Contradection – *detecting **contradictions** in text*

Rami B. Houssami

m.s. Data Science

The George Washington University, Washington DC.

→ ←

THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

Objective: <u>To develop a machine capable method for detecting contradictions in text.</u>

Language structure comes in a variety of forms but is always received sequentially as meaning unfolds to the interpreter. Large volumes of text can create a myriad of relationships and concepts making it challenging to pickup on contradictions that are far apart. This work explores word embedding's in vector spaces and proposes a mapping technique that optimizes negation querying.

Challenges:

Because we are after negation, we have to compare objects to many attributes. These attribute features are sensitive to manipulation in vector subspace while preserving their relation to the object.

There's also the <u>challenge of words with multiple meaning</u> (Polysemy) and the <u>peculiarity of context</u> (which is somewhat captured in embedding's); working with referential terms, abstractions, and predicates.

Method

The method employed here combines Distributional Semantics from Corpus Linguistics techniques with Vector Space Modeling (VSM) and neural probabilistic models (word2vec).

>Group: *for meaning*

By leveraging word embedding techniques, which is a vectorized representation of words in a corpus, it is possible to group words of similar meanings together in the vector subspace.

Kernel PCA> $\mathbf{V}^{k^T}\Phi(\mathbf{x}) = \left(\sum_{i=1}^{N} \mathbf{a_i}^k \Phi(\mathbf{x_i})\right)^T \Phi(\mathbf{x})$

>Reduce: *to compress dimensionality - PCA vs t-SNE*

Word emeddings are generated with very high dimensionality, and different reduction methods generate different grouping outcomes. Hence reduction methods on attributes are different in order to keep opposing meanings on opposing sides of the hyperplane.

Kullback-Leibler divergence> $KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$

>Map: *for object-attribute cosines*

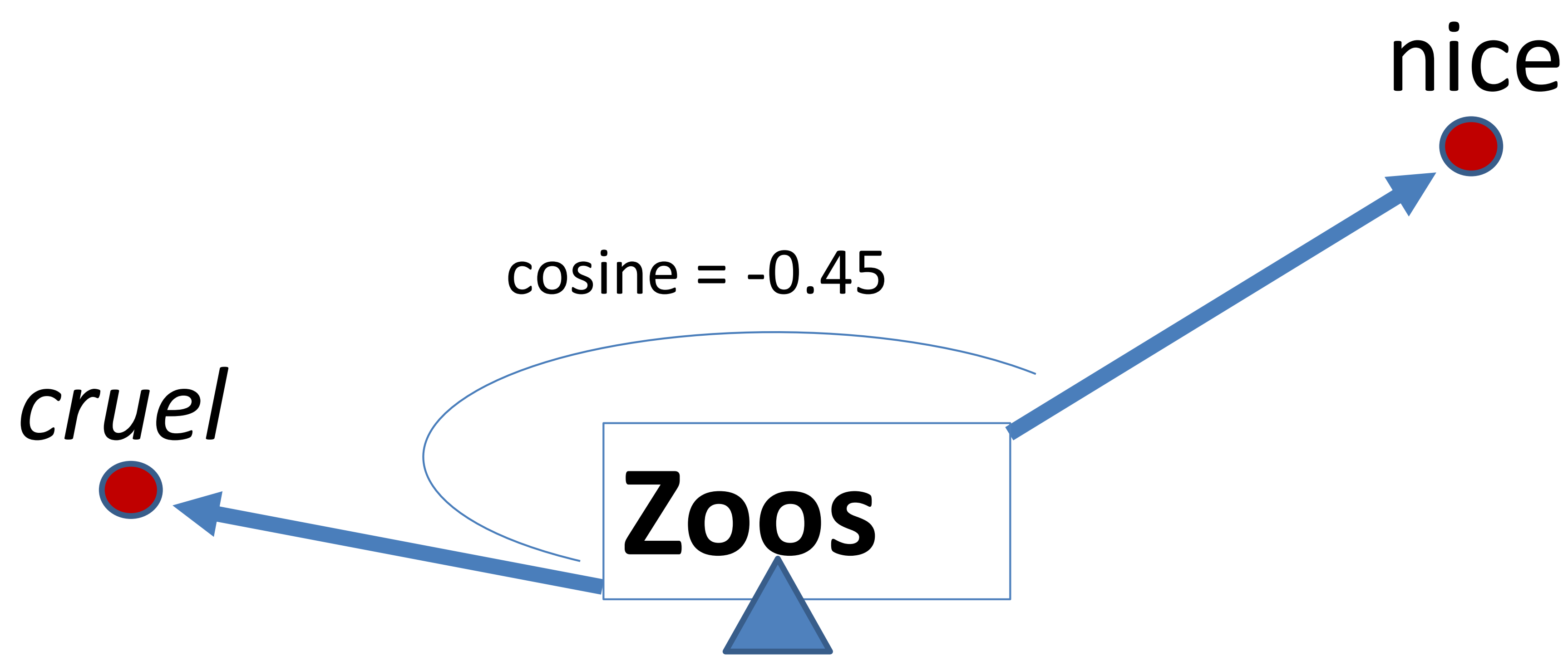
Mapping objects, or groups of objects for abstraction from the initial vectorization of objects to the newly reduced attribute-vector gives us our object-attribute map within the Vector Space Model. We can now utilize the cosine values of each object-attribute_pair to detect opposing relations. The average of which, with respect to each object, can suggest the consistency of this objects description within the target text.

