A Spectral Learning Based Model to Evaluate Semantic Textual Similarity

Akanksha Mehndiratta (mehndiratta.akanksha@gmail.com)
Jaypee Institute of Information Technology
Krishna Asawa (krishna.asawa@jiit.ac.in)
Jaypee Institute of Information Technology

Research Article

Keywords:

DOI: https://doi.org/

Additional Declarations: No competing interests reported.

A Spectral Learning Based Model to Evaluate Semantic Textual Similarity

Akanksha Mehndiratta[1*] and Krishna Asawa[1†]

[1]CSE/IT, Jaypee Institute of Information Technology, A-10, Sector-62, Noida, 201309, Uttar Pradesh, India.

*Corresponding author(s). E-mail(s): mehndiratta.akanksha@gmail.com;
Contributing authors: krishna.asawa@jiit.ac.in;
†These authors contributed equally to this work.

Abstract

Semantic Textual Similarity (STS) is a task in NLP that compares two sentences in a sentence-pair and scores the relationship between them using the degree of semantic equivalence. It has wide applicability in various fields. Consequently, the research around the task is constantly evolving. The demand for new as well as improved methods is endless. Numerous methods have been proposed that largely belong to either unsupervised or supervised learning approaches. The model proposed here is fairly simple and provides a fresh take on this classification problem using spectral learning. The model does not engage a large labeled corpus or lexical database like most STS supervised and unsupervised methods. Although, supervised STS methods achieve an accuracy that outperforms humans in some cases, but are often held back due to a lack of interpretation of the features instrumental in molding the decision-making process. The proposed model on the other hand generates features (latent knowledge) that are easy to ascertain and have a mathematical foundation. Given a sentence pair, the work focuses on finding latent states and variables from each sentence and performs classification by generating a similarity score. The latent variables are a result of projections learned by performing Canonical Correlation Analysis (CCA) amongst the sentence pair. To perform matching and determine the similarity score, Cosine similarity and Word Mover's Distance (WMD) are employed. The performance of the proposed model does exhibit an improvement over various sophisticated supervised techniques such as LSTM and BiLSTM.

Keywords: Spectral Learning; Semantic Textual Similarity, Canonical Correlation Analysis, Natural Language Processing, Word Mover's Distance, Latent State; Hidden Variables, Latent Variables, Hidden state

1 Introduction

Semantic Textual Similarity (STS) is a task that assigns each sentence pair a score that assesses their similarity. The central idea behind the design of STS methods is, given a sentence-pair, identification and alignment of related or semantically similar words and compute the overall similarity as the aggregation of these similarities. In most NLP applications, semantic components were either considered independently or with an understanding of their impact that is mostly superficial in the application. STS on the other hand, provides a unified structure that allows the evaluation of numerous semantic components generated by a technique on real applications. This task ushers a way for a variety of tasks such as textual entailment, machine translation, and many more. Hence the goal of this study is to design a structure that can not only generate and evaluate latent/semantic components but also elevate their part in the learning process. Latent components can play a significant role in the development of a model.

Models designed for STS largely belong to the following class of systems:

1. **Supervised Systems:** The recent developments in machine learning have been exploited by various researchers to design STS methods. The idea here is to employ a machine learning or deep-learning based model that requires an adequate amount of data to train and use the trained model to predict resulting label [1, 2]. Convolutional Neural Networks (CNN) [3], Long Short Term Memory (LSTM) and Bidirectional Long Short Term Memory (Bi-LSTM) [4, 5], are widely used models in designing semantic similarity based methods. A few hybrid models [6] have also been proposed combining two or more aforementioned models. Another breakthrough in the direction of semantic similarity research is the transformer-based models. Transformers are pre-trained models employed for the language understanding task namely ERNIE 2.0 [7], XLNet [8] to name a few. They are extremely expressive and powerful, but are also complex and hence require large computational resources. Additionally, the enhanced performance of such models is largely attributed to the size of the corpus, hence building an ideal corpus poses an immense challenge in the development of supervised STS methods. Another challenge imposed by these "black-box" models is that of interpretability which raises concerns such as adaptability, verifiability and many more.
2. **Unsupervised Systems:** Some of the designs proposed under the umbrella of unsupervised systems lean towards exploiting such simple models that were proposed before STS [9, 10]. While some designs bank upon rich

lexical databases such as paraphrase database (PPDB) [11, 12], wordnet [13], and many more to support context-dependent learning. Wieting et al [14] designed a model for encoding sentences using pre-trained embeddings to encode each word in a sentence as a vector, followed by plain averaging to generate sentence representation. Arora et al [15] on the other hand also generated word encoding in a similar manner but performed weighted averaging to generate sentence representation. Both use cosine similarity to assign a similarity metric. Although both models seem simple in design but performed better than LSTM. Such models inspire researchers to exploit, improve or propose models that are easily scalable and have the ability to process large corpuses.

