	0.6	0.27	0.44	0.81	1.86
O3 - O4	0.26	0.58	0.42	0.81	1.92
M1 - M2	0.42	0.43	0.43	0.82	1.93

Table 4.4: Warp coefficients for InferSent FastText

	InferSent GloVe				
Pairs	ω_c^1	ω_c^2	Ω_c	$z_{s,t}$	Q
A1 - A2	0.32	0.65	0.48	0.83	1.72
A1 - A3	0.34	0.6	0.47	0.83	1.76
A2 - A3	0.5	0.43	0.47	0.83	1.78
I1 - I2	0.4	0.48	0.44	0.84	1.9
I1 - I3	0.3	0.56	0.43	0.84	1.95
I1 - I4	0.67	0.31	0.49	0.84	1.71
I2 - I3	0.35	0.54	0.45	0.84	1.88
I2 - I4	0.74	0.3	0.52	0.83	1.61
I3 - I4	0.87	0.23	0.55	0.83	1.51
O1 - O2	0.32	0.57	0.45	0.82	1.84
O1 - O3	0.67	0.25	0.46	0.81	1.77
O1 - O4	0.49	0.4	0.44	0.83	1.87
O2 - O3	0.85	0.17	0.51	0.8	1.56
O2 - O4	0.64	0.3	0.47	0.82	1.73
O3 - O4	0.3	0.63	0.46	0.84	1.82
M1 - M2	0.42	0.43	0.43	0.85	1.98

Table 4.5: Warp coefficients for InferSent GloVe

	RoBERTa				
Pairs	ω_c^1	ω_c^2	Ω_c	$z_{s,t}$	Q
A1 - A2	0.24	0.54	0.39	0.82	2.1
A1 - A3	0.25	0.5	0.37	0.82	2.2
A2 - A3	0.38	0.32	0.35	0.83	2.39
I1 - I2	0.34	0.42	0.38	0.84	2.2
I1 - I3	0.24	0.49	0.37	0.83	2.26
I1 - I4	0.55	0.22	0.39	0.83	2.14
I2 - I3	0.29	0.47	0.38	0.84	2.2
I2 - I4	0.63	0.21	0.42	0.83	1.96
I3 - I4	0.76	0.16	0.46	0.82	1.78
O1 - O2	0.25	0.48	0.36	0.83	2.27
O1 - O3	0.59	0.2	0.39	0.82	2.09
O1 - O4	0.41	0.33	0.37	0.83	2.28
O2 - O3	0.8	0.14	0.47	0.81	1.71
O2 - O4	0.55	0.23	0.39	0.83	2.1
O3 - O4	0.23	0.55	0.39	0.82	2.11
M1 - M2	0.34	0.35	0.35	0.84	2.41

Table 4.6: Warp coefficients for RoBERTa

			USE		
Pairs	ω_c^1	ω_c^2	Ω_c	$z_{s,t}$	Q
A1 - A2	0.23	0.54	0.38	0.83	2.15
A1 - A3	0.25	0.49	0.37	0.83	2.23
A2 - A3	0.38	0.32	0.35	0.84	2.41
I1 - I2	0.35	0.42	0.38	0.84	2.2
I1 - I3	0.26	0.51	0.39	0.84	2.18
I1 - I4	0.58	0.24	0.41	0.84	2.05
I2 - I3	0.3	0.48	0.39	0.84	2.16
I2 - I4	0.63	0.21	0.42	0.84	1.99
I3 - I4	0.77	0.16	0.47	0.83	1.77
O1 - O2	0.25	0.48	0.36	0.83	2.29
O1 - O3	0.61	0.21	0.41	0.84	2.05
O1 - O4	0.4	0.32	0.36	0.84	2.3
O2 - O3	0.82	0.15	0.49	0.83	1.71
O2 - O4	0.55	0.23	0.39	0.84	2.13
O3 - O4	0.24	0.55	0.4	0.84	2.11
M1 - M2	0.35	0.36	0.35	0.84	2.37

