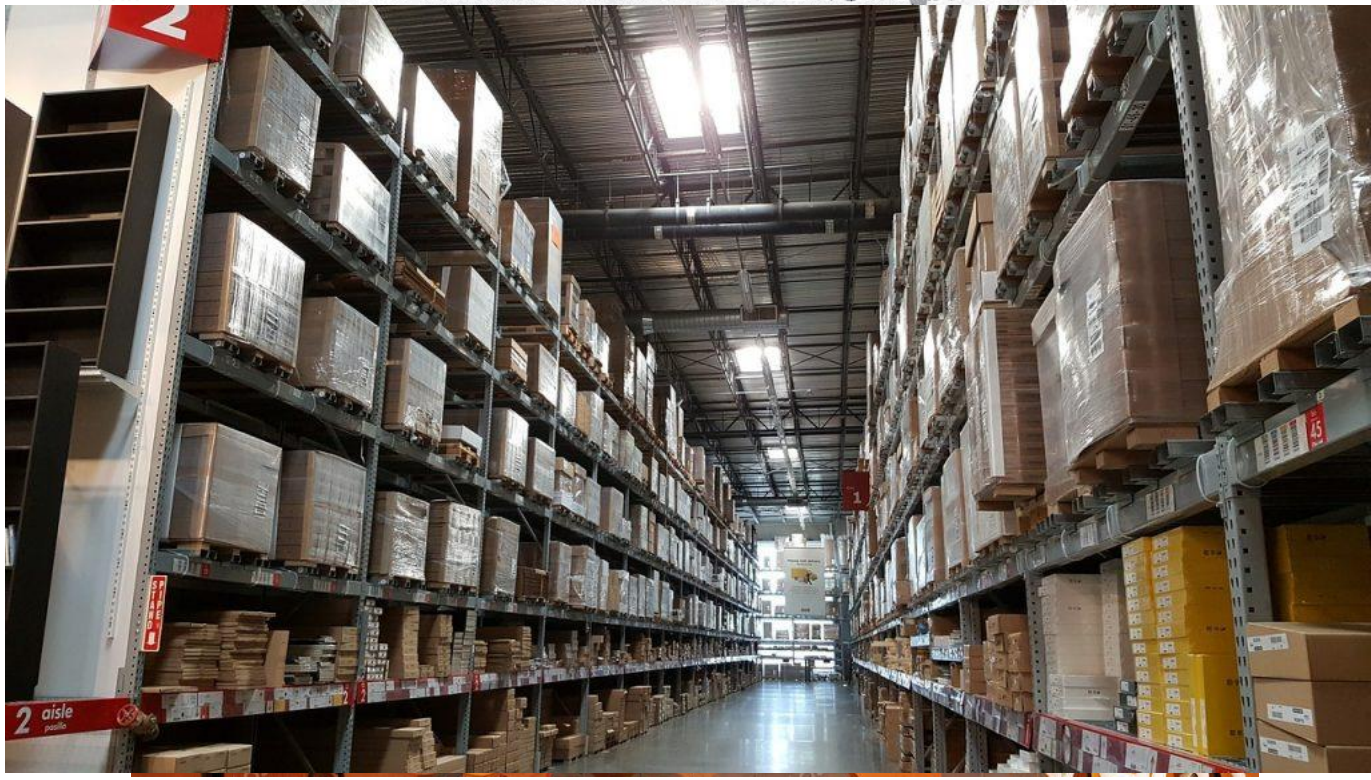
Complementary-Similarity Learning using Quadruplet Network

Mansi Ranjit Mane, **Stephen Guo**, Kannan Achan

E-commerce Recommender Systems

Nancy

eCommerce Customer



Millions of items in eCommerce Catalog/ Warehouse

Item Relationships

- Similarity/ Substitutes
  - Show similar items during exploration phase

Item Relationships

- Complementary
  - Show after purchase add example e.g. if someone has purchased dress, additional suggestions like sandals, purse etc

Traditional Methods

- Popularity Based
  - Always show popular items
  - Challenges:
    - Does not address item level relationship
- Customers who bought X also bought Y
  - based on co-counts
  - Challenges:
    - Cold-start items

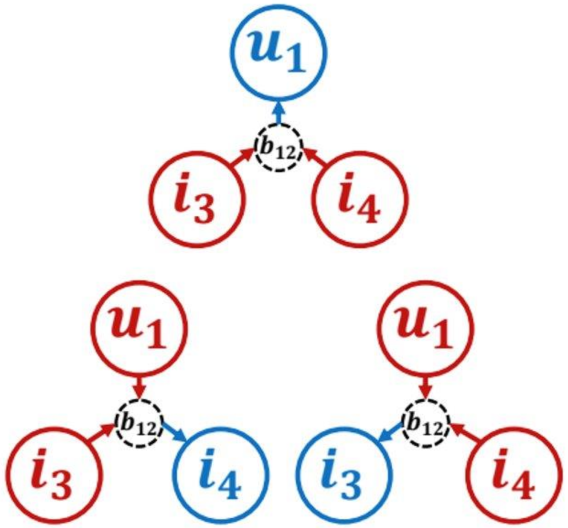
Traditional Methods

- Factorization Inspired Approaches - Factorize the Item-Item Co-Occurrence Matrix to learn low dimensional dual item latent factors
  - Challenges:
    - Cold Start
    - Does not address transitivity

	Item 1	Item 2	Item 3	Item 4
Item 1	1			2
Item 2		5	4	
Item 3				3
Item 4	9		6	

Other Methods

- Triple2vec
  - Maximize dot product between item pairs and user vectors
  - Challengers:
    - Transitivity leads to similar items in recommendations
    - Does not handle Cold-start items