In this study, we aim to map the STS classification problem onto a two-view learning setting. In this setting, there are two views (sometimes in an abstract sense) of the input data, $X = (X^{(1)}, X^{(2)})$, which co-exist and Y is the target variable. Foster [16] explored this very idea under the assumption that $X^{(1)}$, $X^{(2)}$ and Y are conditional independent on some latent state H as depicted in Figure 1.

In text and NLP based applications, this assumption is applicable quite naturally as a data-generating model is typically assumed to be a Hidden Markov Model (HMM) and HMM satisfies the multi-view assumption. Therefore in the STS task, the sentence pair given can be mapped as two views $S = (s^{(1)}, s^{(2)})$ and the H can be defined as the latent state. The study then exploits the probabilistic Canonical Correlation Analysis (CCA) model proposed by Bach and Jordan[17] to estimate this latent state H. Hence the model proposed learns the latent state from each sentence by maximizing the correlation amongst the latent state (projections) of the sentence-pair using spectral learning and uses Cosine Similarity and Word Mover's Distance (WMD) to output a similarity score .

This model does not engage any training model to perform learning nor requires a large labeled dataset. Rather, it is particularly suited for setting where labeled data is scarce. The process of learning the latent state from a sentence pair is transparent and has a mathematical grounding. Moreover, the model and latent features generated can be adapted to fit most NLP tasks, irrespective of language, that aim to capture the relationship amongst phrases or sentences that are used in a similar context such as Question Answer, Translation and many more. As the learning process exploits CCA hence it lends the model certain qualities such as linear, fast, scale-invariant and scalable. Although CCA is linear by design but the model is competitive with various non-linear supervised learning architectures such as LSTM and BiLSTM.

2 Canonical Correlation Analysis

Canonical Correlation Analysis(CCA) [18] is a widely used data analysis tool for purposes such as visualization or dimensionality reduction. It is the analysis of a linear relationship amongst two sets of variables that is captured by

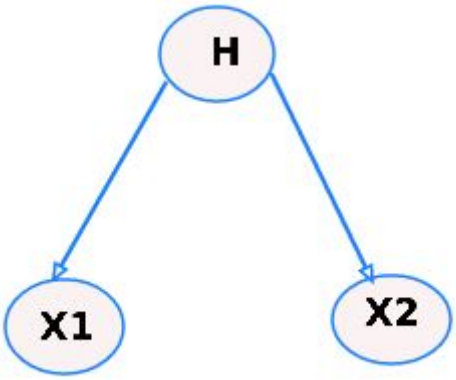

Fig. 1 Latent State Interpretation Model adapted from [17]

studying the latent variables (via projections). It is analogous to Principal Component Analysis (PCA) but defined for a multivariate output or in other words set of outputs. While PCA focuses on finding a direction of maximal covariance for input defined using multiple variables on a single variable output, CCA produces the maximal covariance for a multi variable input on a multi variable output or otherwise on two sets of a multi variable input. Unlike PCA, the input data is not standardized before performing CCA hence it is scale invariant.

Given variables $P \in \mathbb{R}^{(n \times p)}$ and $Q \in \mathbb{R}^{(n \times q)}$ corresponding to two different views and a subspace of dimension $\leq d = \min(p, q)$, CCA tries to generate linear projections onto a subspace $\in \mathbb{R}^{(d)}$ as $\Lambda_p = {U_p}^T P$ and $\Lambda_q = {U_q}^T Q$ using the optimization problem given in equation 1

$$\max_{U_p, U_q} \rho \quad \frac{{U_p}^T C_{PQ} U_q}{\sqrt{{U_p}^T C_{PP} U_p}\sqrt{{U_q}^T C_{QQ} U_q}} \tag{1}$$

where C_{PQ} represents the covariance matrix of variables P and Q. The resultant matrices $U_p \in \mathbb{R}^{(p \times d)}$ and $U_q \in \mathbb{R}^{(q \times d)}$ are composed of first d canonical pairs of direction vectors, (u_{pi}, u_{qi}). The projections when paired $(\Lambda_{pi}, \Lambda_{qj})$ are maximally correlated if $i = j$, with correlation coefficient ρ_i, and uncorrelated otherwise, hence forming a diagonal matrix of canonical correlations $\rho_d = \text{diag}([\rho_0, ..., \rho_d])$.

