Spectral Learning of Semantic Units in a Sentence Pair to Evaluate Semantic Textual Similarity

Akanksha Mehndiratta(✉) and Krishna Asawa

Jaypee Institute of Information Technology, Noida, India
mehndiratta.akanksha@gmail.com, krishna.asawa@jiit.ac.in

Abstract. Semantic Textual Similarity (STS) measures the degree of semantic equivalence between two snippets of text. It has applicability in a variety of Natural Language Processing (NLP) tasks. Due to the wide application range of STS in many fields, there is a constant demand for new methods as well as improvement in current methods. A surge of unsupervised and supervised systems has been proposed in this field but they pose a limitation in terms of scale. The restraints are caused either by the complex, non-linear sophisticated supervised learning models or by unsupervised learning models that employ a lexical database for word alignment. The model proposed here provides a spectral learning-based approach that is linear, scale-invariant, scalable, and fairly simple. The work focuses on finding semantic similarity by identifying semantic components from both the sentences that maximize the correlation amongst the sentence pair. We introduce an approach based on Canonical Correlation Analysis (CCA), using cosine similarity and Word Mover's Distance (WMD) as a calculation metric. The model performs at par with sophisticated supervised techniques such as LSTM and BiLSTM and adds a layer of semantic components that can contribute vividly to NLP tasks.

Keywords: Semantic Textual Similarity · Natural Language Processing · Spectral learning · Semantic units · Canonical Correlation Analysis · Word Mover's Distance

1 Introduction

Semantic Textual Similarity (STS) determines the similarity between two pieces of texts. It has applicability in a variety of Natural Language Processing (NLP) tasks including textual entailment, paraphrase, machine translation, and many more. It aims at providing a uniform structure for generation and evaluation of various semantic components that, conventionally, were considered independently and with a superficial understanding of their impact in various NLP applications.

The SemEval STS task is an annual event held as part of the SemEval/*SEM family of workshops. It was one of the most awaited events for STS from 2012 to

L. Bellatreche et al. (Eds.): BDA 2020, LNCS 12581, pp. 49–59, 2020.
https://doi.org/10.1007/978-3-030-66665-1_4

2017 [1–6], that attracted a large number of teams every year for participation. The dataset is available publicly by the organizers containing up to 16000 sentence pairs for training and testing that is annotated by humans with a rating between 0–5 with 0 indicating highly dissimilar and 5 being highly similar.

Generally, the techniques under the umbrella of STS can be classified into the following two categories:

1. **Supervised Systems:** The techniques designed in this category generate results after conducting training with an adequate amount of data using a machine learning or deep-learning based model [9,10]. Deep learning has gained a lot of popularity in NLP tasks. They are extremely powerful and expressive but are also complex and non-linear. The increased model complexity makes such models much slower to train on larger datasets.
2. **Unsupervised Systems:** To our surprise, the basic approach of plain averaging [11] and weighted averaging [12] word vectors to represent a sentence and computing the degree of similarity as the cosine distance has outperformed LSTM based techniques. Examples like these strengthen the researchers that lean towards the simpler side and exploit techniques that have the potential to process a large amount of text and are scalable instead of increased model complexity. Some of the techniques under this category may have been proposed even before the STS shared task [19,20] whiles some during. Some of these techniques usually rely on a lexical database such as paraphrase database (PPDB) [7,8], wordnet [21], etc. to determine contextual dependencies amongst words.

The technique that is proposed in this study is based on spectral learning and is fairly simple. The idea behind the approach stems from the fact that the semantically equivalent sentences are dependent on a similar context. Hence goal here is to identify semantic components that can be utilized to frame context from both the sentences. To achieve that we propose a model that identifies such semantic units from a sentence based on its correlation from words of another sentence. The method proposed in the study, a spectral learning-based approach for measuring the strength of similarity amongst two sentences based on Canonical Correlation Analysis (CCA) [22] uses cosine similarity and Word Mover's Distance (WMD) as calculation metric. The model is fast, scalable, and scale-invariant. Also, the model is linear and have the potential to perform at par with the non-linear supervised learning architectures such as such as LSTM and BiLSTM. It also adds another layer by identifying semantic components from both the sentences based on their correlation. These components can help develop a deeper level of language understanding.

2 Canonical Correlation Analysis

Given two sets of variables, canonical correlation is the analysis of a linear relationship amongst the variables. The linear relation is captured by studying the latent variables (variables that are not observed directly but inferred) that represent the direct variables. It is similar to correlation analysis but multivariate. In the statistical analysis, the term can be found in multivariate discriminant analysis and multiple regression analysis. It is an analog to Principal Component Analysis (PCA), for a set of outputs. PCA generates a direction of maximal covariance amongst the elements of a matrix, in other words for a multivariate input on a single output, whereas CCA generates a direction of maximal covariance amongst the elements of a pair of matrices, in other words for a multivariate input on a multivariate output.

