A Distributional Semantics Based on Gesture

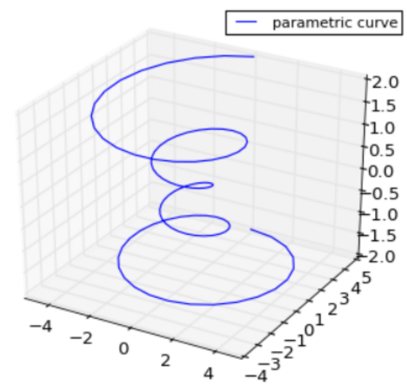
By: Aksel Ozer

(Section 1.1) Introduction and Research purpose:

The broader goal of this paper is to pave a research pathway for developing stronger NLP algorithms. Most algorithms today rely on some form of distributional semantics which attempt to derive semantic relationships between words based on the larger distribution of these words in large language datasets. The most naive implementation of this idea is that words that appear frequently together in similar contexts are more semantically related to each other than ones that don't. Humans intuitive notion of semantics is much different than the statistical distributional models embedded in these computers and therefore it is worth exploring more holistic approaches to generating computers' semantic competence. One way to do this would be to add to the existing distributional semantics with a kinematic distributional semantics based on the iconism and kinematic similarity of semantically related gestures. This can be done by training two different neural nets on the same data and then merging them later. There are multiple methods avaliable to merge two different neural nets, provided the inputs are the same and both share the same number of nodes but I will not be going into that in this paper. To even begin to answer this question, however, we need to discuss what we mean by kinematically similar. In the paper, the mathematical measure used to define kinematic similarity was the summation of distance between the time warping series of two gestures calculated as euclidean distance. I think there is a wealth of additional mathematical features we can exploit in this space to answer the question of how similar any two gestures are kinematically.

(Section 1.2) Explaining the software used:

Electrodes are attached to the key points on the participants bodies so that a 3D motion detection software can capture the entirety of gestures movement and map out the trajectory of the gesture stream onto a three dimensional graph on the computer. Once the movement traces are plotted onto a graph in a computer in the form of a three dimensional parametric curve we can analyze the motion of the gestures in the same way we would analyze any other graph. Note this graph is also essentially four dimensional as we can read off the time of any particular movement in the gesture sequence by simply hovering over the curve and reading off its timestamp.



Section 2.1 Experimental Modifications:

The procedure of most studies attempting to creating a distribtional kinematics based on gesture distances first asked participants to describe a certain set of words and then asked them to convey these words in gesture. The idea is that if semantically related words in the input set, determined by existing NLP algorithms, are also the same sets of words that are kineamtically similar then gesture kinematics becomes a very useful avenue for gleaning semantic relationships between words.

My goal in this section will be to show that we can ultimately recreate a matrix of distances, in the same form as the paper *Semantically Related Gestures Move Alike*, by calculating the n-dimensional Euclidean distance between gestures described as multidimensional vectors of different spatial measures. Each gesture will be assigned a column vector of n rows, with each row holding a numerical value corresponding to a unique spatial measure. Below, I will discuss the various different spatial measures we can use, relying on basic concepts from calculus and differential geometry to obtain a unique vector for each gesture. The calculation of distances between two gestures Gi and Gj that will be put back into the matrix at the ith row and jth column to derive a matrix will be calculated by the equation below:

$$d\left(G^i, G^j\right) = \sqrt{\sum_{k=1}^{n}\left(G_k^i - G_k^j\right)^2}$$

Here we are taking the square root of the sum across all rows of the square of differences in the kth row of two different gestures. This is exactly how we would ordinarily find the

$$d\left(G^i,G^j\right) = \begin{pmatrix} \sqrt{\left(g_1^i - g_1^i\right)^2} \\ \cdots \\ \cdots \\ \sqrt{\left(g_{20}^i - g_{20}^i\right)^2} \end{pmatrix}$$
$$= \quad \sqrt{\left(g_1^i - g_1^i\right)^2} + \ldots + \sqrt{\left(g_{20}^i - g_{20}^i\right)^2}$$

$$\begin{array}{cc} & j \\ i & \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & & & & & \cdot \\ \cdot & & & d\left(G_i,G_j\right) & & & \cdot \\ \cdot & & & & & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} \end{array}$$

distance between two points in two dimensional space just analogized to higher dimensions. Note that all gestures have exactly the same dimensions.

Section 2.2 Constructing the n-dimensional vector:

Using 3D motion software, we can map out on a computer the three dimensional trajectories taken by salient parts of the gesturer's body- called key points- and graph them on a three dimensional axis in parametric form as a function of time and analyze the gestures motion as we would any other graph.