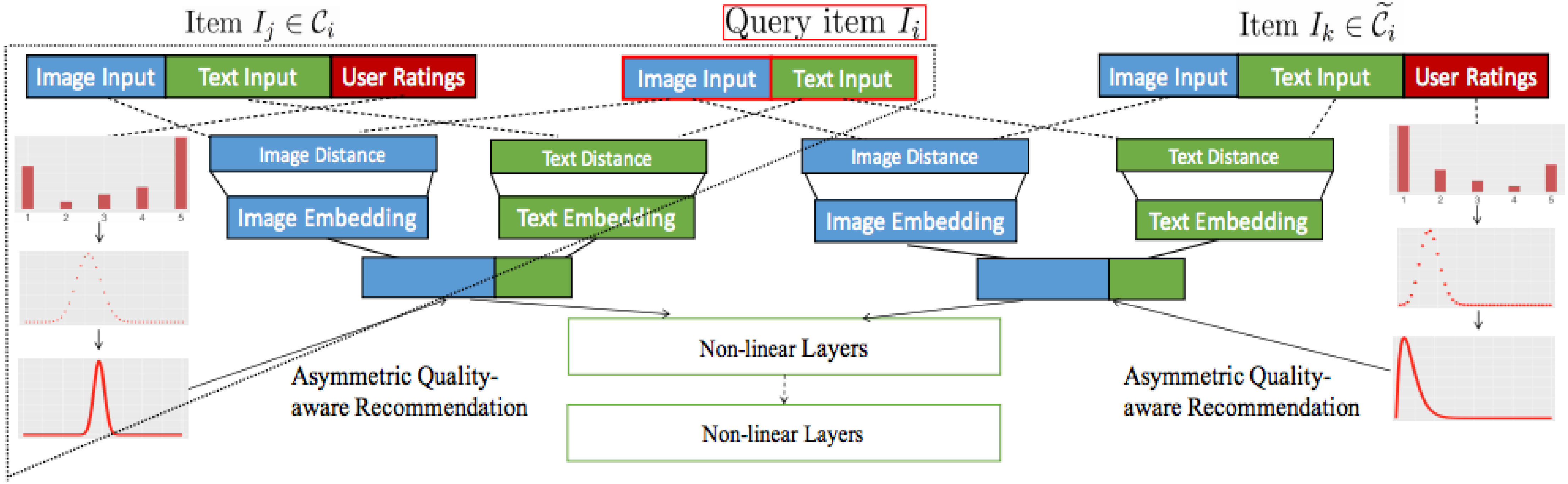
Other Methods

- Quality-Aware Neural Complementary Item Recommendation

$$P_{comp} = \sigma\left(d(a_f, c_f)\right) = \frac{1}{1+e^{d(a_f, c_f)-\eta}} \text{ [4]}$$

Challenges:

- Transitivity



Challenges

- Cold-start items
- Diversity
- Differentiate between similar and complementary items

Dataset

- Amazon Clothing, Shoes, and Jewelry data[2]
  - Category information and title
  - Complementary pairs: bought together by users from different categories
  - Similar pairs: items that lie in same category
  - Negative items: Randomly sample items which do not meet above criteria
  - Quads: anchor, complementary, similar, negative items
  - Train quads: 3.3M, Test quads:0.3M

https://github.com/mansimane/quadnet-comp-sim

Amazon Dataset Attribute Availability

Attribute	Coverage
Image	99.99
Description	5.68
Title	99.95
Price	38.29
Brand	6.25

Example Quad

Anchor	Complementary	Similar	Negative
			
Lee Dungarees Men’s Big, Tall Carpenter Jean	Key Apparel Men's Big-Tall Short Sleeve Heavyweight Pocket Tee Shirt	Wrangler Men's Rugged Wear Relaxed Straight Fit Jean	Black and White Herringbone Wool Suiting Extra Long Tie

Motivation

- Why not just optimize for complementary items?

$$P_{comp} = \sigma\left(d\left(a_f, c_f\right)\right) = \frac{1}{1+e^{d\left(a_f, c_f\right)-\eta}} \text{ [4]}$$