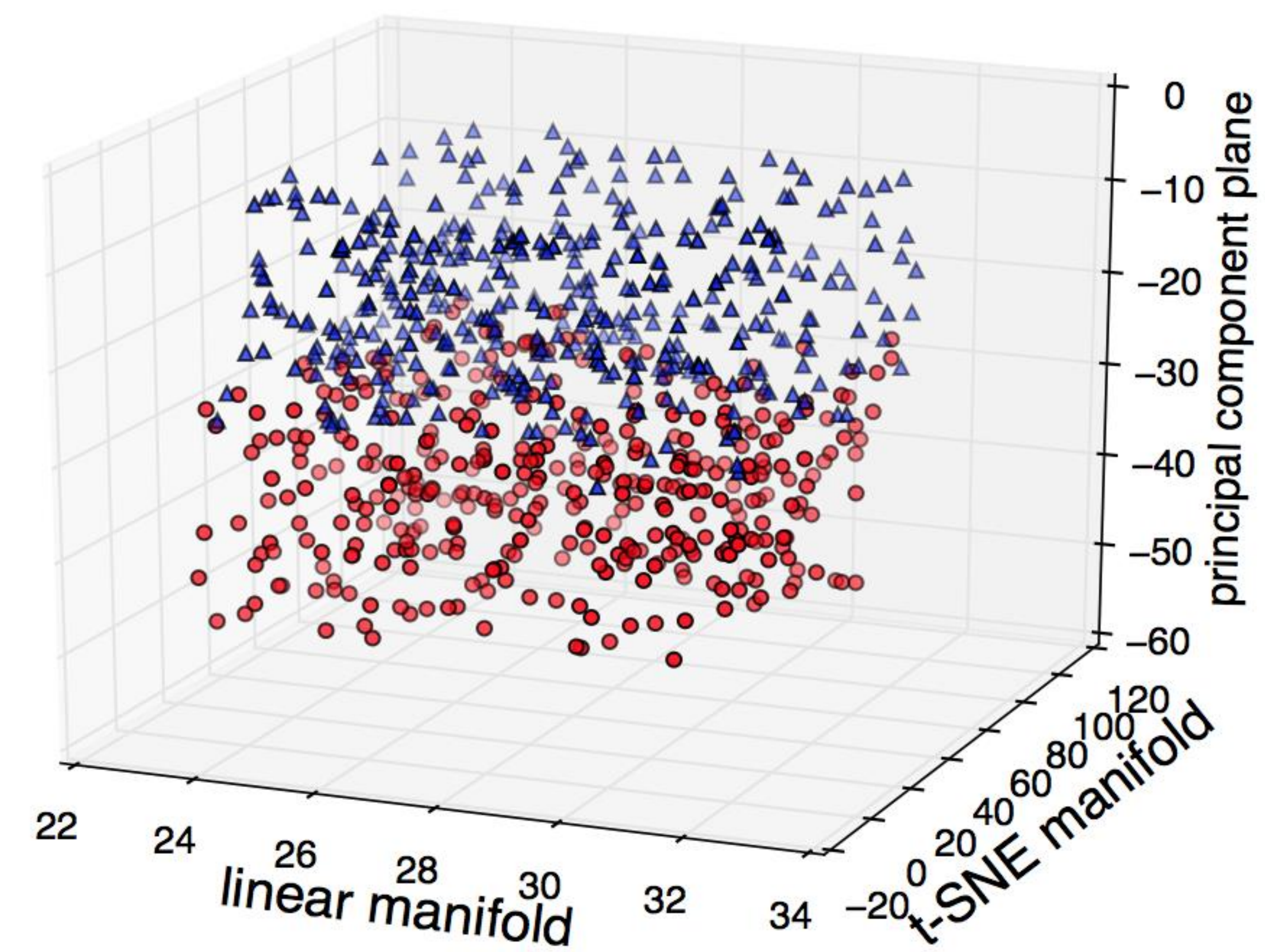
Next:

The current phase of this work is focused on combining the attribute embedding field with the object embedding's through subspace exploration. The next phase will utilize successful grouping techniques for object vectors, such as Word2Vec, to map object clusters to attributes such that referential words may also be considered in the object-attribute mapping. Further reduction and scaling methods are expected to be explored.

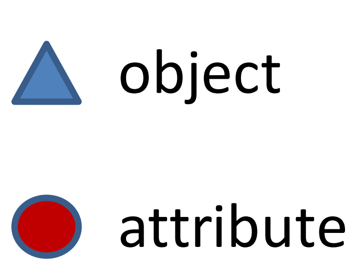
similarity function> $sim(d_k, q) = \frac{\sum_{j=1}^{n}\sum_{i=1}^{n} w_{i,k} * w_{j,q} * t_i \cdot t_j}{\sqrt{\sum_{i=1}^{n} w_{i,k}^2} * \sqrt{\sum_{i=1}^{n} w_{i,q}^2}}$

<u>*Purpose</u>: With a major motivation for this work being to support civil society, equality advocacy groups, and policy researchers, model tuning is done with a focus on the practical types of negation that would best empower analysts.
A well tuned detector may also be used to reconcile opposing theories, conduct dialogues analysis, or analyze the harmony in history and mathematics.






```
#def idealized detecting function
def contradection(corpus):
    #def relationships within the reduced plane
    relations = nx.DiGraph()
    relations.add_edge_from(Obj, Attr_pair)
    #looking for the cosine here
    def evaluate(relations):
        return np.cosine(relations)
    #return relations that are apparently contradictory
    for Obj in Corpus:
        if evaluate(Obj) =< 0.00
    return Obj
```



<u>Valued References:</u>

Word2Vec – word embedding training model (Mikolov, Tomas, et al. 2013)

T-SNE – Distributed Stochastic Neighbor Embedding (Van Der Maaten & Hinton 2008)