© 2017 IJSRSET | Volume 3 | Issue 8 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Comparison of Cosine, Euclidean Distance and Jaccard Distance

Manpreet Singh Lehal
Department of Computer Science and IT, Lyallpur Khalsa College, Jalandhar, Punjab, India

ABSTRACT

The task of measuring sentence similarity is defined as determining how similar the meaning of two sentences is. The higher the score, the more similar the meaning of the two sentences. The task of identifying similarity is not an easy one because of variability in natural language expressions. Hence the similarity metrics give varied results in many of the cases and choosing the right measure is crucial to the efficiency of the system. This paper compares and analyses three similarity measures: Euclidean Distance, Cosine Similarity and Jaccard Distance and points out the usage of each metric.

Keywords : Euclidean Distance, Cosine Similarity, Jaccard Distance

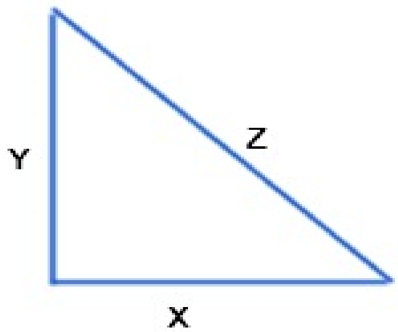
I. INTRODUCTION

The task of measuring sentence similarity is defined as determining how similar the meanings of two sentences are. Computing sentence similarity is not a trivial task, due to the variability of natural language - expressions. Measuring semantic similarity of sentences is closely related to semantic similarity between words. It makes a relationship between a word and the sentence through their meanings. The intention is to enhance the concepts of semantics over the syntactic measures that are able to categorize the pair of sentences effectively. Semantic similarity plays a vital role in Natural language processing, Informational Retrieval, Text Mining, Q & A systems, text-related research and application area.

II. The Euclidean distance

In either the plane or 3-dimensional space is simply the shortest distance between the two points. It is also called Pythagorean metric as it forms a right-angled triangle and is used to find the similarity between the two points. It helps to identify the sameness of vectors and hence find translation pairs in NLP. The higher the score, the less similar are the vectors.

In a right-angled triangle, as shown below, the square of the hypotenuse (the side denoted by Z) is equal to the sum of the squares of the other two sides (Y and Z); that is, $Z^2 = X^2 + Y^2$.



The immediate consequence of this is that the squared length of a vector $\vec{v} = [v_1, v_2]$ is the sum of the squares of its coordinates (see triangle OPA in **Figure 1**, or triangle OPB $-|OP|^2$ denotes)

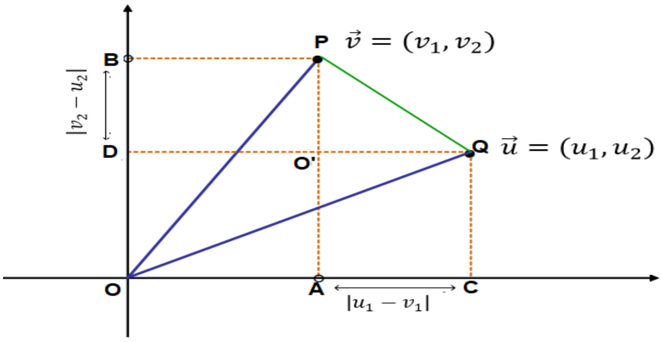


Figure 1 Pythagoras' theorem applied to distances in 2-D space

the squared length of v , that is the distance between point O and P); and the squared distance between two $\vec{u} = [u_1, u_2]$ and $\vec{v} = [v_1, v_2]$ is the sum of squared differences in their coordinates (see triangle PQO' in **Figure 1**; $|OQ|^2$ denotes the squared distance between points P and Q). To denote the distance between vectors $\vec{u}$ and $\vec{v}$ we can use the notation $\delta_{u,v}$ so that this last result can be written as:

In rt. Angle triangle Δ PO'Q

$$|PQ| = \sqrt{O'Q^2 + O'P^2}$$

$$\delta_{u,v} = \sqrt{(u_1 - v_1)^2 + (v_2 - u_2)^2}$$

The distance between points P and O is the distance between the vector $\vec{u} = [u_1, u_2]$ and the zero vector $\vec{0} = [0,0]$ with coordinates all zero:

In rt. Angle triangle Δ OAP

$$|OP| = \sqrt{OA^2 + OB^2}$$

$$\delta_{0,v} = \sqrt{{v_1}^2 + {v_2}^2}$$

In rt. Angle triangle Δ OCQ

$$|OQ| = \sqrt{OC^2 + OD^2}$$

$$\delta_{u,0} = \sqrt{{u_1}^2 + {u_2}^2}$$

which we could just denote by δ_u. The zero vector is called the origin of the space.