a

Wonder Nation Twist-Front Graphic T-Shirt

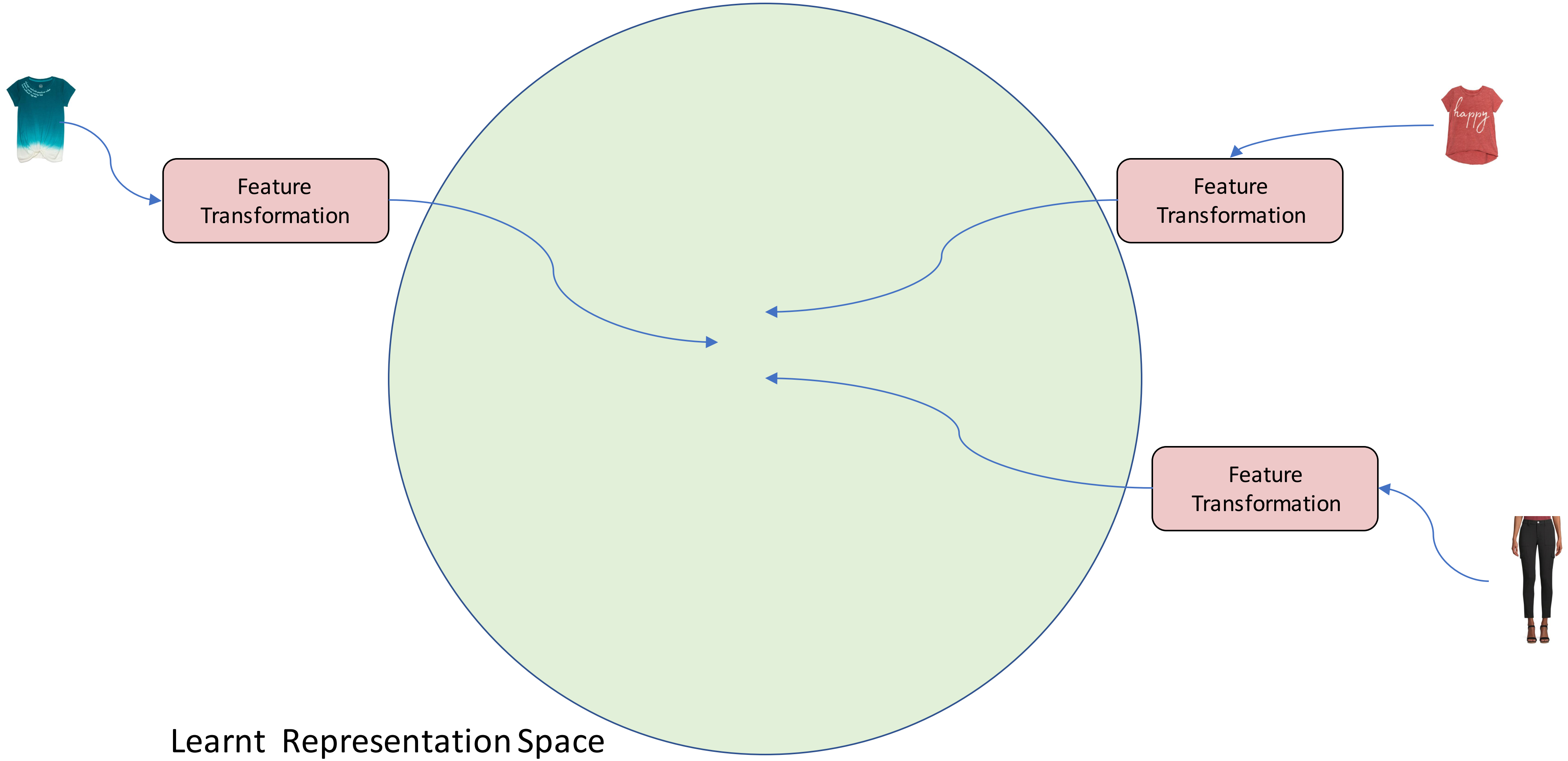
c

Women's Knit Skinny Cargo Pant

b

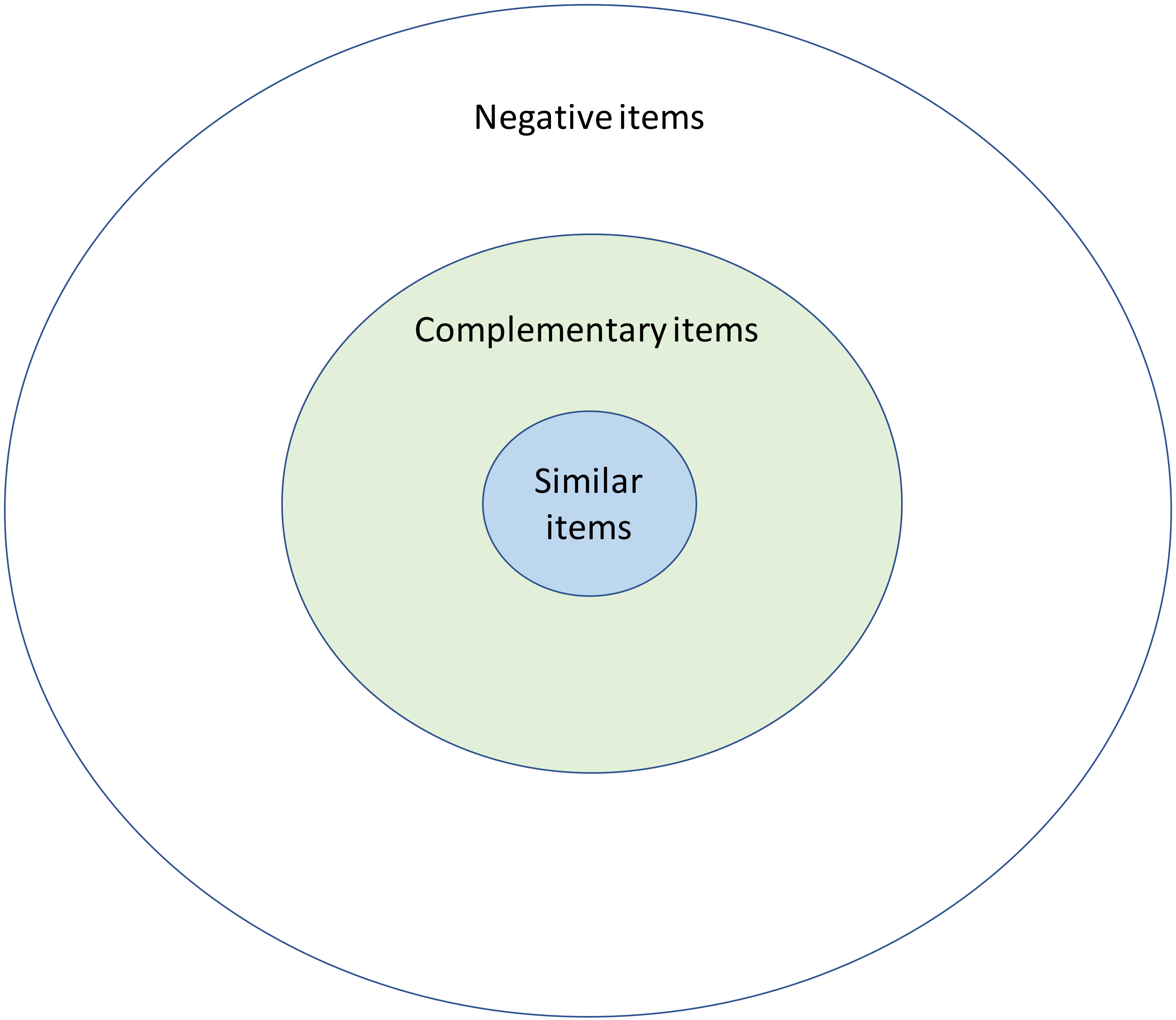
Wonder Nation Graphic Hi-Lo T-Shirt

Motivation


Feature Transformation
Feature Transformation
Feature Transformation
Learnt Representation Space

Goal