The optimal value for U_p, U_q and ρ are found by exploiting the eigenvalues, eigenvectors of each vector and inverse of the covariance matrices. Hence, the general eigenvalue problem can be applied to compute CCA. A major roadblock for CCA is computing the inverse but recent advances [19] in performing the inverse of a matrix using eigen decomposition or singular value decomposition (SVD) makes the computations easy even on a large scale making CCA

fast and scalable. The operation on variable P and Q is defined by equation 2

$$(\Lambda_{\mathrm{P}}, \Lambda_{\mathrm{Q}}) = CCA(P, Q) \tag{2}$$

More specifically, consider a study that wants to conduct two survey's on the same group of people. Survey 1 contains a set of features that define the financial background of a person and Set 2 contains features that define the academic background. Let there be p number of features defined in survey 1 represented as P_1, P_2, P_3... P_{p} similarly q number of features from survey 2 are represented as Q_1, Q_2, Q_3....Q_{q}. Using the relation given by equation 3 we can compute the first projection for survey 1

$$\Lambda_{\mathrm{P}} = x_1 P_1 + x_2 P_2 + x_3 P_3 + ...x_{\mathrm{p}} P_{\mathrm{p}} \tag{3}$$

And using the relation given by equation 4 we can compute the first projection for the second survey

$$\Lambda_{\mathrm{Q}} = y_1 Q_1 + y_2 Q_2 + y_3 Q_3 + ...y_{\mathrm{q}} Q_{\mathrm{q}} \tag{4}$$

Where x_1, x_2, x_3 ..... x_{p} and y_1, y_2, y_3 .... y_{q} are called weights and the outputted projection-pair is a weighted average of the original feature set. Additionally, the projection-pair is a linear transformation of feature set from survey1 and survey 2. The correlation between Λ_{P} and Λ_{Q} is maximized by the way these weights are learned. Using the residuals of the first projection-pair we compute the second pair of projections. CCA can generate maximum d such projections where d is the minimum of p and q. The projections are orthogonal i.e. they are independent of each other.

3 Model

3.1 Dataset Collection

The widely popular dataset for STS task is the SemEval STS task dataset. SemEval/*SEM is a family of workshops that conducts SemEval STS task as an annual event. From 2012 to 2017 it was the most anticipated and awaited event for STS [20–25], that attracted huge participation every year. The dataset is available publicly and for standardization has been organised in development, training and testing. It contains sentence pairs in english language although in later years a data set containing cross-lingual and multilingual sentence pairs were also included in the event. It is annotated by humans using a scale from 0 (unlikeness) to 5 (semantic resemblance) as a similarity metric. A sample from the STS dataset contains a sentence pair along with a score as shown in Table 1.

The model designed here is tested on the SemEval 2017 STS task dataset. Last in this series, it is given the name Semantic Textual Similarity Benchmark (STS-B) dataset [25]. STS-B is a corpus that comprises of datasets in english language that were used in STS tasks from 2012 to 2017 containing sentence

Table 1 A snippet of sentence-pair from the STS dataset.

	Example - 1	**Example - 2**
Sentence 1	A man is cutting a potato.	A man is slicing some potato.
Sentence 2	Two men standing in grass staring at a car.	A woman in a pink top posing with beer.
Similarity Score	4.4	0.2

pairs from newswire headlines, MT postedits and various other resources. With a total of 8628 sentence pairs the dataset is divided as training (5749 sentence-pairs), development (1500 sentence-pairs) and testing (1379 sentence-pairs) to provide a standard benchmark. Furthermore, General Language Understanding Evaluation (GLUE) benchmark [26] has listed this dataset under sentence-pair NLU tasks.