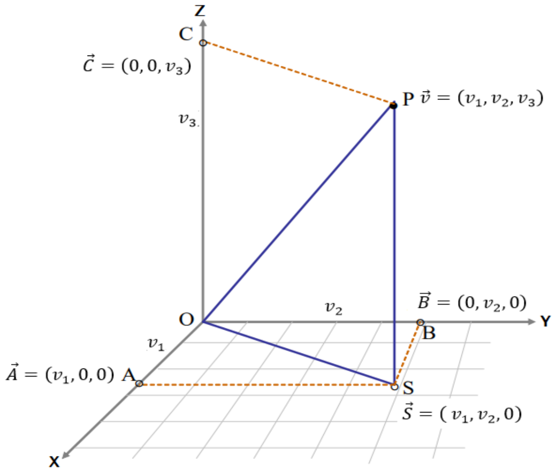


Figure 2 Pythagoras' theorem extended into 3-D space

We move immediately to a three-dimensional point $\vec{v} = [v_1, v_2, v_3]$ shown in **2**. The three coordinates are at points A, B and C along the axes, and the angles AOB, AOC and COB are all 90° as well as the angle OSP at S, where the point P (depicting vector $\vec{v}$) is like projection onto the 'floor'. Using Pythagoras' theorem twice we have:

$$|OA|^2 = {v_1}^2$$

$$|OB|^2 = {v_2}^2$$

$$|OC|^2 = {v_3}^2$$

In rt. Angle triangle Δ OSP

$$|OP|^2 = |OS|^2 + |SP|^2 - - - - (1)$$

In rt. Angle triangle Δ OAS

$$|OS|^2 = |OA|^2 + |AS|^2 - - - - - (2)$$

From (1) and (2) we will get

$$|OP|^2 = |OA|^2 + |AS|^2 + |SP|^2$$

$$|OP|^2 = {v_1}^2 + {v_2}^2 + {v_3}^2$$

$$|OP| = \sqrt{{v_1}^2 + {v_2}^2 + {v_3}^2}$$

$$\delta_v = \sqrt{{v_1}^2 + {v_2}^2 + {v_3}^2}$$

It is also clear that placing a point Q in Figure 2.5 to depict another vector $\vec{u}$ and going through the motions to calculate the distance between $\vec{u}$ and $\vec{v}$ will lead to

$$\delta_{u,v} = \sqrt{(u_1 - v_1)^2 + (v_2 - u_2)^2 + (v_3 - u_3)^2}$$

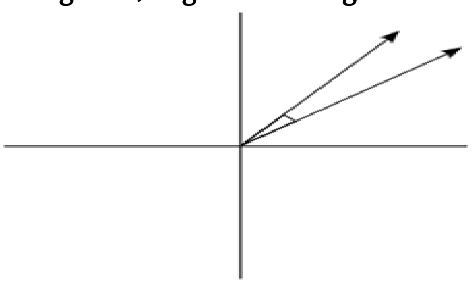
III. Cosine similarity

Is a measure of the cosine of the angle between two non-zero vectors (arrays of the word count) projected in a multi-dimensional space, where both vectors are normalized to 1 and computes the similarity of documents independent of the size of the documents. The value of cosine of 0 degree is 1 and it is less than 1 for the angles between (0, pie) radians. The cosine similarity is used in positive space, where the output is clearly represented in binary forms of zeros and one.

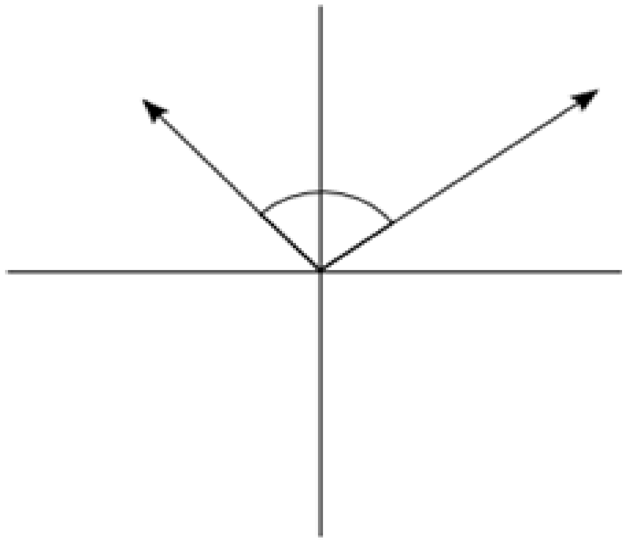
It measures the angle or orientation of the documents unlike Euclidean distance which measures the magnitude. If the two documents have higher distance (means they are far from one another in terms of Euclidean distance) still they can have smaller angle. There will be a greater number of common words in large documents but this doesn't mean they are similar. Cosine similarity is the solution to this problem. Two vectors which are parallel or oriented in the same direction will have a smaller angle and their cosine similarity will be 1, means they will be similar; vectors which are perpendicular to one another will have larger angle and their cosine similarity is 0, means they are dissimilar; vectors which are in opposite directions will have a cosine similarity of -1, means they are similar. It implies that if the vectors are far away from one another still they can have a smaller angle and prove their similarity in case of cosine similarity.