- Learn representation space which can differentiate between similar and complementary items

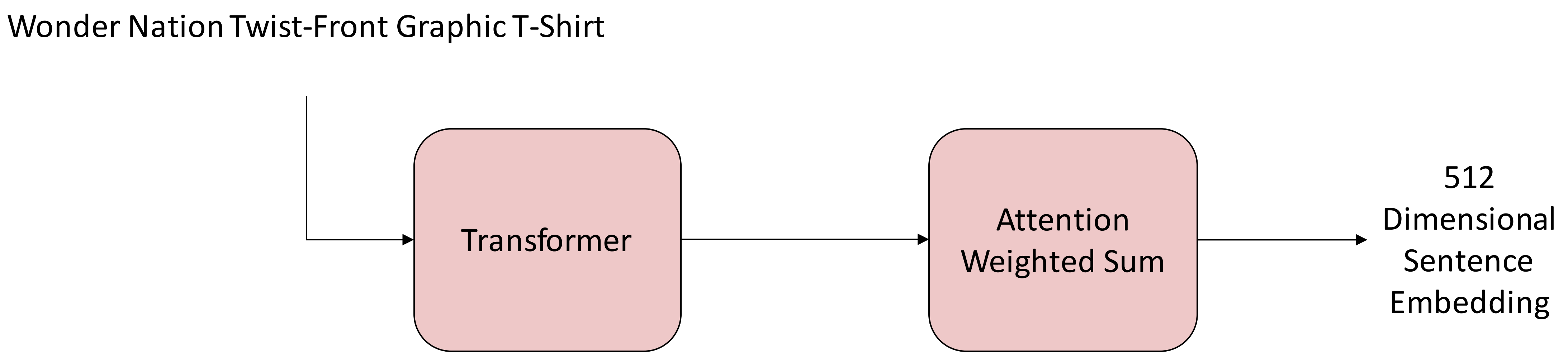


Problem Formulation

- a: Anchor item
- c: Complementary item to anchor item
- s: Similar item to anchor item
- n: Negative item to anchor item
- a_f' , c_f' , s_f' , n_f' : Normalized learnt feature representation for a, c, s, n