3.2 Data Preprocessing

Preprocessing input data is a vital instrument in enhancing the efficiency of a model. It transforms the raw data into a structured format or an encoding. Using python nltk each sentence in a pair goes through the following stages: tokenization, removing punctuations, replacing numbers and removing stop words. Each word is then represented using Google's word2vec while each sentence in a sentence pair is represented as a list of pre-trained embeddings appended in order of their presence in the sentence given as, $s = (w_1, w_2, ..., w_m)$, where w_i is the embedding counterpart for the i^{th} word in sentence s.

3.3 Modeling Latent State Using CCA

An interpretation of latent state was proposed by Bach and Jordan[17] on a probabilistic model as shown in figure 1 which implies that conditional on some latent state the variables are independent. The interpretation suggested that estimates of the parameters of the given model leads to canonical correlation directions.

Lemma 1 Assume model as shown in figure 1 be a probabilistic generative model. The model is defined by 5

$$\begin{aligned} H &\sim \mathcal{N}(0, I_d) \\ X^z|H &\sim \mathcal{N}(W_z H + \mu_z, \Psi_z) \end{aligned} \tag{5}$$

Here $H \in \mathbb{R}^{(d)}$ is the shared latent state that is multiple normally distributed with a mean vector and covariance matrix. $\mathcal{N}$ = Multivariate Normal distribution, I is an Identity matrix, $z = 1$ or 2 to represent the two views/random variables and a = min(a1, a2). The maximum likelihood estimates of the the model parameters $W_z \in \mathbb{R}_{(az \times a)}$, μ_z and Ψ_z can be derived in terms of the canonical correlation directions

as 6

$$\begin{aligned} \hat{W_z} &= C_{zz} U_z M_z \\ \hat{\Psi^z} &= C_{zz} - \hat{W}_z \hat{W}_{z^T} \\ \hat{\mu^z} &= \mu^z \end{aligned} \tag{6}$$

where spectral norms of the arbitrary matrices M_z, $\in \mathbb{R}_{(d \times d)}$ such that $M_1 {M_2}^T = \rho_d$, be smaller than one.

Proof Appendix A □

The lemma advocates that classical CCA explicitly divulges the shared latent state. Therefore, given a probabilistic model CCA provides a latent state generation method appropriate for use in an algorithm. Foster et al [16] exploit this latent state interpretation provided a target variable along with two views of the input data. The aim was to generate a projection of input data with reduced dimensionality but without affecting its predictive power. To conduct the study, the authors here assume that the two views of input are independent of each other conditional on a latent state, represented by fig1 called as conditional independence assumption.

Similarly, in this study given a sentence pair $S = (s_1, s_2)$ and a similarity metric Sim as the target, we consider a conditional independence model, given by 7, that implies

$$Prob(s^{(1)}, s^{(2)}|H) = Prob(s^{(1)}|H) Prob(s^{(2)}|H) \tag{7}$$

The intent is to exploit CCA to learn the latent state via projections of input data. Thus, rather than exploring the STS problem as a classification problem, here it is perceived as a two-view learning problem. The model capitalizes on the relation between $s^{(1)}$ and $s^{(2)}$, where $s^1 \in \mathbb{R}^{(n \times p)}$ and $s^2 \in \mathbb{R}^{(n \times q)}$, to implicitly learn about the target variable Sim.

Given a sentence pair, the algorithm for latent state generation using CCA is given in Algorithm1.

Algorithm 1 Algorithm for generating latent state

1: **Input:** $S_i = [s_{1i}, s_{2i}]$
2: **Output:** Λ_{1i},Λ_{2i}
3: num_of_projections = min(len(s_{1i}), len(s_{2i}))
4: cca = CCA(n_components=num_of_projections)
5: cca.fit(s_{1i},s_{2i}) ▷ Fit Model to Data
6: Λ_{1i}, Λ_{2i} = cca.transform(s_{1i},s_{2i}) ▷ Return the Projections Λ_{1i}, Λ_{2i} as Linear Transformation of s_{1i},s_{2i} respectively

The input to the algorithm is an encoding developed for each sentence in the sentence pair during data processing. A sentence encoding constitutes

of encoded words appended in the order of their occurrence in the sentence. For the sentence encoding as input, we employ SKLearn an open source Python library to perform CCA. We utilize two methods Fit(A, B) to fit the model to data and Transform(A,[B, copy]) to return latent variable pairs (via projections) from each input by maximizing the correlation amongst the two.