In case of information retrieval, the terms are characterized by different dimensions and the documents are represented in the form of vectors. The value of vector is equal to the frequency of the times it appears in the document. Cosine similarity here measures the similarity of the two documents in terms of their content/words.

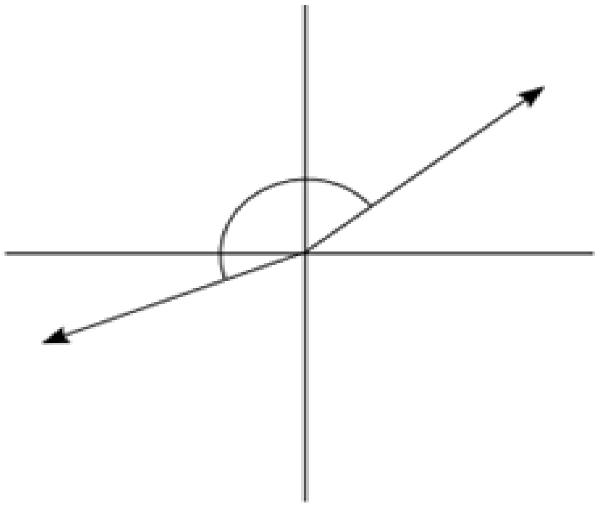
The cosine similarity equation is solved as a dot product for cos theta. It generates a metric that says how related are two documents by looking at the angle instead of magnitude, as shown below as shown in **Figure 3**, **Figure 4** and **Figure 5**:



[1]Figure 3 Vectors in the same direction
(angle between them is nearly 0 degree; cosine of the angle is near 1)



[2]Figure 4 Orthogonal vector
(angle is nearly 90 degree; cosine of angle is near 0)



[3]Figure 5 Vectors in opposite direction
(angle between them is nearly 180 degree; cosine of angle is -1)

Cosine similarity can overcome the problem of higher count of terms because even if the vectors points are far away from one another, they still can have a small angle between them. Let's say, we have a term which

[1] https://blog.christianperone.com/wp-content/uploads/2013/09/cosinesimilarityfq1.png

[2] https://blog.christianperone.com/wp-content/uploads/2013/09/cosinesimilarityfq1.png

[3] https://blog.christianperone.com/wp-content/uploads/2013/09/cosinesimilarityfq1.png

occurs 100 times in one document and only 10 times in another document, they can have a small angle because they point towards the same direction, however the Euclidean distance between them will be more.

IV. Jaccard Similarity or Jaccard index

is a measure to find similarity and difference of sample sets. Jaccard coefficient finds the similarity and is obtained by dividing the intersection by the union of the sets. Jaccard distance finds the dissimilarity and is obtained by subtracting the coefficient from 1. The value of dissimilarity will be 0. The Jaccard Index, also known as the Jaccard similarity coefficient, measures similarity between finite sample sets, and is formally defined as the size of the intersection divided by the size of the union of the sample sets. The mathematical representation of the index is written as:

$$Jaccard(\alpha, \beta) = \frac{|\alpha \cap \beta|}{|\alpha \cup \beta|} = \frac{|\alpha \cap \beta|}{|\alpha| + |\beta| - |\alpha \cap \beta|}$$

Whereas Jaccard index measures similarity, Jaccard distance measures dissimilarity between sample sets. It is calculated by finding the Jaccard index and subtracting it from 1, or alternatively dividing the differences by the intersection of the two sets.

To find the Jaccard Index we divide the number in both sets by the number in either set, multiplied by 100. It gives the similarity between the sets in the form of percentage. When we subtract this percentage from 1, we get the Jaccard distance. For example, if the similarity measurement is 45%, then the Jaccard distance (1 - 0.45) is 0.55 or 55%.