Table 4.7: Warp coefficients for USE

	Lexical Weights				
Pairs	ω_c^1	ω_c^2	Ω_c	$z_{s,t}$	Q
A1 - A2	0.34	0.68	0.51	0.83	1.63
A1 - A3	0.34	0.6	0.47	0.85	1.8
A2 - A3	0.51	0.45	0.48	0.83	1.73
I1 - I2	0.5	0.58	0.54	0.79	1.47
I1 - I3	0.29	0.54	0.41	0.75	1.81
I1 - I4	0.67	0.31	0.49	0.7	1.42
I2 - I3	0.33	0.51	0.42	0.7	1.68
I2 - I4	0.73	0.29	0.51	0.72	1.41
I3 - I4	0.89	0.24	0.56	0.68	1.2
O1 - O2	0.31	0.56	0.43	0.79	1.82
O1 - O3	0.73	0.3	0.51	0.7	1.37
O1 - O4	0.53	0.44	0.48	0.71	1.47
O2 - O3	0.91	0.21	0.56	0.7	1.25
O2 - O4	0.65	0.31	0.48	0.71	1.48
O3 - O4	0.34	0.68	0.51	0.72	1.42
M1 - M2	0.44	0.45	0.44	0.78	1.77

Table 4.8: Warp coefficients for Lexical Weights

4.2.3 Mean of Q

The quality metric Q is calculated for the 16 pairs of books. The mean and standard deviation of Q over all pairs are calculated and shown in Table 4.9. The bar plot in Figure 4.43 plots these values, indicating how well the eight encoding methods performed on these different books. The plot shows that all methods except InferSent and Lexical weights had similar performance, and Lexical Weights was by far the worst. This is consistent with the results obtained from the human raters analysis. Here is the plot.

Methods	Mean Q	Std
DeCLUTR Base	2.11	0.19
DeCLUTR Small	2.11	0.19
InferSent FastText	1.86	0.15
InferSent GloVe	1.77	0.13
DistilBERT	2.05	0.20
RoBERTa	2.14	0.19
USE	2.13	0.19
Lexical Weights	1.55	0.20

Table 4.9: Mean of Q and standard deviation

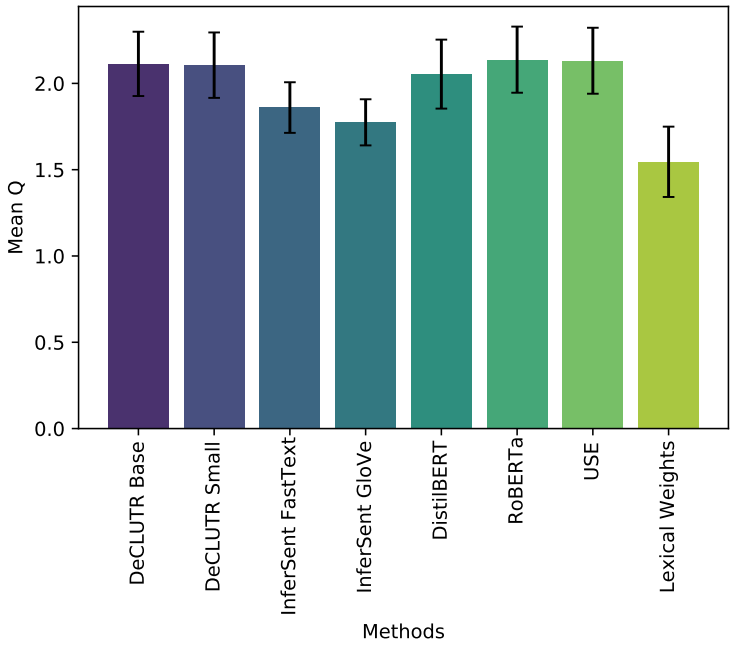


Figure 4.43: Mean of Q along with error bars

Finally, Figure 4.44 plots the two quality metrics, G and *nean* Q, against each other for all eight methods. The following observations can be made:

1. The two metrics has a monotonic relationship, indicating that they are both capturing similar information. This is important because the values of the two metrics come from completely different methods.

