Neural Compatibility Ranking for Text-based Fashion Matching

Suthee Chaidaroon
schaidaroon@scu.edu
Santa Clara University
Santa Clara, CA, USA

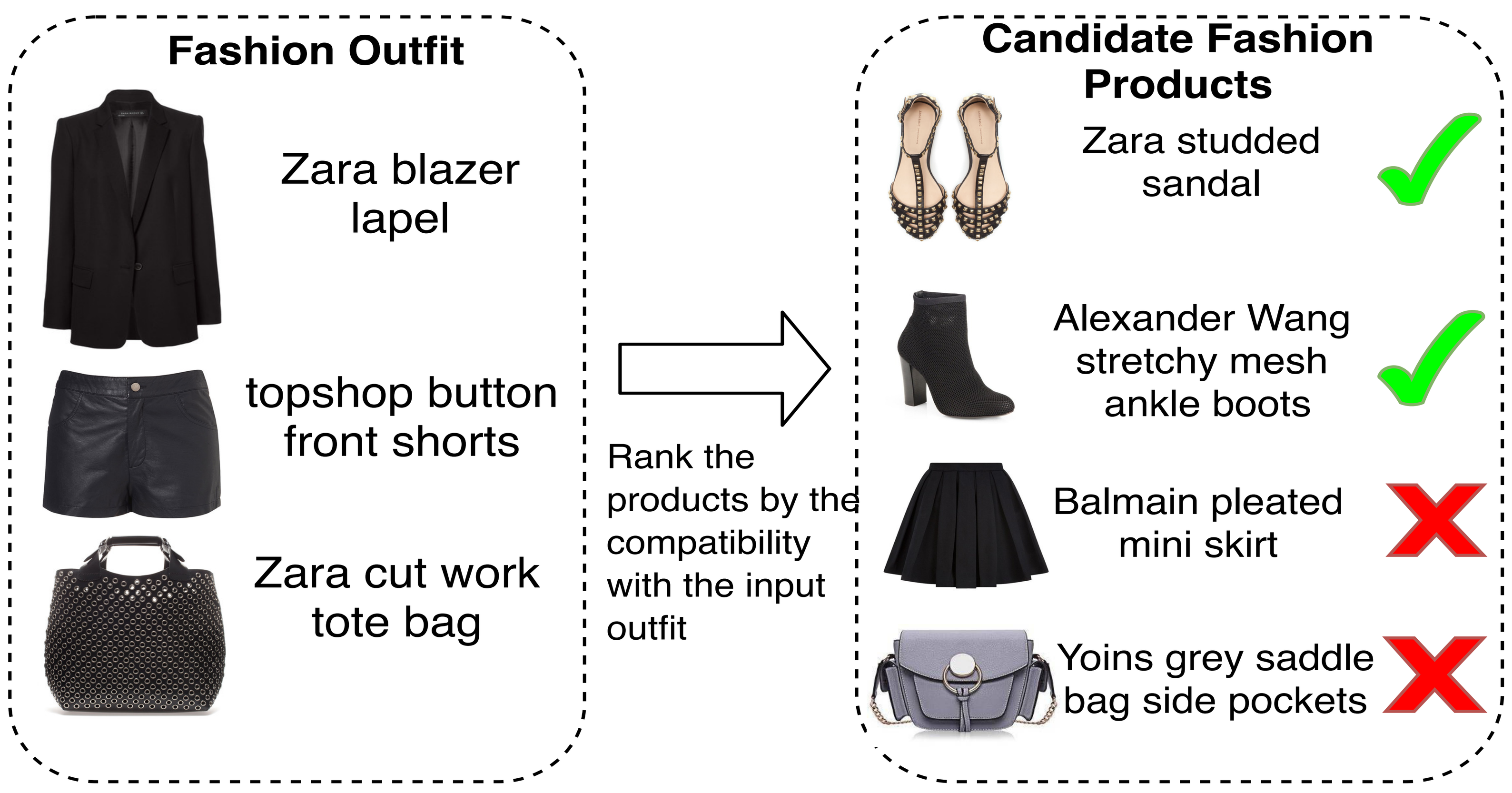
Yi Fang
yfang@scu.edu
Santa Clara University
Santa Clara, CA, USA

Min Xie
mxie@walmartlabs.com
Walmart Labs
Sunnyvale, CA, USA

Alessandro Magnani
amagnani@walmartlabs.com
Walmart Labs
Sunnyvale, CA, USA

Text-based Fashion Matching Problem

Given an outfit as a set of text descriptions, each represents a fashion product; we want to learn a ranking function to retrieve the top K most compatible fashion items from the candidate set. In this work, we do not use image information.



What type of information we can get from text descriptions?

- **Lexical matching information**: products that share the **same words/terms** are related, but this approach may suffer from the **vocabulary gap**.
- **Semantic matching information**: products from the same outfit are semantically similar, but the co-occurrence information pays more attention to a **high-frequency** co-occurrence and often ignores **non-popular** products.