Consider two random multivariable x and y. Given $\mathrm{C_{xx}}$, $\mathrm{C_{yy}}$, $\mathrm{C_{yx}}$ that represents the within-sets and between-sets covariance matrix of x and y and $\mathrm{C_{xy}}$ is a transpose of $\mathrm{C_{yx}}$, CCA tries to generate projections $\mathrm{CV_1}$ and $\mathrm{CV_2}$, a pair of linear transformations, using the optimization problem given by Eq. 1.

$$\max_{CV_1, CV_2} \quad \frac{CV_1^T C_{\mathrm{xy}} CV_2}{\sqrt{CV_1^T C_{\mathrm{xx}} CV_1}\sqrt{CV_2^T C_{\mathrm{yy}} CV_2}} \tag{1}$$

Given x and y, the canonical correlations are found by exploiting the eigenvalue equations. Here the eigenvalues are the squared canonical correlations and the eigenvectors are the normalized canonical correlation basis vectors. Other than eigenvalues and eigenvectors, another integral piece for solving Eq. 1 is to compute the inverse of the covariance matrices. CCA utilizes Singular value decomposition (SVD) or eigen decomposition for performing the inverse of a matrix. Recent advances [24] have facilitated such problems with a boost on a larger scale. This boost is what makes CCA fast and scalable.

More specifically, consider a group of people that have been selected to participate in two different surveys. To determine the correlation between the two surveys CCA tries to project a linear transformation of the questions from survey 1 and questions from survey 2 that maximizes the correlation between the projections. CCA terminology identifies the questions in the survey as the variables and the projections as variates. Hence the variates are a linear transformation or a weighted average of the original variables. Let the questions in survey 1 be represented as $\mathrm{x_1, x_2, x_3 .... x_n}$ similarly questions in survey 2 are represented as $\mathrm{y_1, y_2, y_3 .... y_m}$. The first variate for survey 1 is generated using the relation given by Eq. 2.

$$CV_1 = a_1x_1 + a_2x_2 + a_3x_3 + ...a_{\mathrm{n}}x_{\mathrm{n}} \tag{2}$$

And the first variate for survey 2 is generated using the relation given by Eq. 3.

$$CV_1 = b_1y_1 + b_2y_2 + b_3y_3 + ...b_{\mathrm{m}}y_{\mathrm{m}} \tag{3}$$

Where $\mathrm{a_1, a_2, a_3 ..... a_n}$ and $\mathrm{b_1, b_2, b_3 .... b_m}$ are weights that are generated in such a way that it maximizes the correlation between $\mathrm{CV_1}$ and $\mathrm{CV_2}$. CCA can generate the second pair of variates using the residuals of the first pair of variates and many more in such a way that the variates are independent of each other i.e. the projections are orthogonal.

When applying CCA the following fundaments are needed to be taken care of:

1. Determine the minimum number of variates pair be generated.
2. Analyze the significance of a variate from two perspectives – one being the magnitude of relatedness between the variate and the original variable from which it was transformed and the magnitude of relatedness between the corresponding variate pair.

2.1 CCA for Computing Semantic Units

Given two views $\mathrm{X} = (\mathrm{X}^{(1)}, \mathrm{X}^{(2)})$ of the input data and a target variable Y of interest, Foster [23] exploits CCA to generate a projection of X that reduces the dimensionality without compromising on its predictive power. Authors assume, as represented by Eq. 4, that the views are independent of each other conditioned on a hidden state h, i.e.

$$P(X^{(1)}, X^{(2)}|h) = P(X^{(1)}|h)P(X^{(2)}|h) \tag{4}$$

Here CCA utilizes the multi-view nature of data to perform dimensionality reduction.

STS is an estimate of the prospective of a candidate sentence to be considered as a semantic counterpart of another sentence. Measuring text similarity has had a long-serving and contributed widely in applications designed for text processing and related areas. Text similarity has been used for machine translation, text summarization, semantic search, word sense disambiguation, and many more. While making such an assessment is trivial for humans, making algorithms and computational models that mimic human-level performance poses a challenge. Consequently, natural language processing applications such as generative models typically assume a Hidden Markov Model (HMM) as a learning function. HMM also indicates a multi-view nature. Hence, two sentences that have a semantic unit(s) c with each other provide two natural views and CCA can be capitalized, as shown in Eq. 5, to extract this relationship.

$$P(S_1, S_2|c) = P(S_1|c)P(S_2|c) \tag{5}$$

Where S_1 and S_2 mean sentence one and sentence two that are supposed to have some semantic unit(s) c. It has been discussed in the previous section that CCA is fast and scalable. Also, CCA neither requires all the views to be of a fixed length nor have the views to be of the same length; hence it is scale-invariant for the observations.

