Language Model Metrics and Procrustes Analysis for Improved Vector Transformation of NLP Embeddings

Thomas Conley
University of Colorado
Colorado Springs
1420 Austin Bluffs Pkwy
Colorado Springs, CO, USA
tconley@uccs.edu

Jugal Kalita
University of Colorado
Colorado Springs
1420 Austin Bluffs Pkwy
Colorado Springs, CO, USA
jkalita@uccs.edu

Abstract

Artificial Neural networks are mathematical models at their core. This truism presents some fundamental difficulty when networks are tasked with Natural Language Processing. A key problem lies in measuring the similarity or distance among vectors in NLP embedding space, since the mathematical concept of distance does not always agree with the linguistic concept. We suggest that the best way to measure linguistic distance among vectors is by employing the Language Model (LM) that created them. We introduce Language Model Distance (LMD) for measuring accuracy of vector transformations based on the Distributional Hypothesis (LMD Accuracy). We show the efficacy of this metric by applying it to a simple neural network learning the Procrustes algorithm for bilingual word mapping.

1 Introduction

The Distributional Hypothesis (Firth, 1961) inspired the development of embeddings that capture the meaning of language based on how words co-occur with each other (Mikolov et al., 2013a). Natural Language Processing relies heavily on these high dimensional vectors to represent words, phrases, sentences or documents, in a form that can be processed by deep neural networks which were originally designed for tasks related to computer vision. Input embeddings are transformed by network layers into output vectors which represent solutions to many NLP tasks (Ruder et al., 2019).

In order to learn these transformations, a network must be able to calculate the difference between predicted vectors and actual word vectors. This distance calculation is a crucial part of measuring loss, and performing back-propagation. These core functions of neural networks have primarily relied on mathematical processes without regard to linguistic principles. We demonstrate that NLP embedding transformation is better measured using linguistic similarity functions rooted in knowledge of languages rather than concepts such as Euclidean or angular distance, which assumes vectors to be “physical” objects.

1.1 Procrustes Analysis

Matrix transformation of vector spaces has been accomplished using Generalized Procrustes Analysis (GPA), ever since a computationally viable solution was devised (Gower, 1975). In particular, GPA has been used to great effect in geo-spatial shape manipulation (Duta, 2015; Crosilla et al., 2019) and qualitative data analysis (Maurício et al., 2016).

Shapes are represented by a series of landmark points in 2 or 3 dimensions. And in survey research, qualitative opinion data is represented by a Likert scale (Likert, 1932), occupying low dimensional space. In both cases, the vector spaces must be realigned and resized for meaningful comparison. Although these fields seem to differ, they use data structures that share characteristics with Natural Language Processing.

The orthogonal Procrustes algorithm produces an optimal transformation matrix $R$ for mapping one vector space to another and appears to be useful in converting vector spaces for NLP tasks such as bilingual word mapping (Kementchedjhieva et al., 2018).

1.2 Procrustes Analysis for NLP tasks

Can a neural network learn to do Procrustes transformation? The answer, yes, should be non-controversial, since every neural network performs tensor transformation of input to output. However, tasks which require nuanced understanding of the meaning of words, such bilingual word mapping, are particularly difficult. Although there is some success when massive amounts of text are available for training, the problem is more acute when
resources for learning are scarce, as in machine translation of under resourced languages.

The difficulty with vector transformations in NLP is based on the nature of the data. NLP transformations by neural networks use distance measurements designed to work in $L_p$ space. This implies numerical data. We show that such calculations of distance and accuracy are not as effective as measurements based on language models.

1.3 Image data and language data

We consider image data as raw data with physical dimensionality where, each dimension in a vector can be considered similar in measurement and meaning. As such, this data occupies $L_p$ space; where vectors can be added together or multiplied by scalars without loss of their inherent meaning. For example, a vector representing a pixel is measured the same way, and has the same meaning, regardless of where it is in the image.

Thus, distance measurement among image vectors can use $L_p$ norm or trigonometric calculations such as cosine distance. One specific kind of euclidean distance measurement is called Procrustes Distance and is the basis of Procrustes Analysis (Crosilla et al., 2019).

In NLP, distance measurement is less meaningful when it is based on Euclidean axioms rather than linguistic principles. Distance is the basis of error calculations and back-propagation, and so, the ability to calculate the derivative of these functions is essential for classic stochastic gradient descent (SGD) which is employed by neural networks today. Although there has been some research in non-differentiable losses (Engilberge et al., 2019) the mathematical requirements for these functions are not always suitable for NLP.

As opposed to raw data, feature data consists of vectors in which each dimension may have disparate meaning and measurement. Feature data does not exist in $L_p$ space, and therefore measures of distance that rely on $L_p$ norm or trigonometric calculations may not be meaningful. We consider NLP embeddings to be feature data, although they share some characteristics with raw data.