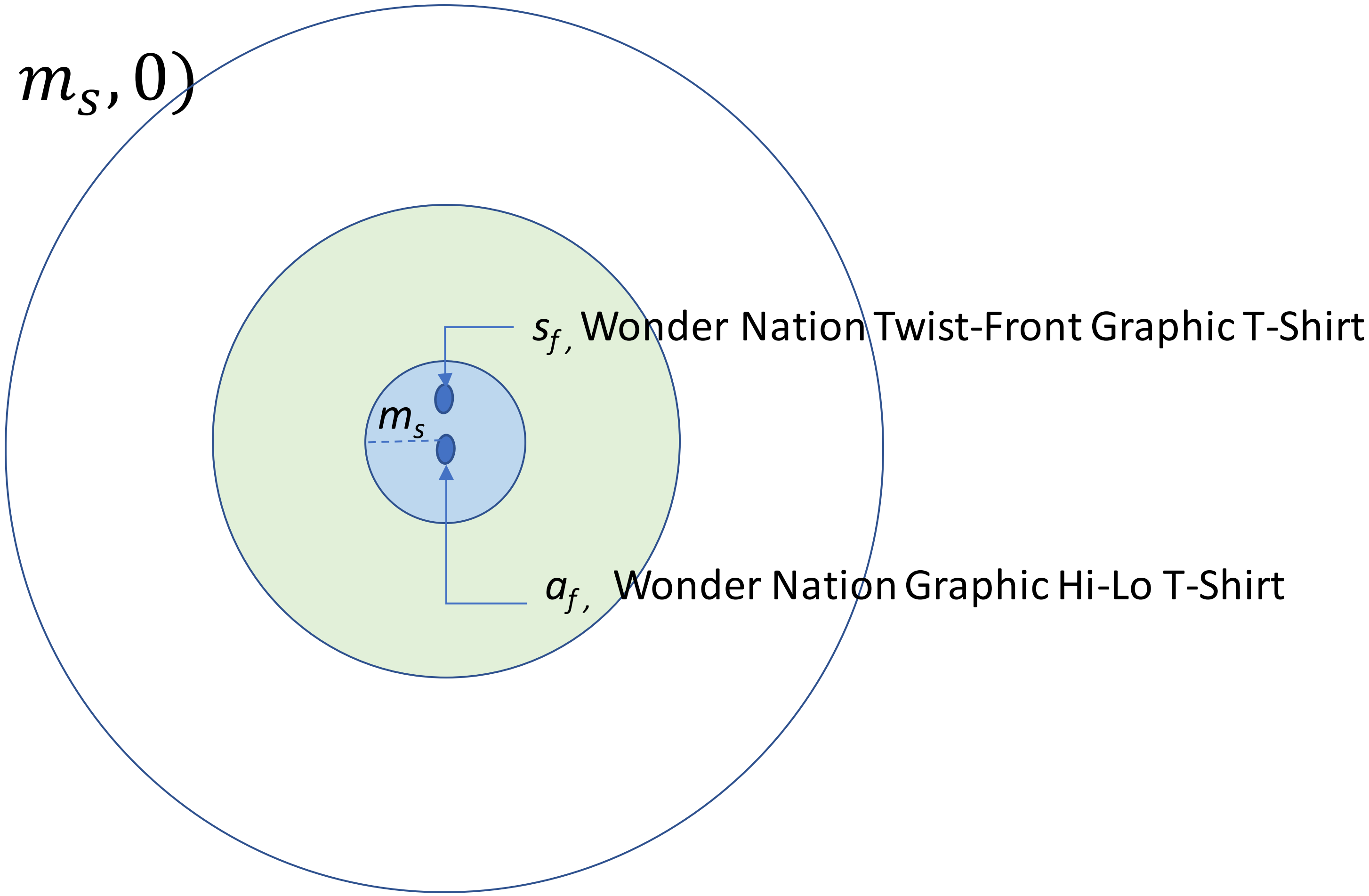
Feature Representation

- Universal Sentence Encoder [1]