3 Model

3.1 Data Collection

We test our model in three textual similarity tasks. All three of which were published in SemEval semantic textual similarity (STS) tasks (2012–2017). The first dataset considered for experimenting was from SemEval -2017 Task 1 [6], an ongoing series of evaluations of computational semantic analysis systems with a total of 250 sentence pairs. Another data set was SemEval textual similarity dataset 2012 with the name “OnWN” [4]. The sentence pair in the dataset is generated from the Ontonotes and its corresponding wordnet definition. Lastly, SemEval textual similarity dataset 2014 named “headlines” [2] that contains sentences taken from news headlines. Both the datasets have 750 sentence pairs. In all the three datasets a sentence pair is accompanied with a rating between 0–5 with 0 indicating highly dissimilar and 5 being highly similar. An example of a sentence pair available in the SemEval semantic textual similarity (STS) task is shown in Table 1.

Table 1. A sample demonstration of sentence pair available in the SemEval semantic textual similarity (STS) task publically available dataset.

	Example - 1	Example - 2
Sentence 1	Birdie is washing itself in the water basin	The young lady enjoys listening to the guitar
Sentence 2	The bird is bathing in the sink	The woman is playing the violin
Similarity Score	5 (The two sentences mean the same thing hence are completely equivalent)	1 (The two sentences may be around the same topic but are not equivalent)

3.2 Data Preprocessing

It is important to pre-process the input data to improve the learning and elevate the performance of the model. Before running the similarity algorithm the data collected is pre-processed based on the following steps.

1. **Tokenization -** Processing one sentence at a time from the dataset the sentence is broken into a list of words that were essential for creating word embeddings.
2. **Removing punctuations -** Punctuations, exclamations, and other marks are removed from the sentence using regular expression and replaced with empty strings as there is no vector representation available for such marks.
3. **Replacing numbers -** The numerical values are converted to their corresponding words, which can then be represented as embeddings.

4. **Removing stop words -** In this step the stop words from each sentence are removed. A stop word is a most commonly used word (such as “the”, “a”, “an”, “in”) that do not add any valuable semantic information to our sentence. The used list of stop words is obtained from the nltk package in python.

3.3 Identifying Semantic Units

Our contribution to the STS task adds another layer by identifying semantic units in a sentence. These units are identified based on their correlation with the semantic units identified in the paired sentence. Each sentence $\mathrm{s_i}$ is represented as a list of the word2vec embedding, where each word is represented in the m -dimensional space using Google’s word2vec. $\mathrm{s_i} = (\mathrm{w_{i1}, w_{i2}, ..., w_{im}})$, i = 1, 2, ..., m, where each element is the embedding counterpart of its corresponding word. Given two sentences $\mathrm{s_i}$ and $\mathrm{s_j}$, CCA projects variates as linear transformation of $\mathrm{s_i}$ and $\mathrm{s_j}$. The number of projections to be generated is limited to the length, i.e. no. of words, of the smallest vector between $\mathrm{s_i}$ and $\mathrm{s_j}$. E.g. if the length of $\mathrm{s_i}$ and $\mathrm{s_j}$ is 8 and 5 respectively, the maximum number of correlation variates outputted is 5. Conventionally, word vectors were considered independently and with a superficial understanding of their impact in various NLP applications. But these components obtained can contribute vividly in an NLP task. A sample of semantic units identified on a sentence pair is shown in Table 2.

Table 2. A sample of semantic units identified on a sentence pair in the SemEval dataset.

Sentence	The group is eating while taking in a breath-taking view.	A group of people take a look at an unusual tree.
Pre-processed tokens	[‘group’, ‘eating’, ‘taking’, ‘breath-taking’, ‘view’]	[‘group’, ‘people’, ‘take’, ‘look’, ‘unusual’, ‘tree’]
Correlation variates	[‘group’, ‘taking’, ‘view’, ‘breathtaking’, ‘people’]	[‘group’, ‘take’, ‘look’, ‘unusual, ‘people’]

3.4 Formulating Similarity

The correlation variates projected by CCA are used to generate a new representation for each sentence $\mathrm{s_i}$ as a list of the word2vec vectors, $\mathrm{s_i} = (\mathrm{w_{i1}, w_{i2}, ..., w_{in}})$, i = 1, 2, ..., n, where each element is the Google’s word2vec word embedding of its corresponding variate identified by CCA.