2 Language Models and Data

As in raw data, NLP vectors dimensions typically share values that are treated similarly and are thus undifferentiated in a sense. This seems to contradict the assertion that each NLP embedding dimension has a specific unique meaning like feature data. Instead, the meaning of a dimension is more like probability, representing how often a word is used with a particular meaning, rather than the actual meaning of the word.

Vectors with dimensions that differ in meaning, as in NLP embeddings, cannot be used with typical spatial measurements such as $L_p$ norm and cosine distance. We contend that NLP vector distance can best be measured by the language models which represent the vectors. Therefore, we seek to replace mathematic calculations with predictions from language models. We simply rely on the language model itself to provide a distance measurement for our custom metric.

In this research, we use the Word2Vec model (Mikolov et al., 2013b) to produce a custom bilingual word mapping dataset. This dataset, combined with the GenSim model of keyed vectors (Řehůrek and Sojka, 2010), provides a distributional distance measurement based on word movers distance (Kusner et al., 2015).

Our neural network is a simple Multilayer Perceptron (MLP) which accepts Spanish word vectors as input and predicts English word vectors. This simple model was chosen because it is analogous to any layer found in innumerable, more complex, neural networks. Showing improved efficacy in this model should demonstrate improvement in any NLP task.

![Figure 1: Illustration of the Distributional Hypothesis and Language Model Distance. The accuracy of predicted vectors $\hat{p}_i$ and $\hat{p}_j$, is based on membership in the set of $k = 2$ or $k = 3$ neighbors.](image-url)
3 Language Model Distance

An exact measurement of equality is not possible for high-dimensional NLP embeddings. Embeddings of several hundred dimensions, and one-hot encoded vectors on the order of tens of thousands of dimensions, are particularly difficult to measure.

\[ LMD(p, t, m, k) = \begin{cases} True, & \text{if } t \in m.set(p, k) \\ False, & \text{otherwise} \end{cases} \]  

(1)

Instead, we suggest that the true measure of NLP vector distance is best provided by the model which defines the vectors. We present a family of metrics, Language Model Distance (LMD), which calculates distance and equality among NLP vectors by using the language model itself. LMD is defined as in Equation 1 where the distance between predicted vector \( \hat{p} \) and known truth vector \( t \), is provided by model \( m \), given neighbor threshold \( k \).

The distance measure is binary because it is based on set inclusion, and not physical or Euclidean distance. Thus, \( LMD \) can be used as a measure of accuracy, and records a true positive when \( t \) is within the neighborhood of the predicted vector \((t \in m.set(\hat{p}, k))\).

3.1 Measuring Accuracy with Language Model Distance

Figure 1 illustrates the distributional hypothesis by showing a simple clustering along 2 non-numeric dimensions. The circles represent neighborhoods \( m.set(\hat{p}, k = 2) \) and \( m.set(\hat{p}, k = 3) \). Note that the predicted vectors \((\hat{p})\) have no words directly associated with them, because no exact match is possible for floating point numeric vectors.

Thus we say that \( LMD_Accuracy(k) \) measures a positive result when truth vector \((t)\) is within the \( k \) sized neighborhood of the predicted vector \((t \in m.set(\hat{p}, k))\). For example, \( LMD_Accuracy(3) \) measures the percentage of times that the true word answer was among the top 3 closest predicted words.

Distributional distance functions can be used in neural network metrics, loss, or activation functions, or used directly in similarity computation. However, inserting external language models into neural networks can be difficult as these networks are firmly rooted in mathematics which is not compatible with linguistic processes.

We solve these difficulties by defining a simple class shown in Figure 2. By including the language model as a static member of the class, methods of the class may be used as network internal functions with access to external language models.

![Figure 2: Implementation of Distributional Accuracy based on Language Model Distance. A static language model (line 2) allows linguistic functionality to be included in purely mathematical models.](image)

4 Learning Orthogonal Procrustes Analysis

The Orthogonal Procrustes Algorithm is a process for finding the optimal mapping of one set of vectors to another. Typically, the vectors represent points in 2 or 3 dimensional space, for image processing, or they represent qualitative data measured in few dimensions (Mauricio et al., 2016). After resizing and repositioning of vectors, an optimal rotation matrix \( R \) is produced by a method similar to singular value decomposition.

This classic approach to vector transformation has been explored as a solution for some NLP tasks (Sen et al., 2019; Kim et al., 2019). Therefore we ask: Can a neural network be trained to perform the same optimal transformation for NLP embeddings which occupy a much higher dimensional space?

Our task is to train a simple MLP to learn the optimal mapping \( R \), between two disparate vector spaces representing a bilingual dataset. We measure the success of this task using \( LMD \) as the basis for accuracy as in Figure 2 and Equation 1.