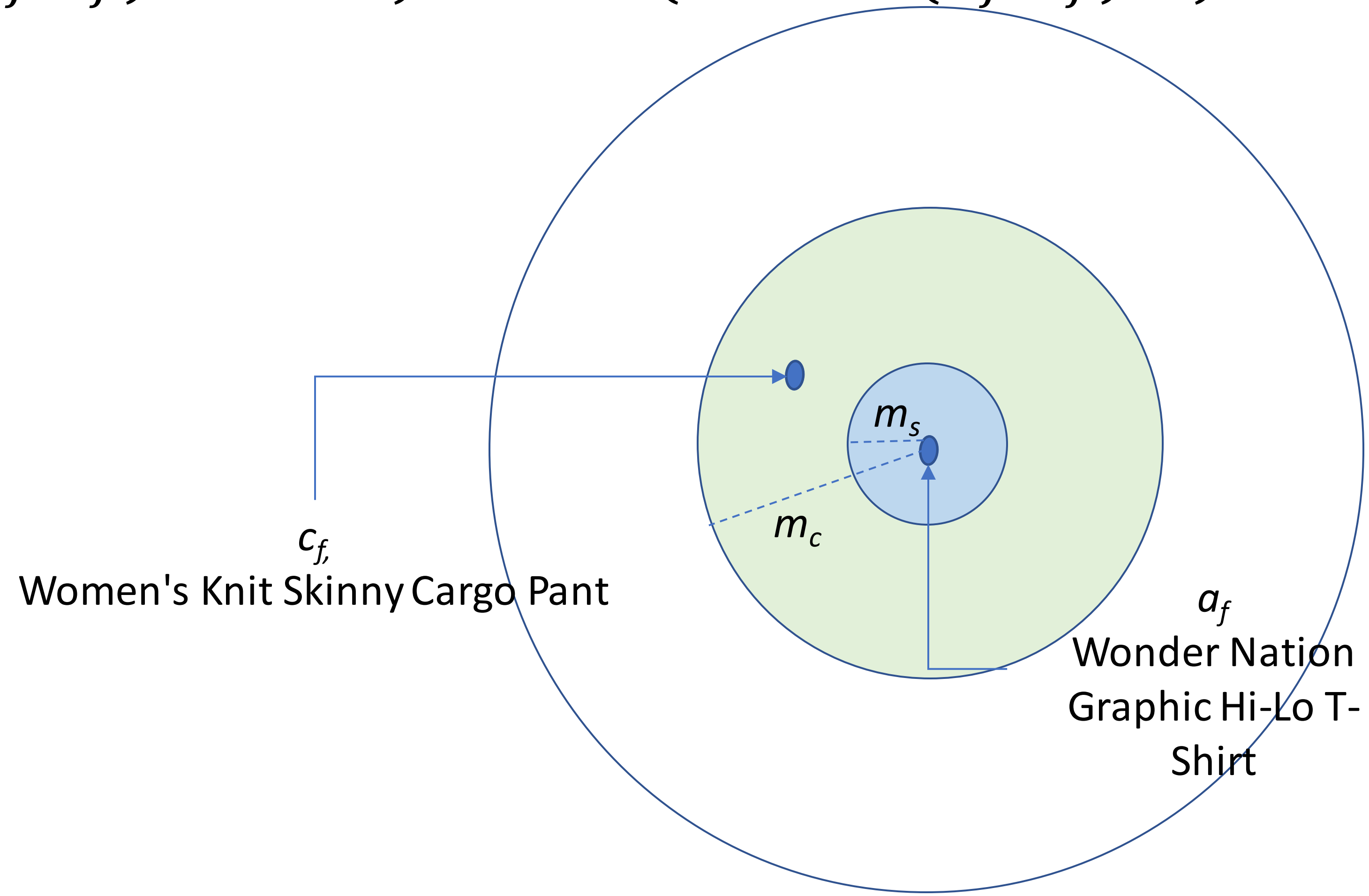
Similarity Loss

- $L_{sim} = \max(d(a_f' - s_f') - m_s, 0)$



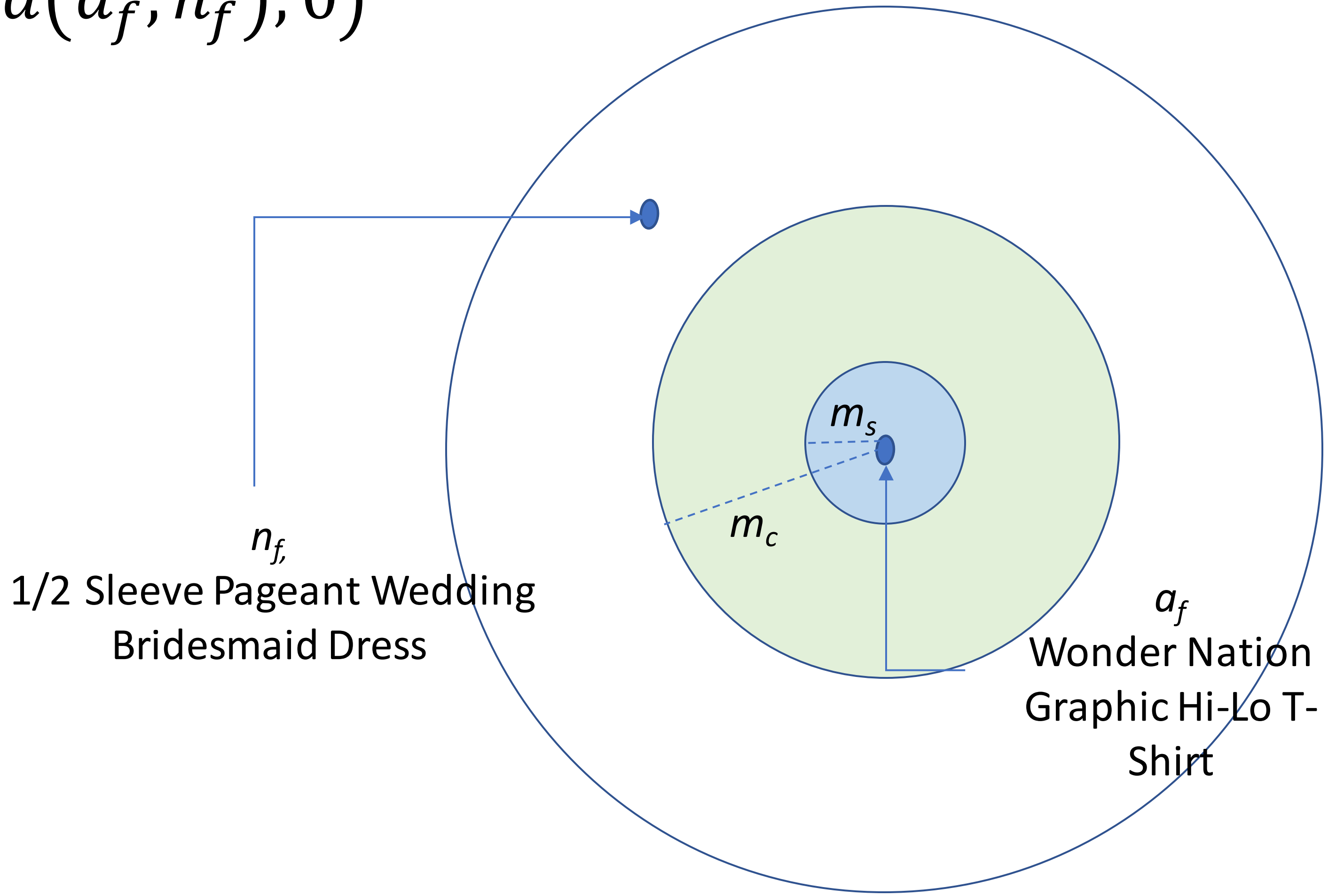
Complementary Loss

- $L_{comp} = \max\left(d\left(a_f', c_f'\right) - m_c, 0\right) + \max\left(m_s - d\left(a_f', c_f'\right), 0\right)$