Only a limited number of such latent variable pairs can be outputted by CCA, the magnitude of which is restricted to the size of the smallest vector. E.g. let number of words in s_1 be 5 and in s_2 be 8 then the maximum number of projection-pair outputted is 5. Table 2 exhibits an example of projections that are identified as latent variable pairs in this model. Google's word2vec is used to interpret the projected encoding to a word.

3.4 Formulating Similarity

The projections outputted by CCA for each sentence in a pair are used to compute the similarity metric using the following:

1. Cosine similarity: It is a measure for similarity that is popular and very common. Using the generated projections, $\Lambda_1 = (p^1{}_1 , p^2{}_1 , \ldots , p^m{}_1)$ and $\Lambda_2 = (p^1{}_2 , p^2{}_2 , \ldots , p^m{}_2)$ similarity for each projection pair is computed using equation 8

$$sim(\Lambda_1, \Lambda_2) = \frac{\sum_{k=1}^{m} p^k{}_1 p^k{}_2}{\sqrt{\sum_{k=1}^{m} (p^k{}_1)^2} \sqrt{\sum_{k=1}^{m} (p^k{}_2)^2}} \tag{8}$$

   For each projection-pair we compute the cosine similarity and then perform mean to determine the similarity score.
2. Word Mover's Distance (WMD): WMD enables us to find distance between two documents even if they do not share a common word. It computes the distance in a meaningful way using embeddings derived from advanced word–to-vector encoding techniques such as Glove [27] or Word2Vec. These encodings are considered semantically superior as vectors of semantically relevant words are closest in a subspace. Harnessing this quality, WMD pairs the closest word vectors amongst the word-set of the given two documents and then computes the minimum cumulative distance.

We apply equation 9 to perform normalisation on the cosine and WMD similarity score to scale the similarity metric to 5.

$$sim_{\text{scaled}} = \frac{5 * (sim - sim_{\min})}{sim_{\max} - sim_{\min}} \tag{9}$$

Table 2 demonstrates a similarity score generated for a sentence pair using the aforementioned similarity metrics.

Table 2 A demonstration of sentence pair, its projection-pairs and similarity scores outputted by the model using cosine similarity and WMD along with the ground truth similarity score.

Sentence Pair	**Projections**	**Cosine Similarity**	**WMD**	**Ground Truth**
S_1 [A group of men play soccer on the beach.] S_2 [A group of boys are playing soccer on the beach]	Λ_1 [1^{st}: 'group', 2^{nd}: 'soccer', 3^{rd}: 'beach', 4^{th}: 'beach', 5^{th}'play'] Λ_2 [1^{st}: 'group', 2^{nd}: 'soccer', 3^{rd}: 'beach', 4^{th}: 'boys, 5^{th}: 'beach']	4.49	4.20	3.60

4 Results and Analysis

The General Language Understanding Evaluation (GLUE) benchmark [26] is a collection of diverse NLU tasks and a platform for evaluating the performance of models. In order to produce baselines for various NLP tasks they proposed a few methods and conducted experiment on several sentence-to-vector based existing approaches.

One of the proposed method contained two-layers, BiLSTM with max pooling (1500 dimension(D) per direction) and GloVe to perform word-to-vector representation resulting in a 300D embedding for each word. The sentences were encoded independently to produce a vector pair and passed the encoded pair to a classifier. A multi layer perceptron with a 512D hidden layers is employed as the classifier. Another method proposed was an extension of the previous, where a layer is added that captures the attention mechanism among all pairs of words. The attention layer is followed by a second BiLSTM with max pooling. More variants were proposed by augmenting the methods with Embeddings from Language Models (ELMo)[28] and CoVe[29]. Both encodes words to embeddings as a function of the entire input sequence. While ELMo exploits two-layer neural language model on the other hand CoVe employs a two-layer BiLSTM encoder. Lastly, each sentence in a pair is encoded using the following pre-trained sentence-to-vector based models: average bag-of-words using GloVe embeddings (CBoW) [27], Skip-Thought [30], InferSent [31], DisSent [32], and GenSen [33] and the aforementioned classifier is trained on the generated encodings. To determine the baseline results the models are trained on the STS-B dataset along with eight english datasets that comprises of single sentence tasks, similarity and paraphrase tasks and inference task. Furthermore, the authors developed a method to perform multi-task training on each model.