Given a range of variate pairs, there are two ways of generating a similarity score for sentence $\mathrm{s_i}$ and $\mathrm{s_j}$:

1. Cosine similarity: It is a very common and popular measure for similarity. Given a pair of sentence represented as $\mathrm{s_i} = (\mathrm{w_{i1}}, \mathrm{w_{i2}}, ..., \mathrm{w_{im}})$ and $\mathrm{s_j} = (\mathrm{w_{j1}}, \mathrm{w_{j2}}, ..., \mathrm{w_{jm}})$, cosine similarity measure is defined as Eq. 6

$$sim(s_{\mathrm{i}}, s_{\mathrm{j}}) = \frac{\sum_{k=1}^{m} w_{\mathrm{ik}} w_{\mathrm{jk}}}{\sqrt{\sum_{k=1}^{m} w_{\mathrm{ik}}^2}\sqrt{\sum_{k=1}^{m} w_{\mathrm{jk}}^2}} \tag{6}$$

   Similarity score is calculated by computing the mean of cosine similarity for each of these variate pairs.
2. Word Mover's Distance (WMD): WMD is a method that allows us to assess the "distance" between two documents in a meaningful way. It harnesses the results from advanced word –embedding generation techniques like Glove [13] or Word2Vec as embeddings generated from these techniques are semantically superior. Also, with embeddings generated using Word2Vec or Glove it is believed that semantically relevant words should have similar vectors. Let $\mathrm{T} = (\mathrm{t_1}, \mathrm{t_2}, ..., \mathrm{t_m})$ represents a set with m different words from a document A. Similarly $\mathrm{P} = (\mathrm{p_1}, \mathrm{p_2}, ..., \mathrm{p_n})$ represents a set with n different terms from a document B. The minimum cumulative distance traveled amongst the word cloud of the text document A and B becomes the distance between them.

A min-max normalization, given in Eq. 7, is applied on the similarity score generated by cosine similarity or WMD to scale the output similarity score to 5.

$$x_{\text{scaled}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{7}$$

4 Results and Analysis

The key evaluation criterion is the Pearson's coefficient between the predicted scores and the ground-truth scores. The results from the "OnWN" and "Headlines" dataset published in SemEval semantic textual similarity (STS) task 2012 and 2014 respectively is shown in Table 3. The first three results are from the official task rankings followed by seven models proposed by Weintings [11]. The last two column indicate the result from the model proposed with cosine similarity and WMD respectively. The dataset published in SemEval semantic textual similarity (STS) tasks 2017 is identified as Semantic Textual Similarity Benchmark (STS-B) by the General Language Understanding Evaluation (GLUE) benchmark [16]. The results of the official task rankings for the task STS-B are shown in Table 4. Table 5 indicate the result from the model proposed with cosine similarity and WMD respectively. Since the advent of GLUE, a lot models have been proposed for the STS-B task, such as XLNet [17], ERNIE 2.0 [18] and many more, details of these models are available on the official website of GLUE[1], that produces result above 90% in STS-B task. But the increased model complexity

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

Table 3. Results on SemEval -2012 and 2014 textual similarity dataset (Pearson's r x 100).

Dataset	50%	75%	Max	PP	proj	DAN	RNN	iRNN	LSTM (output gate)	LSTM	CCA (CoSim)	CCA (WMD)
OnWN	60.8	65.9	72.7	70.6	70.1	65.9	63.1	70.1	65.2	56.4	60.5	37.1
Headlines	67.1	75.4	78.4	69.7	70.8	69.2	57.5	70.2	57.5	50.9	62.5	55.8

Table 4. Results on STS-B task from GLUE Benchmark (Pearson's r x 100).

Model	STS-B
Single task training	
BiLSTM	66.0
+ELMo [14]	64.0
+CoVe [15]	67.2
+Attn	59.3
+Attn, ELMo	55.5
+ATTN, CoVe	57.2
Multi-task training	
BiLSTM	70.3
+ELMo	67.2
+CoVe	64.4
+Attn	72.8
+Attn, ELMo	74.2
+ATTN, CoVe	69.8
Pre-trained sentence representation models	
CBow	61.2
Skip-Thought	71.8
Infersent	75.9
DisSent	66.1
GenSen	79.3

Note. Adapted from "Glue: A multi-task benchmark and analysis platform for natural language understanding" by Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S. R.(2019), In: International Conference on Learning Representations (ICLR).

makes such models much slower to train on larger datasets. The work here focuses on finding semantic similarity by identifying semantic components using an approach that is linear, scale-invariant, scalable, and fairly simple.