2. Both metrics show that five methods – DeCLUTR Base, DeCLUTR Small, DistilBERT, RoBERTa, and USE – have very similar quality on both metrics, the two InferSent methods have lower quality but are very close to each other, and the Lexical Weights method is much worse.

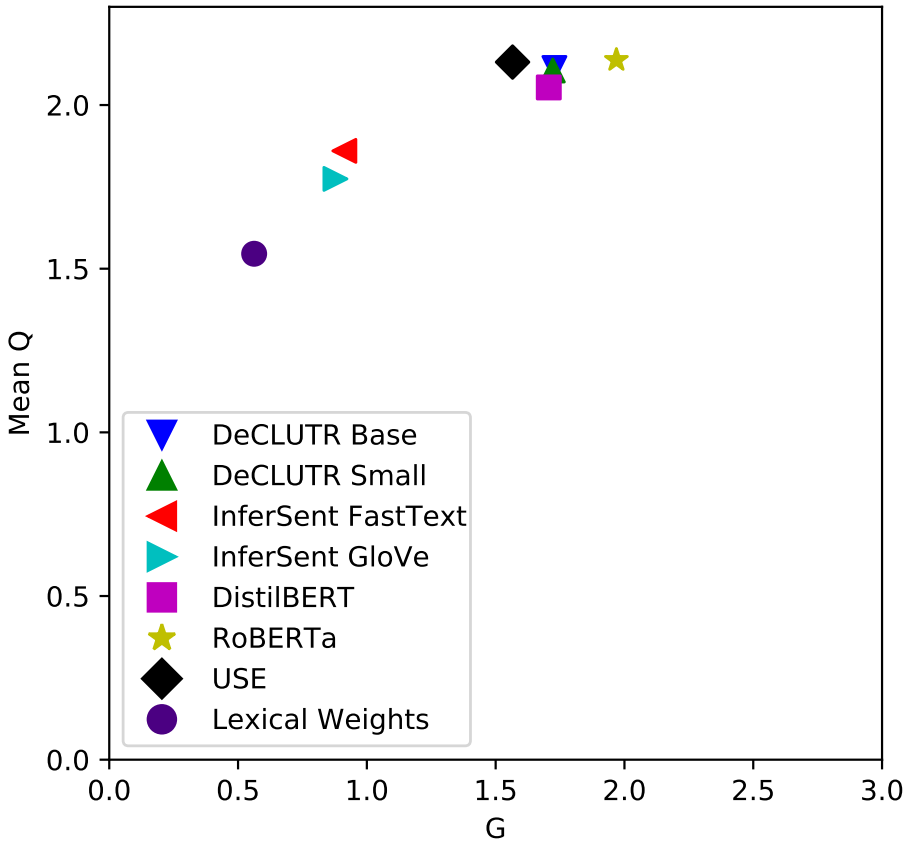


Figure 4.44: Scatter plot of Mean Q vs G

Chapter 5

Conclusions and Future Work

This thesis has analyzed various semantic embedding methods by comparing the semantic structure they infer using SSMs, time-series, correlation plots and using different methods to evaluate how well these embedding methods capture the meaning of sentences. The analysis done in Section 3.6 gives insights about the quality ofthe different methods compared in the thesis. To get a numeric value of their similarity, correlation plots are obtained. Threshold plots of the SSMs also give interesting insights on how the skeletal structure of a given document looks.

The evaluation of MSSP and LSSP gives interesting results on how the sentence pairs deemed similar/dissimilar by the embedding methods fare against evaluations by human raters. The analysis of human ratings using metric G and of multiple translations using the derived metric Q help determining which method is most consistent in capturing the semantic essence of the text.

5.1 Goals and Aims

This section reiterates the goals and aims for this thesis and concludes its findings.

1. Identifying and implementing several methods for obtaining sentence similarity, including those based on: 1) Lexical networks; 2) Transformer based models; 3) Bi-LSTM models.

2. Identifying a set of real-world corpora and documents for evaluation, including: 1) Multiple translations of the same non-English language texts; 2) Texts from different genres such as poetry, fiction, philosophy, etc.