Later on I will motivate and explain the reason behind each of the spatial measures I have chosen but for now I will merely list how to calculate each one.

$$\left[R_1\right]\text{:}\textit{Total Distance over x axis} = \int_{t_0}^{t_1} \left|v_x(t)\right| dt$$

$$\left[R_2\right]\text{:}\textit{Total Distance over y axis} = \int_{t_0}^{t_1} \left|v_y(t)\right| dt$$

$$\left[R_3\right]\text{:}\textit{Total Distance over z axis} = \int_{t_0}^{t_1} \left|v_z(t)\right| dt$$

$$\left[R_4\right]\text{:}\textit{Total Displacement over x axis} = \int_{t_0}^{t_1} v_x(t) dt$$

$$\left[R_5\right]\text{:}\textit{Total Displacement over y axis} = \int_{t_0}^{t_1} v_y(t) dt$$

$$\left[R_6\right]\text{:}\textit{Total Displacement over z axis} = \int_{t_0}^{t_1} v_z(t) dt$$

$$\left[R_{7-9}\right]\text{: }\textit{Average Velocity over x,y,z axis} = \frac{1}{t}\int_{t_0}^{t_1} a_{x,y,z}(t) dt$$

$$\left[R_{10-12}\right]\text{:}\textit{Av. Acceleration over x,y,z axis} = \frac{1}{t}\int_{t_0}^{t_1} a'_{x,y,z}(t) dt$$

$$\left[R_{13-15}\right]\text{: }\textit{Total Velocity over x,y,z axis} = \int_{t_0}^{t_1} a_{x,y,z}(t) dt$$

$$\left[R_{16-18}\right]\text{:}\textit{Total Acceleration over x,y,z axis} = \int_{t_0}^{t_1} a'_{x,y,z}(t) dt$$

$$\left[R_{19}\right]\text{: }\textit{Average Curvature over curve} = \frac{1}{s}\int_C \frac{\left|T'\right|}{\left|r'\right|} ds$$

$$\left[R_{20}\right]\text{: }\textit{Total Curvature over curve} = \int_C \frac{\left|T'\right|}{\left|r'\right|} ds$$

Section 2.2 Justifying the measures used:

The first six measures are a good way to tell us roughly where the majority of the gesture sequence took place in the gesture space. By themselves, each measure does not tell us too much about how the gesture sequences unfold, however pairing the total distance with the total displacement can give us useful insights about the motion. For example if total displacement is zero but total distance is a large number we know that there was lots of motion across the x axis just that it was symmetrical across the positive and negative axises and therefore canceled each other out in the absence of the absolute value used in the displacement summation. These values could be created by a sweeping motion with the hands moving first in the left direction and back and then the right direction and back. Potentially, someone could have made these motions when attempting to convey the idea that it could go either way, placing no special merit to one side or the other. The sign of displacement tells us the positive or negative nature of the motion across the axis of choice and the distance calculation with an absolute value tells us the total amount of motion. Comparing each distance or displacement value for each of the three axises then tells us which axises or pair of axes were favored by the gesture sequence. Just with these three data points it becomes very easy to measure symmetry across the axises. For example if the displacement of the x,y, and z axes were all roughly the equal we would know that this gesture series was roughly symmetrical across all axes. Over large datasets we could expect to see certain shapes emanating for similar gesture sequences do to the fact that for example, cube-like gesture series, those symmetrical across all axises, would be closer to each other in the distance calculation then they would be to triangle-like gestures which only would be symmetrical across the x-axis and y-axis and asymmetrically positive or negative across the z-axis. Thus the utility of these seemingly very simple features [R1-R6] cannot be underestimated as they are sufficient to derive shapes which can be useful building blocks for inconism in gesture to semantic mapping. Measures [R7-R18] function much in the same way that measures [R1-R6] by adding useful information about the relative speed of motion and acceleration across the axises and provide a useful tool for comparing the sorts of motion across different axises. Curvature determines the degree to which a path deviates from a straight line. Thus lines have zero curvature and a right angle would have maximum curvature. Measure [R19] looks at average curvature and would be a good method for determining globally how curvy or a choppy a gesture sequence is. Measure [R20] is needed to accompany average curvature, as averages can often dilute the effects of outliers. Now in most cases outliers are not of interest however it could be the case the a point of high curvature occurs at a salient point in a gesture series and is central to conveying the semantic content of the gesture. Thus [R20] merely adds of points of curvature throughout the sequences and would therefore be sensitive to the effects of points of high curvature. This is expected to be a useful measure as we could intuitively image that participants attempting to conveying abrupt events that happen suddenly, for example a car crash or kicking a ball into a goal, to convey these events in a gestuer series that is highly choppy, with moments of high curvature. Gesture series with low curvature, on the other hand, could be imagined to convey ideas such as a peaceful walk in the park and a lazy day at home watching television.

Sources:

(1) Pouw, Wim, et al. “Semantically Related Gestures Move Alike: Towards a Distributional Semantics of Gesture Kinematics.” 2021, doi:10.31219/osf.io/pgq6m.

(2) Beecks, Christian, et al. “Spatiotemporal Similarity Search in 3d Motion CAPTURE Gesture Streams.” *Advances in Spatial and Temporal Databases*, 2015, pp. 355–372., doi:10.1007/978-3-319-22363-6_19.

(3) “Vision-Based Hand Gesture Recognition for Human-Computer Interaction.” *The Universal Access Handbook*, 2009, pp. 577–606., doi:10.1201/9781420064995-46.

(4) Cheng, Jun, et al. “Feature Fusion for 3d Hand Gesture Recognition by Learning a Shared Hidden Space.” *Pattern Recognition Letters*, vol. 33, no. 4, 2012, pp. 476–484., doi:10.1016/j.patrec.2010.12.009.