Table 5. Results of proposed spectral learning-based model on the SemEval 2017 dataset (Pearson's r x 100).

Model	STS-B
CCA (Cosine similarity)	73.7
CCA (WMD)	76.9

5 Conclusion

We proposed a spectral learning based model namely CCA using cosine Similarity and WMD, and compared the model on three different datasets with various other competitive models. The model proposed utilizes a scalable algorithm hence it can be included in any research that is inclined towards textual analysis. With an added bonus that the model is simple, fast and scale-invariant it can be an easy fit for a study.

Another important take from this study is the identification of semantic units. The first step in any NLP task is providing a uniform structure for generation and evaluation of various semantic units that, conventionally, were considered independently and with a superficial understanding of their impact. Such components can help in understanding the development of context over sentence in a document, user reviews, question-answer and dialog session.

Even though our model couldn't give best results it still performed better than some models and gave competitive results for others, which shows that there is a great scope for improvement. One of the limitations of the model is its inability to identify semantic units larger than a word for instance, a phrase. It will also be interesting to develop a model that is a combination of this spectral model with a supervised or an unsupervised model. On further improvement the model will be helpful in various ways and can be used in applications such as document summarization, word sense disambiguation, short answer grading, information retrieval and extraction, etc.

References

1. Agirre, E., et al.: 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, June 2015
2. Agirre, E., et al.: 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, August 2014
3. Agirre, E., et al.: SemEval-2016 task 1: semantic textual similarity, monolingual and cross-lingual evaluation. In: SemEval 2016, 10th International Workshop on Semantic Evaluation, San Diego, CA, Stroudsburg (PA), pp. 497–511. Association for Computational Linguistics (2016)

4. Agirre, E., Bos, J., Diab, M., Manandhar, S., Marton, Y., Yuret, D.: *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), pp. 385–393. Association for Computational Linguistics (2012)
5. 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, June 2013
6. Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., Specia, L.: SemEval-2017 task 1: semantic textual similarity-multilingual and cross-lingual focused evaluation. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval 2017), Vancouver, Canada, pp. 1–14. Association for Computational Linguistics (2017)
7. 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, pp. 148–153. Association for Computational Linguistics, June 2015
8. Wu, H., Huang, H.Y., 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, August 2017
9. 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, June 2016
10. 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. Association for Computational Linguistics, June 2016
11. Wieting, J., Bansal, M., Gimpel, K., Livescu, K.: Towards universal paraphrastic sentence embeddings. In: International Conference on Learning Representations (ICLR) (2015)
12. Arora, S., Liang, Y., Ma, T.: A simple but tough-to-beat baseline for sentence embeddings. In: International Conference on Learning Representations (ICLR) (2016)
13. 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, pp. 1532–1543, October 2014
14. Peters, M.E., et al.: Deep contextualized word representations. In: Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (2018)
15. McCann, B., Bradbury, J., Xiong, C., Socher, R.: Learned in translation: contextualized word vectors. In: Advances in Neural Information Processing Systems, pp. 6297–6308 (2017)
16. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: GLUE: a multi-task benchmark and analysis platform for natural language understanding. In: International Conference on Learning Representations (ICLR) (2019)

17. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V.: XLNet: generalized autoregressive pretraining for language understanding. In: Advances in Neural Information Processing Systems, pp. 5753–5763 (2019)
18. Sun, Y., et al.: ERNIE 2.0: a continual pre-training framework for language understanding. In: AAAI, pp. 8968–8975 (2020)
19. Islam, A., Inkpen, D.: Semantic text similarity using corpus-based word similarity and string similarity. ACM Trans. Knowl. Discov. Data (TKDD) **2**(2), 1–25 (2008)
20. Li, Y., McLean, D., Bandar, Z.A., O'shea, J.D., Crockett, K.: Sentence similarity based on semantic nets and corpus statistics. IEEE Trans. Knowl. Data Eng. **18**(8), 1138–1150 (2006)
21. Wu, H., Huang, H.: Sentence similarity computational model based on information content. IEICE Trans. Inf. Syst. **99**(6), 1645–1652 (2016)
22. Hotelling, H.: Canonical correlation analysis (CCA). J. Educ. Psychol. **10** (1935)
23. Foster, D.P., Kakade, S.M., Zhang, T.: Multi-view dimensionality reduction via canonical correlation analysis (2008)
24. Golub, G.H., Reinsch, C.: Singular value decomposition and least squares solutions. In: Bauer, F.L. (ed.) Linear Algebra, pp. 134–151. Springer, Heidelberg (1971). https://doi.org/10.1007/978-3-662-39778-7_10