We create two separate language models from a parallel corpus of European Parliament translations, the so called EuroParl dataset (Koehn, 2005). We use the Word2Vec model in continuous bag-of-words (CBOW) mode (Mikolov et al., 2013a) to build two separate distributions. By using a bilingual corpus, and training language models sep-
arately, we ensure that the models share a common
domain, but the vector spaces remain separate. For
training, we then map word vectors from one distri-
bution to the other, using a set of 1000 most com-
mon words pairs, obtained from from a language
learning website\footnote{http://www.englishnspanish.com}.

4.1 Results
Our results show that Orthogonal Procrustes Anal-
ysis can be learned for multilingual mapping of
word vectors. Furthermore, Figure 3 demonstrates
that LMD is effective as a basis for measuring the
accuracy of this task.

Figure 3 indicates that \textit{LMD\_Accuracy} is better
at measuring similarity in NLP embeddings than
\textit{cosine similarity}. In this plot, \textit{LMD\_Accuracy(1)}
indicates that the model exactly predicted the
correct word in the output language. When
\textit{LMD\_Accuracy(1)} is near 100% the value of \textit{cosine
similarity} should be near 1 which would indicate an
exact match. The fact that \textit{cosine similarity} cannot
measure this exact match shows a weakness in this
purely mathematical measurement compared with
our language model-based measurement.

5 Learning General Procrustes Analysis
To further test, we try to learn General Procrustes
Analysis; a much harder task because it requires
the network to generalize.

We have just shown that a simple neural net-
work can learn to transform vectors. This is non-
controversial since all neural networks perform this
task at every layer. However, not all networks are
able to generalize. Using the same network config-
uration as before, we now evaluate embeddings that
we have not seen in training, as is common. This is
equivalent to learning the \textit{Generalized Procrustes
Algorithm}.

Results in Figure 4 show that \textit{LMD\_Accuracy} is
more like cosine distance when generalization is
required. Note that we use \textit{LMD\_Accuracy} only
for metrics. This model uses \textit{cosine similarity}
for error calculation and back-propagation. We
conclude that such \textit{Lp norm} measurements can
only drive generalization as far as they are able to
measure accuracy.

The local variation in \textit{LMD\_Accuracy}, evident
in Figure 4, may be significant as it may make
determining the derivative of the function difficult.
The derivative of \textit{LMD\_Accuracy} must be worked
out before it can be incorporated into a loss function
and be used in back-propagation. The overall shape
of the curve, despite irregularities is encouraging
as the slope may be computed using ordinary least
squares in a calculation of rolling regression, or by
other numerical methods.

6 Conclusion
We suggest that language model metrics described
here may be incorporated directly into activation
and loss functions, and may be used as an error mea-
surement for back-propagation. We suggest this
basic enhancement would improve the Generalized
Procrustes Algorithm and other NLP processing in
general. This is left for future work.
References

Fabio Crosilla, Alberto Beinat, Andrea Fusiello, Eleonora Maset, and Domenico Visintini. 2019. Orthogonal procrustes analysis. In Advanced Procrustes Analysis Models in Photogrammetric Computer Vision, pages 7–28. Springer.

Nicolae Duta. 2015. Procrustes Shape Distance, pages 1278–1279. Springer US, Boston, MA.

Martin Engilberge, Louis Chevallier, Patrick Pérez, and Matthieu Cord. 2019. Sodeep: a sorting deep net to learn ranking loss surrogates. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 10792–10801.

John Rupert Firth. 1961. Papers in Linguistics 1934-1951: Repr. Oxford University Press.

John C Gower. 1975. Generalized procrustes analysis. Psychometrika, 40(1):33–51.

Yova Radoslavova Kementchedjhieva, Sebastian Ruder, Ryan Cotterell, and Anders Søgaard. 2018. Generalizing procrustes analysis for better bilingual dictionary induction. In 22nd Conference on Computational Natural Language Learning (CoNLL 2018), pages 211–220. Association for Computational Linguistics.

Yunsu Kim, Petre Petrov, Pavel Petrushkov, Shahram Khadivi, and Hermann Ney. 2019. Pivot-based transfer learning for neural machine translation between non-english languages. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 865–875.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In MT summit, volume 5, pages 79–86. Citeseer.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. volume 37 of Proceedings of Machine Learning Research, pages 957–966, Lille, France. PMLR.

Rensis Likert. 1932. A technique for the measurement of attitudes. Archives of psychology.

Angélica Mauricio, A.B. Palazzo, Valeria Caselato, and Helena Bolini. 2016. Generalized procrustes analysis and external preference map used to consumer drivers of diet gluten free product. Food and Nutrition Sciences, 07:711–723.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013a. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111–3119.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 26, pages 3111–3119. Curran Associates, Inc.

Radim Řehůrek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pages 45–50, Valletta, Malta. ELRA. http://is.muni.cz/publication/884893/en.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019. A survey of cross-lingual word embedding models. Journal of Artificial Intelligence Research, 65:569–631.

Sukanta Sen, Kamal Kumar Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2019. Multilingual unsupervised nmt using shared encoder and language-specific decoders. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3083–3089.