Jaccard similarity is based on set theory so repetition of words does not affect it whereas in case of Cosine similarity repetition of words affect the calculations.

We will compute similarity between three documents to compare the three measures. Suppose we have three documents:

d1 - Music is a universal language

d2 - Music is a miracle

d3 - Music is a universal feature of the human experience

We want to find document Similarity of d3 with other two documents d1-d3 and d2-d3

Jaccard Distance between d1 and d3 –

$$Jaccard(d1, d3) = \frac{|d1 \cap d3|}{|d1 \cup d3|}$$

J(d1, d3) = 4/10

= 0.4

Jaccard Distance between d2 and d3 –

$$Jaccard(d2, d3) = \frac{|d2 \cap d3|}{|d2 \cup d3|}$$

J(d2, d3) = 3/10

= 0.3

Euclidean Distance between d1 and d3 using relative term frequency values –

E(d1, d3) = sqrt[(0.2 - 0.11)^2 + ... + (0 - 0.11)^2 + (0 - 0.11)^2 + (0 - 0.11)^2]

= sqrt[0.0081 + 0.0081 + 0.0081 + 0.0081 + 0.012 + 0.012 + 0.012 + 0.012 + 0.012]

= sqrt(0.1329)

= 0.364554523

Euclidean Distance between d2 and d3 using relative term frequency values –

E(d2, d3) = sqrt[(0.25 - 0.11)^2 + (0.25 - 0.11)^2 + ... + (0 - 0.11)^2 + (0 - 0.11)^2]

= sqrt[0.0196 + 0.0196 + 0.0196 + 0.012 + 0.0625 + 0.012 + 0.012 + 0.012 + 0.012 + 0.012]

= sqrt(0.1939)

= 0.440340777

Cosine similarity between d1 and d3 –

num = [0, 0, 0, 0.035, 0.095, 0, 0, 0, 0, 0, 0] * [0, 0, 0, 0.019, 0, 0, 0.052, 0.052, 0.052, 0.052, 0.052]

= 0*0 + 0*0 + 0*0 + 0.035*0.019 + 0.095*0 + 0*0 + 0*0.052 + 0*0.052 + 0*0.052 + 0*0.052 + 0*0.052

= 0.000665

den = sqrt[0 + 0 + 0 + 0.0012 + 0.009 + 0 + 0 + 0 + 0 + 0 + 0] * sqrt[0 + 0 + 0 + 0.0003 + 0 + 0 + 0.0027 + 0.0027 + 0.0027 + 0.0027 + 0.0027]

= 0.0102 + 0.0138

= 0.024

cosθ = 0.000665 / 0.024

= 0.028

Cosine similarity between d2 and d3 –

num = [0, 0, 0, 0, 0, 0.119, 0, 0, 0, 0, 0] * [0, 0, 0, 0.019, 0, 0, 0.052, 0.052, 0.052, 0.052, 0.052]

= 0*0 + 0*0 + 0*0 + 0*0.019 + 0*0 + 0.119*0 + 0*0.052 + 0*0.052 + 0*0.052 + 0*0.052 + 0*0.052

= 0

Thus, cosθ = 0

We find that d1 and d3 have greater Cosine Similarity as is intuitive. Here most accurate document similarity is provided by Cosine Similarity. Jaccard Distance is fairly accurate as it states that the document pair d1 and d3 are more similar as compared to d2 and d3. Euclidean Distance, also gives fairly accurate value.

V. CONCLUSION

Thus, the effectiveness of the metric depends on the task for which they are being used. Some tasks, such as preliminary data analysis, benefit from cosine as well as Euclidean distance measure; each of them allows the extraction of different insights on the structure of the data. Text classification, generally function better under Euclidean distances. Some more, such as retrieval of the most similar texts to a given document, generally function better with cosine similarity. Cosine similarity is generally used for measuring distance when the magnitude of the vectors does not matter. This happens for example when working with text data represented by word counts. We could assume that when a word (e.g. science) occurs more frequent in document 1 than it does in document 2, that document 1 is more related to the topic of science. When we work with documents of uneven length, some words occur more in a longer document. In such cases cosine similarity proves beneficial.

VI. REFERENCES

[1]. Aliguliyev, R. M. (2009). A new sentence similarity measure and sentence based extractive technique for automatic text summarization. Expert Systems with Applications. https://doi.org/10.1016/j.eswa.2008.11.022

[2]. Bandyopadhyay, S., & Mallick, K. (2013). A new path based hybrid measure for gene ontology similarity. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 11(1), 116–127.