Negative Loss

- $L_{neg} = \max(m_n - d(a'_f, n'_f), 0)$

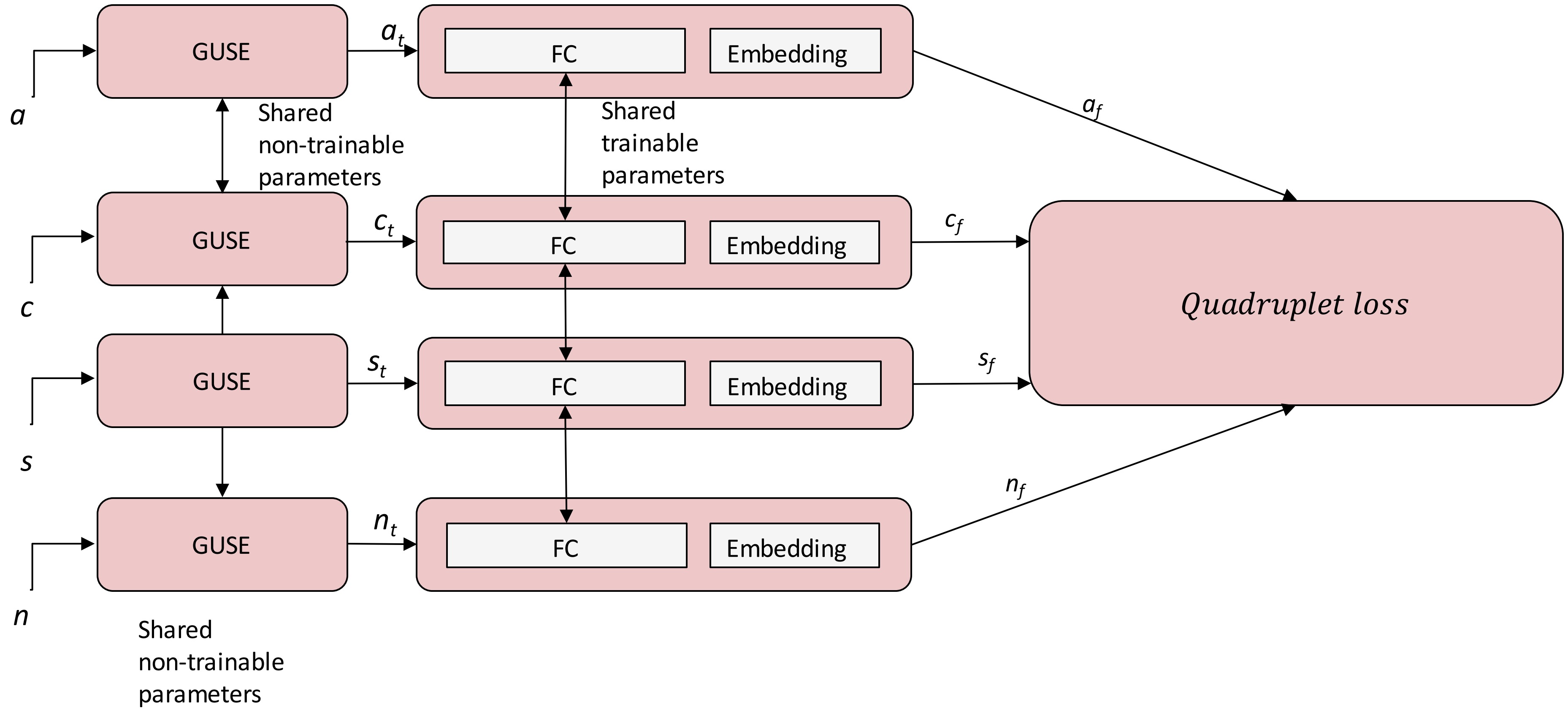


Quadruplet Loss

- $L_{sim} = \max(d\left(a'_f - s'_f\right) - m_s, 0)$
- $L_{comp} = \max\left(d\left(a'_f, c'_f\right) - m_c, 0\right) + \max\left(m_s - d\left(a'_f, c'_f\right), 0\right)$
- $L_{neg} = \max\left(m_n - d\left(a'_f, n'_f\right), 0\right)$

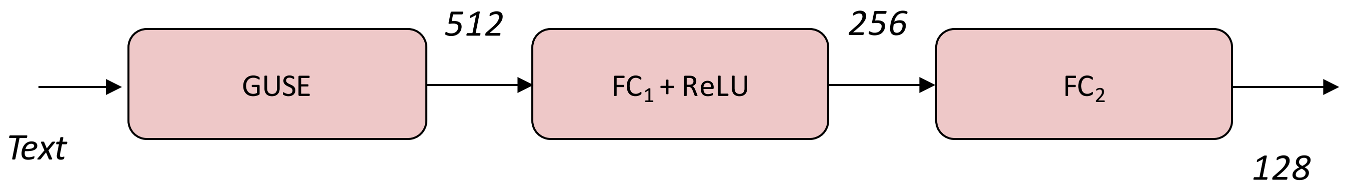
- $L_{quad} = L_{sim} + L_{comp} + L_{neg} + \lambda L_{l2}$