3. Processing all the documents in the selected corpora through all the sentence similarity models to obtain similarity data from each document, including: 1) Sentence similarity matrices; 2) Time-series of similarity variation between successive sentences.

4. Defining a suite of analytical tests, characteristics, and metrics to characterize the data.

5. Applying these methods to the characterization, comparison, and analysis of the embedding methods on the target documents to determine the relative quality of the metrics.

5.2 Conclusions

The following conclusions can be derived from the studies in this thesis:

1. The results from 4.1 show insightful ways to visualize the metric of *sentence similarity* through various plots. The results show that all the embedding-based methods infer quite similar semantic structure from the same text, with methods based on the same approach showing the highest similarity.

2. The Lexical Weights methods does show some positive correlations with the structure inferred by the other methods in most cases. However, for texts with a lot of dialog and short sentences, it fares very poorly.

3. While RoBERTa stands out as a slightly better method compared to the others, USE has the attribute of showing almost the same level of high correlation with all other embedding-based methods. This suggests that USE might be inferring the most comprehensive semantic representation.

4. The results from 4.2 enable the determination of the best embedding method amongst the eight methods compared. The evaluation done by the human raters reveals that MSSP and LSSP determined by RoBERTa align most closely with a human understanding of natural language. Apart from RoBERTa, USE, DistilBERT and DeCLUTR also show high quality, while InferSent lags behind aand the Lexical Weights approach performs poorly – at least on these corpora.

5. The mean Q values obtained from the analysis of multiple translations indicate that DeCLUTR, DistilBERT, RoBERTa and USE compare very well to each other. The relative difference between their means is negligible. However, InferSent performs a bit worse, and Lexical Weights has the lowest performance.

6. Overall, Study II suggests that DeCLUTR, DistilBERT, RoBERTa and USE are all approximately of equal quality. This is consistent with the results of Study I showing that they also infer very correlated semantic structure.

5.3 Future Work

This thesis investigated seven state-of-the-art of sentence encoders on four literary books and thirteen different translations spanning across four books. To take this research further,

1. Further analysis could be done for validating the methods more with help of additional human raters.

2. Different genres of text like political speeches, medical texts, news could be analyzed.

3. This research can be extended to other domains for learning and creating superior knowledge representations.

4. The method of using Gramian Angular Field (GAF) [57] to turn time-series into images could be explored to compare translations using convolutional neural networks.

5. A new sentence hybrid encoder could be developed by comapring the strengths and weaknesses of the encoders investigated in this study.

6. Additional analysis of this semantic structure could be done with reference to the semantic content of the underlying text.

Bibliography

[1] S. Arora, Y. Liang, and T. Ma. A simple but tough-to-beat baseline for sentence embeddings. In *Proceedings of the International Conference on Learning Representations*, 2017.

[2] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information, 2017.

[3] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference, 2015.

[4] J. A. Bullinaria and J. P. Levy. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39:510–526, 2007.

[5] M. Candadai, A. Vanarase, M. Mei, and A. A. Minai. ANSWER: An unsupervised attractor network method for detecting salient words in text corpora. In *Proceedings of IJCNN 2015*, 2015.

[6] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 2017.

[7] D. Cer, Y. Yang, S.-Y. Kong, N. Hua, N. Limtiaco, R. St. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, and R. Kurzweil. Universal sentence encoder, 2018.

[8] Freddy Y. Y. Choi. Advances in domain independent linear text segmentation. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*, 2000.

[9] R. R. Choudhary. Construction and visualization of semantic spaces for domain-specific text corpora, 2021.

[10] K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16:22–29, 1990.

[11] A. Conneau and D. Kiela. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*, 2018.

[12] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes. Supervised learning of universal sentence representations from natural language inference data, 2018.

[13] A. Conneau, G. Kruszewski, G. Lample, L. Barrault, and M. Baroni. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia, 2018. Association for Computational Linguistics.

[14] S. Deerwester, S. Dumais, T. Landauer, G. Furnas, and R. Harshman. Indexing by latent semantic analysis. *Journal of American Society of Information Science*, 41:391–407, 1990.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 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, Stroudsburg, PA, USA, 2019. Association for Computational Linguistics.