Pearson score, amongst the similarity score predicted by the model and the ground truth scores, is used as the evaluation criterion for STS methods. The Pearson score for the proposed models and the official task rankings for STS-B and are shown in Table 3. The Pearson value on STS-B task for models proposed in the General Language Understanding Evaluation (GLUE) benchmark have been adapted from [26]. It is evident that both the models

Table 3 Pearson's r x 100 value on STS-B task.

Training on Single Task							
Model	BiLSTM	+ ELMo	+ CoVe	+ Attn	+Attn, ELMo		+Attn, CoVe
STS-B	66.0	64.0	**67.2**	59.3	55.5		57.2
Training on Multi Task							
Model	BiLSTM	+ ELMo	+ CoVe	+ Attn	+ Attn, ELMo		+ Attn, CoVe
STS-B	70.3	67.2	64.4	72.8	**74.2**		69.8
Pre-Trained Sentence Representation Models						Proposed Models	
Model	CBow	Skip-Thought	Infersent	DisSent	GenSen	CCA (WMD)	CCA (Cosine Similarity)
STS-B	61.2	71.8	75.9	66.1	**79.3**	**76.9**	73.7

outperform the baseline methods and all its variants and are competitive with most pre-trained sentence representation models except for GenSen. GenSen is an encoder decoder based sequence-to-sequence model that employs a bidirectional GRU for encoding and decoding on a 124 Million sentence pair multitask dataset.

In past decade deep learning and deep neural networks have evolved by leaps and bounds. These architectures are advanced and multifaceted. They produce astounding results in almost every task including STS-B. Just to mention a few XLNet [8], ERNIE 2.0 [7] and many more with an accuracy of 90% and above. Details of various deep learning and transformer based models for STS-B are available on the official website of GLUE[1].

Although extremely powerful, increased model complexity on larger datasets inadvertently extends the training period. In addition, transformers and all the models proposed in GLUE benchmark are trained on labeled corpus that is diverse and large in size for enhanced classification rate. Further, in these black-box models, one cannot ascertain the features on the basis of which they accomplish such high classification rate.This not only hampers their acceptance in various field but also inhibits their adaptation on small datasets and NLP tasks for low resource languages.

On the other hand, our model relies solely on a pre-trained contextual word embedding. The model provides a structure to not only learn but harness the latent features in any role deem fit by a researcher for most NLP tasks. The model is simple ans hence can be redesigned low resource language based tasks. This study is particularly suited for a setting where it is easy to obtain unlabeled samples but labeled are scarce.

5 Conclusion

This work attempts to create an understanding of spectral learning and its application in NLP. The idea is to unearth the latent knowledge that captures the relationship amongst phrases or sentences that are used in similar context. Spectral learning not only provides us with a structure to generate the latent state but does so without the leaning on large labeled corpus. Rather,

[1] https://gluebenchmark.com/leaderboard

the model befits any corpus, irrespective of language, that is unlableled and small in size. Additionally, the development of latent knowledge has grounding in mathematics hence It can be interpreted and exploited to fit most NLP tasks. There are various applications where this setting may be applicable such as turn-taking conversation, single document summarization, multi-document summarisation etc.

Although the learning model is linear by design but the model is competitive with various sophisticated learning architectures. Further, the model is not the best compared to some but it is certainly competitive and has much to offer to tasks that extend beyond classification. It would be interesting to explore CCA as a component of large probabilistic models. Moreover, explore variants of CCA such as nonlinear, supervised, kernel and many more for variety of applications.

Declarations

Ethical Approval

Not Applicable

Availability of Supporting Data

The data used to support the findings of this study are included within the article.

Competing Interests

The authors declare that they have no financial or non-financial competing interests.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Authors' Contributions

All authors contributed equally to this work. Krishna Asawa contributed to defining the research question and the experimental design together with her Ph.D. student Akanksha Mehndiratta. Both authors have made substantial contributions to the conception of this study and the design of the presented model. Akanksha Mehndiratta under the supervision of Krishna Asawa performed data acquisition, implementation of the model, analysis of the result and wrote the first draft of the manuscript. Krishna Asawa performed a critical revision of the manuscript for important intellectual content and approved the final version of the manuscript.

Acknowledgments

We are grateful to all the authors whose work we have referenced in this article. It is their contribution that has helped in shaping the foundation of this work. I would also like to extend my gratitude to the editorial team of the Journal of Intelligent Systems and the anonymous reviewers for their comments on the article.