Architecture

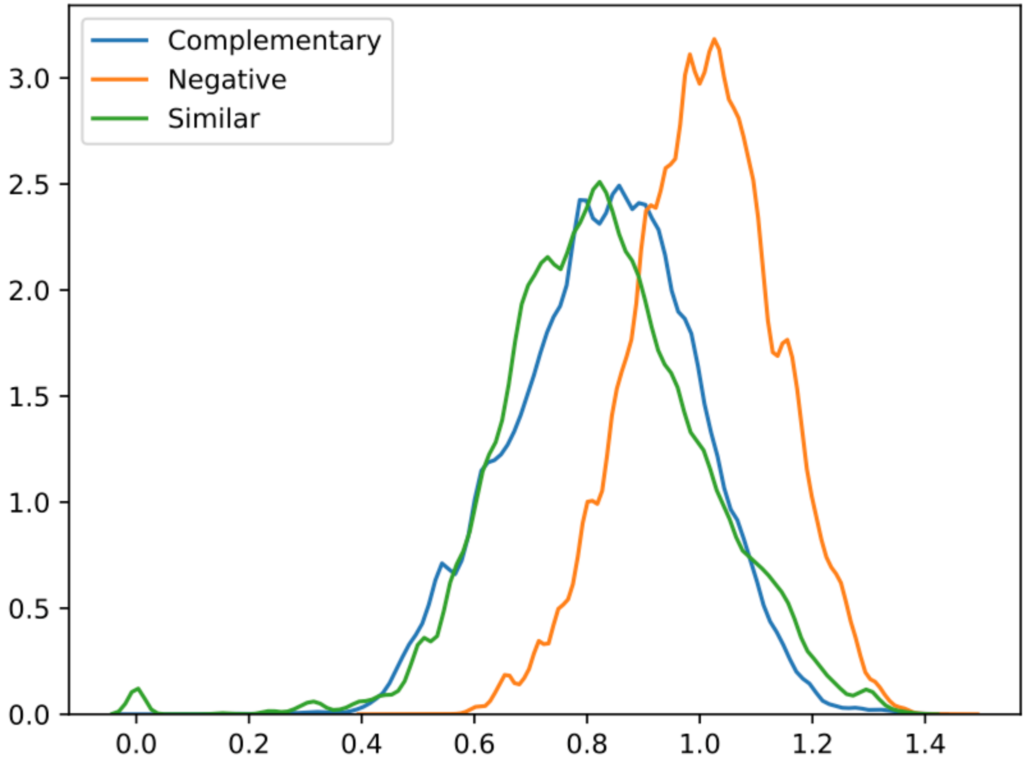


Hyperparameters

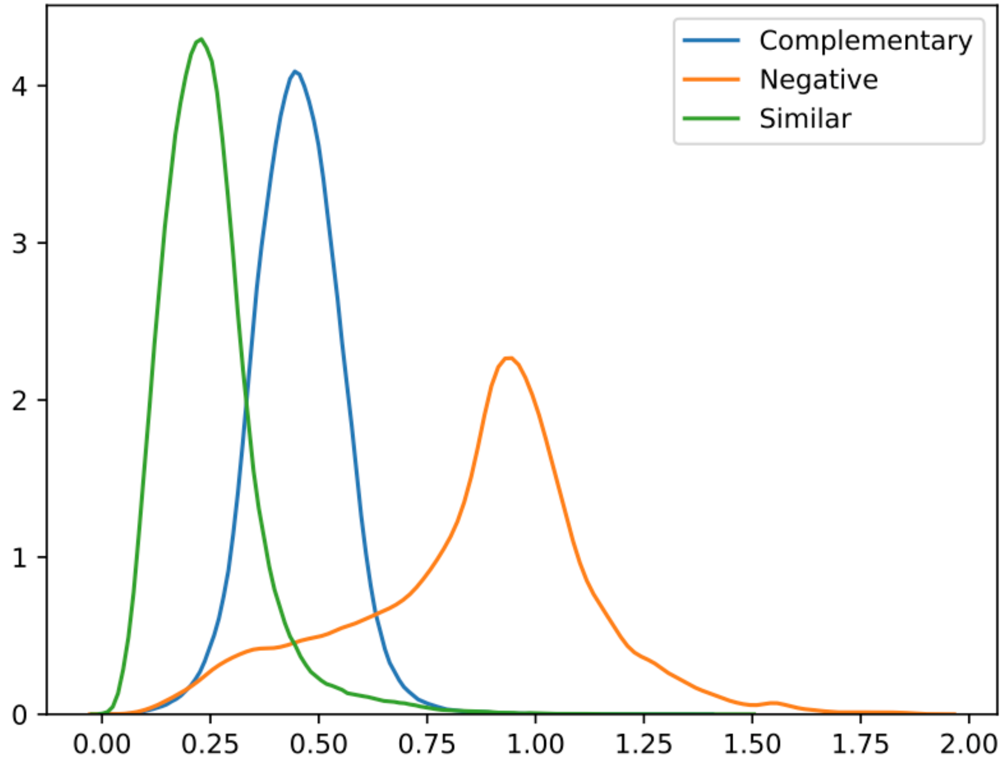
- Input feature dimension: 512
- Epochs: 50
- Weight Initialization: Xavier
- Learning rate: 0.001
- m_s : 0.1
- m_n: 0.4
- m_c :0.8
- Mapping function:



Distance Distribution



Before Training



After Training

Distance Distribution

	Similar		Complementary		Negative	
	Mean	Std Dev	Mean	Std Dev	Mean	Std Dev
Train Data Before training	0.82119	0.17611	0.81975	0.15910	0.99804	0.13286
Test Data Before training	0.82752	0.17853	0.83086	0.15937	1.0037	0.12949
Train Data After training	0.24069	0.11226	0.45845	0.11485	0.86774	0.27724
Test Data After training	0.24772	0.11485	0.45181	0.09963	0.86023	0.27182

Accuracy

Method	Ranking Acc	Complementary Acc	Similarity Acc
Universal Sentence Encoder	37.68	-	-
Veit et al. [2]	14.92	91.05	56.45
Quadruplet Network	**67.15**	86.92	**68**

- Ranking accuracy is calculated as: $d_s < d_c < d_n$
- Complementary Accuracy: $margin_s < d_c < margin_c$
- Similarity Accuracy: $d_s < margin_s$

Future Work

- Modelling asymmetry between relationships
- Large scale experiments on Amazon dataset with more evaluation metrics
- Clustering analysis on learnt embedding space

References

1. Cer, Daniel, et al. "Universal sentence encoder." *arXiv preprint arXiv:1803.11175* (2018).
2. Veit, Andreas, et al. "Learning visual clothing style with heterogeneous dyadic co-occurrences." *Proceedings of the IEEE International Conference on Computer Vision*. 2015.
3. Vasileva, Mariya I., et al. "Learning type-aware embeddings for fashion compatibility." *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.
4. Zhang, Yin, et al. "Quality-aware neural complementary item recommendation." *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 2018.
5. McAuley, Julian, Rahul Pandey, and Jure Leskovec. "Inferring networks of substitutable and complementary products." Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, 2015
6. Mengting Wan, Di Wang, Jie Liu, Paul Bennett, and Julian McAuley. Representing and recommending shopping baskets with complementarity, compatibility and loyalty. In CIKM, pages 1133–1142, 2018.

Thank You

Stepen Guo: sguo@walmartlabs.com
Mansi Mane: mansi.mane@walmartlabs.com,
mansimane5@gmail.com