[3]. Bani-Ahmad, S., Cakmak, A., Ozsoyoglu, G., & Hamdani, A. A. (2005). Evaluating Publication Similarity Measures. IEEE Data Engineering Bulletin.

[4]. Barrón-Cedeno, A., Rosso, P., Agirre, E., & Labaka, G. (2010). Plagiarism detection across distant language pairs. Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), 37–45.

[5]. Chen, F., Farahat, A., & Brants, T. (2004). Multiple similarity measures and source-pair information in story link detection. Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004, 313–320.

[6]. Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. Journal of the American Society for Information Science, 41(6), 391–407.

[7]. Gabrilovich, E., Markovitch, S., & others. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. IJcAI, 7, 1606–1611.

[8]. Gomaa, W. H., Fahmy, A. A., & others. (2013). A survey of text similarity approaches. International Journal of Computer Applications, 68(13), 13–18.

[9]. Hall, P. A. V, & Dowling, G. R. (1980). Approximate string matching. ACM Computing Surveys (CSUR), 12(4), 381–402.

[10]. Heltshe, J. F. (1988). Jackknife Estimate of the Matching Coefficient of Similarity. Biometrics, 44(2), 447. https://doi.org/10.2307/2531858

[11]. Islam, A., & Inkpen, D. (2008). Semantic text similarity using corpus-based word similarity and string similarity. ACM Transactions on

Knowledge Discovery from Data (TKDD), 2(2), 1–25.

[12]. Jaro, M. A. (1989). Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. Journal of the American Statistical Association, 84(406), 414–420.

[13]. Jaro, M. A. (1995). Probabilistic linkage of large public health data files. Statistics in Medicine, 14(5–7), 491–498.

[14]. Jiang, J. J., & Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. ArXiv Preprint Cmp-Lg/9709008.

[15]. Jiang, Y., Zhang, X., Tang, Y., & Nie, R. (2015). Feature-based approaches to semantic similarity assessment of concepts using Wikipedia. Information Processing & Management, 51(3), 215–234.

[16]. Kolb, P. (2009). Experiments on the difference between semantic similarity and relatedness. Proceedings of the 17th Nordic Conference of Computational Linguistics (NODALIDA 2009), 81–88.

[17]. Krause, E. F. (1986). Taxicab geometry: An adventure in non-Euclidean geometry. Courier Corporation.

[18]. Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychological Review, 104(2), 211.

[19]. Leacock, C., & Chodorow, M. (1998). Combining local context and WordNet similarity for word sense identification. WordNet: An Electronic Lexical Database, 49(2), 265–283.

[20]. Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. Proceedings of the 5th Annual International Conference on Systems Documentation, 24–26.

[21]. Liu, N., Zhang, B., Yan, J., Yang, Q., Yan, S., Chen, Z., Bai, F., & Ma, W. Y. (2004). Learning similarity measures in non-orthogonal space. International Conference on Information and Knowledge Management, Proceedings.

[22]. Mandreoli, F., Martoglia, R., & Tiberio, P. (2002). Searching Similar (Sub) Sentences for Example-Based Machine Translation. SEBD.

[23]. Mihalcea, R., Corley, C., Strapparava, C., Jiang, J. J., Conrath, D. W., Leacock, C., Chodorow, M., Wu, Z., Palmer, M., Hirst, G., St-Onge, D., others, Banerjee, S., Pedersen, T., Patwardhan, S., Hall, P. A. V, Dowling, G. R., Peterson, J. L., Jaro, M. A., … Inkpen, D. (2006). Approximate string matching. WordNet: An Electronic Lexical Database, 104(2), 1606–1611.

[24]. Miller, G. A., Beckwith, R., Fellbaum, C. D., Gross, D., & Miller, K. (n.d.). WordNet: An online lexical database. 1990. Int. J. Lexicograph, 3(4).

[25]. Needleman, S. B., & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology, 48(3), 443–453.

[26]. Peterson, J. L. (1980). Computer programs for detecting and correcting spelling errors. Communications of the ACM, 23(12), 676–687.

[27]. Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. ArXiv Preprint Cmp-Lg/9511007.

[28]. Vijaymeena, M. K., & Kavitha, K. (2016). A survey on similarity measures in text mining. Machine Learning and Applications: An International Journal, 3(2), 19–28.

[29]. Xia, P., Zhang, L., & Li, F. (2015). Learning similarity with cosine similarity ensemble. Information Sciences, 307, 39–52.