References

[1] Rychalska, B., Pakulska, K., Chodorowska, K., Walczak, W., Andruszkiewicz, P.: Samsung Poland NLP team at SemEval-2016 task 1: Necessity for diversity; combining recursive autoencoders, WordNet and ensemble methods to measure semantic similarity. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 602–608. Association for Computational Linguistics, San Diego, California (2016). https://doi.org/10.18653/v1/S16-1091. https://aclanthology.org/S16-1091

[2] Brychcín, T., Svoboda, L.: Uwb at semeval-2016 task 1: Semantic textual similarity using lexical, syntactic, and semantic information. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 588–594 (2016)

[3] Shao, Y.: Hcti at semeval-2017 task 1: Use convolutional neural network to evaluate semantic textual similarity. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 130–133 (2017)

[4] Tien, N.H., Le, N.M., Tomohiro, Y., Tatsuya, I.: Sentence modeling via multiple word embeddings and multi-level comparison for semantic textual similarity. Information Processing & Management **56**(6), 102090 (2019)

[5] Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks. 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), pp. 1556–1566. Association for Computational Linguistics, Beijing, China (2015). https://doi.org/10.3115/v1/P15-1150. https://aclanthology.org/P15-1150

[6] He, H., Lin, J.: Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 937–948 (2016)

[7] Sun, Y., Wang, S., Li, Y., Feng, S., Tian, H., Wu, H., Wang, H.: Ernie 2.0: A continual pre-training framework for language understanding. Proceedings of the AAAI Conference on Artificial Intelligence **34**(05), 8968–8975 (2020). https://doi.org/10.1609/aaai.v34i05.6428

[8] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding. Advances in neural information processing systems **32** (2019)

[9] Islam, A., Inkpen, D.: Semantic text similarity using corpus-based word similarity and string similarity. ACM Transactions on Knowledge Discovery from Data (TKDD) **2**(2), 1–25 (2008)

[10] Li, Y., McLean, D., Bandar, Z.A., O'shea, J.D., Crockett, K.: Sentence similarity based on semantic nets and corpus statistics. IEEE transactions on knowledge and data engineering **18**(8), 1138–1150 (2006)

[11] Sultan, M.A., Bethard, S., Sumner, T.: DLS@CU: Sentence similarity from word alignment and semantic vector composition. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pp. 148–153. Association for Computational Linguistics, Denver, Colorado (2015). https://doi.org/10.18653/v1/S15-2027. https://aclanthology.org/S15-2027

[12] Wu, H., Huang, H., Jian, P., Guo, Y., Su, C.: BIT at SemEval-2017 task 1: Using semantic information space to evaluate semantic textual similarity. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 77–84. Association for Computational Linguistics, Vancouver, Canada (2017). https://doi.org/10.18653/v1/S17-2007. https://aclanthology.org/S17-2007

[13] Wu, H., Huang, H.: Sentence similarity computational model based on information content. IEICE TRANSACTIONS on Information and Systems **99**(6), 1645–1652 (2016)

[14] Wieting, J., Bansal, M., Gimpel, K., Livescu, K.: Towards universal paraphrastic sentence embeddings. In: Bengio, Y., LeCun, Y. (eds.) 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings (2016). http://arxiv.org/abs/1511.08198

[15] Arora, S., Liang, Y., Ma, T.: A simple but tough-to-beat baseline for sentence embeddings. In: International Conference on Learning Representations (2017)

[16] Foster, D.P., Kakade, S.M., Zhang, T.: Multi-view dimensionality reduction via canonical correlation analysis (2008)

[17] Bach, F., Jordan, M.: A probabilistic interpretation of canonical correlation analysis (2005)

[18] Hotelling, H.: Relations between two sets of variates. In: Breakthroughs in Statistics, pp. 162–190. Springer, ??? (1992)

[19] Golub, G.H., Reinsch, C.: Singular value decomposition and least squares solutions. In: Linear Algebra, pp. 134–151. Springer, ??? (1971)

[20] Agirre, E., Banea, C., Cardie, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W., Lopez-Gazpio, I., Maritxalar, M., Mihalcea, R., Rigau, G., Uria, L., Wiebe, J.: SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pp. 252–263. Association for Computational Linguistics, Denver, Colorado (2015). https://doi.org/10.18653/v1/S15-2045. https://aclanthology.org/S15-2045