[16] S. Doumit, N. Marupaka, and A. A. Minai. Thinking in prose and poetry: A semantic neural model. In *Proceedings of IJCNN 2013*, 2013.

[17] S. Doumit and A. A. Minai. Effect of associative rules on the dynamics of conceptual combination in a neurodynamical model. In *Proceedings of IJCNN 2015*, 2015.

[18] Dynamic time warping. Dynamic time warping — Wikipedia, the free encyclopedia, 2010. [Last edited; 16 October 2021, at 12:51 (UTC)].

[19] J. Giorgi, O. Nitski, B. Wang, and G. Bader. DeCLUTR: Deep contrastive learning for unsupervised textual representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 879–895, Online, August 2021. Association for Computational Linguistics.

[20] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network, 2015.

[21] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–80, 1997.

[22] J. Howard and S. Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, 2018. Association for Computational Linguistics.

[23] A. G. Huth, W. A. de Heer, T. L. Griffiths, F. E. Theunissen, and J. L. Gallant. Natural speech reveals the semantic maps that tile human cerebral cortex. *Nature*, 532:453–458, 2016.

[24] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé III. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1681–1691, Beijing, China, July 2015. Association for Computational Linguistics.

[25] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification, 2016.

[26] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland, 2014. Association for Computational Linguistics.

[27] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems*, 3, 01 2002.

[28] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler. Skip-thought vectors. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28, pages 3294–3302. Curran Associates, Inc., 2015.

[29] T. K. Koo and M. Y. Li. A guideline of selecting and reporting intraclass correlation coefficients for reliability research. 15(2):155–163, June 2016.

[30] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy. Race: Large-scale reading comprehension dataset from examinations, 2017.

[31] T. Landauer and S. Dumais. A solution to plato's problems: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104:211–240, 1997.

[32] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32, pages 1188–1196, Bejing, China, 2014. PMLR.

[33] Rensis Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 140:1–55, 1932.

[34] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[35] W. Lowe. Towards a theory of semantic space. In *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*, pages 576–581, 2001.

[36] K. Lund and K. Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behaviour Research, Methods, Instruments, & Computers*, 28:203–208, 1996.

[37] M. Mei and A. A. Minai. Divergent thinking in a neurodynamical model of ideation. In *Proceedings of IJCNN 2016*, 2016.

[38] M. Mei, Z. Ren, and A. A. Minai. Mining the temporal structure of thought from text. In *Unifying Themes in Complex Systems IX (Proceedings of ICCS'18)*, pages 291–298. Springer, 2018.

[39] M. Mei, A. Vanarase, and A. A. Minai. Chunks of thought: Finding salient semantic structures in texts. In *Proceedings of IJCNN 2014*, 2014.

[40] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In Y. Bengio and Y. LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[41] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[42] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[43] F. Pulvermüller. How neurons make meaning: brain mechanisms for embodied and abstract-symbolic semantics. *Trends in Cognitive Sciences*, 17:458–470, 2013.

[44] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019.

[45] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100,000+ questions for machine comprehension of text, 2016.

[46] N. Reimers and I. Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, page 3982–3992, 2019.

[47] M. Röder, A. Both, and A. Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408, 2015.

[48] D. L. T. Rohde and D. C. Plaut. An improved model of semantic similarity based on lexical co-occurrence. 2005.

[49] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26:43–49, 1978.

[50] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.

[51] M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[52] H. Schütze. Word space. In *Advances in Neural Information Processing Systems 5*, pages 895–902. Morgan Kaufmann, 1993.

[53] Peter D Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188, 2010.

[54] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2017.

[55] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images, 2016.

[56] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2019.

[57] Z. Wang and T. Oates. Imaging time-series to improve classification and imputation, 2015.

[58] A. Williams, N. Nangia, and S. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122. Association for Computational Linguistics, 2018.

[59] Y. Zhang, K. Han, R. Worth, and Z. Liu. Connecting concepts in the brain by mapping cortical representations of semantic relations. *Nature Communications*, 11:1877, 2020.