[21] Agirre, E., Banea, C., Cardie, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W., Mihalcea, R., Rigau, G., Wiebe, J.: SemEval-2014 task 10: Multilingual semantic textual similarity. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 81–91. Association for Computational Linguistics, Dublin, Ireland (2014). https://doi.org/10.3115/v1/S14-2010. https://aclanthology.org/S14-2010

[22] Agirre, E., Banea, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Mihalcea, R., Rigau, G., Wiebe, J.: SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 497–511. Association for Computational Linguistics, San Diego, California (2016). https://doi.org/10.18653/v1/S16-1081. https://aclanthology.org/S16-1081

[23] Agirre, E., Bos, J., Diab, M., Manandhar, S., Marton, Y., Yuret, D. (eds.): *SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012). Association for Computational Linguistics, Montréal, Canada (2012). https://aclanthology.org/S12-1000

[24] Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W.: *SEM 2013 shared task: Semantic textual similarity. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, pp. 32–43. Association for Computational Linguistics, Atlanta, Georgia, USA (2013). https://aclanthology.org/S13-1004

[25] Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., Specia, L.: SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 1–14. Association for Computational Linguistics, Vancouver, Canada (2017). https://doi.org/10.18653/v1/S17-2001. https://aclanthology.org/S17-2001

[26] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.: GLUE: A multi-task benchmark and analysis platform for natural language understanding. In: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks For NLP, pp. 353–355. Association for Computational Linguistics, Brussels, Belgium (2018). https://doi.org/10.18653/v1/W18-5446. https://aclanthology.org/W18-5446

[27] Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)

[28] Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 2227–2237. Association for Computational Linguistics, New Orleans, Louisiana (2018). https://doi.org/10.18653/v1/N18-1202. https://aclanthology.org/N18-1202

[29] McCann, B., Bradbury, J., Xiong, C., Socher, R.: Learned in translation: Contextualized word vectors. Advances in neural information processing systems **30** (2017)

[30] Kiros, R., Zhu, Y., Salakhutdinov, R.R., Zemel, R., Urtasun, R., Torralba, A., Fidler, S.: Skip-thought vectors. Advances in neural information processing systems **28** (2015)

[31] Conneau, A., Kiela, D., Schwenk, H., Barrault, L., Bordes, A.: Supervised learning of universal sentence representations from natural language inference data. arXiv preprint arXiv:1705.02364 (2017)

[32] Nie, A., Bennett, E.D., Goodman, N.D.: Dissent: Sentence representation learning from explicit discourse relations. arXiv preprint arXiv:1710.04334 (2017)

[33] Subramanian, S., Trischler, A., Bengio, Y., Pal, C.J.: Learning general purpose distributed sentence representations via large scale multi-task learning. arXiv preprint arXiv:1804.00079 (2018)

Appendix A Proof For Theorm 1

Given the two views $X = (X^{(1)}, X^{(2)})$, we define

$$W = \begin{pmatrix} W_1 \\ W_2 \end{pmatrix}$$

and

$$\Psi = \begin{pmatrix} \Psi_1 & 0 \\ 0 & \Psi_2 \end{pmatrix}$$

under the linear probabilistic model (3) the marginal mean and covariance matrix are

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$$

and $C = W\ W^T + \Psi$, therefore, the negative log-likelihood of the data similar to the proof in [17], can be written as

$$l_1 = \frac{n(a_1 + a_2)}{2} log 2\pi + \frac{n}{2} log \mathsf{C} + \frac{1}{2} \sum_{j=1}^{n} trC^{-1}(X_j - \mu)(X_j - \mu)^T$$

$$= \frac{n(a_1 + a_2)}{2} log 2\pi + \frac{n}{2} log \mathsf{C} + \frac{n}{2} trC^{-1}\tilde{C} + \frac{n}{2}(\tilde{\mu} - \mu)C^{-1}(\tilde{\mu} - \mu)^T$$

Here $\tilde{x}$ means sample mean. Let us maximize l_1 with respect to μ, this results a maximum at $\mu = \tilde{\mu}$ .Plugging this value in the log likelihood results in

$$l_1 = \frac{n(a_1 + a_2)}{2} log 2\pi + \frac{n}{2} log \mathsf{C} + \frac{n}{2} trC^{-1}\tilde{C}$$

The rest of the proof follows immediately along the line of proof in[17].