What are our hypotheses?

- The **outfits** curated by professional fashion bloggers consist of **compatible** fashion products.
- Any **product** from the **same outfit** is **more compatible** than the products from **other outfits**.
- Product's text **descriptions** contain a useful **compatibility** signal.

Contributions

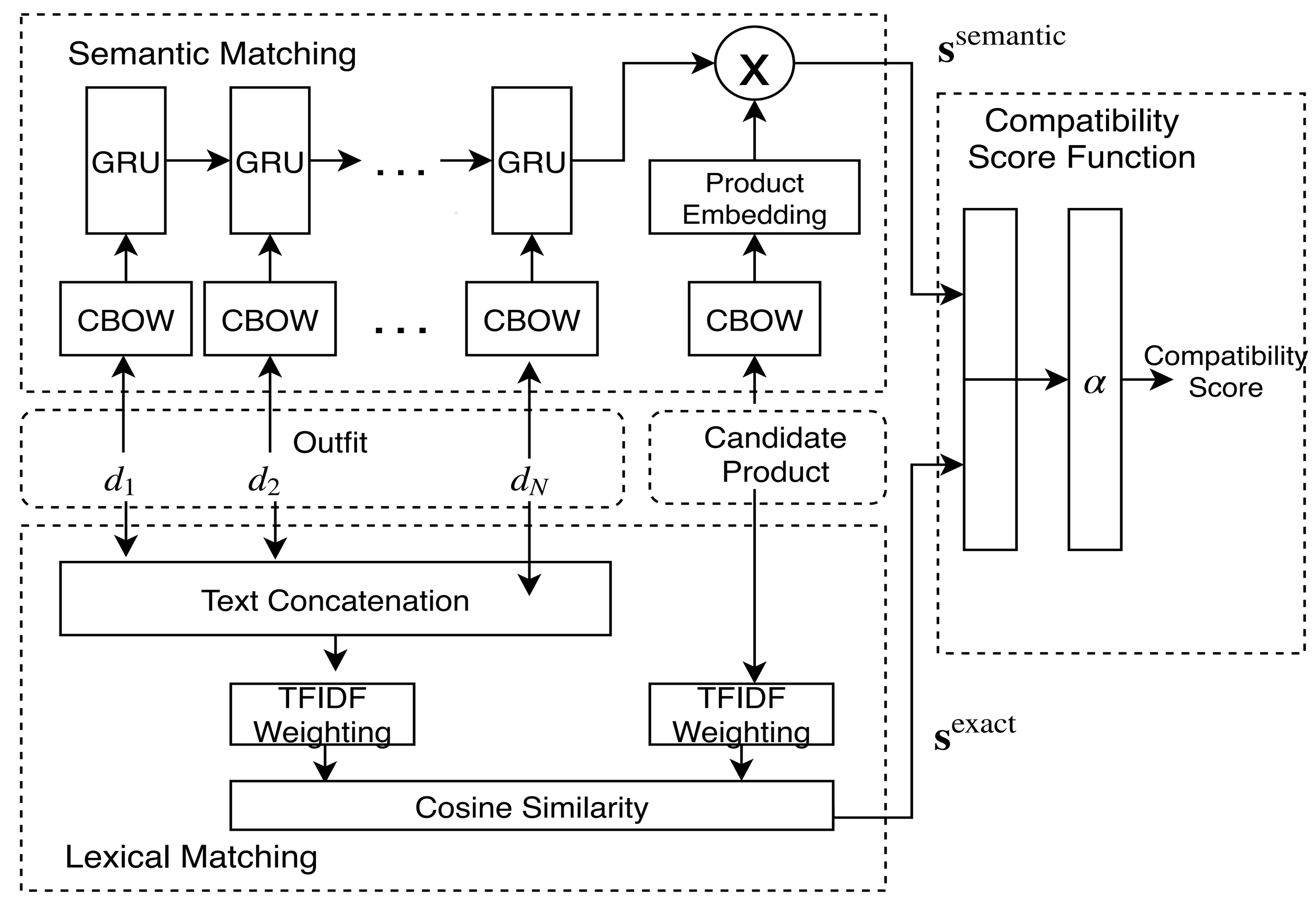
- Propose a novel **neural ranking architecture** for fashion matching by capturing both **common** and **rare fashion concepts** via **semantic** and **lexical** matching.
- Demonstrate that a **product description** contains rich information about product **compatibility** and has not been fully utilized by the prior works. Our work can achieve **state-of-the-art** results on **fashion matching task** and **comparable** performances with the models that use **both image and text** data.

Neural Compatibility Ranking model (NCR)

The model consists of three components:

- **Semantic matching component**: Capture **common** fashion concepts
  - Use pre-trained GloVe word embedding and CBOW to represent a product description. We freeze the embedding lookup table.
  - Use GRU to aggregate the product vectors from the input outfit and use the last hidden state as the output vector of 32 dimensions.
  - Use 3-layer feedforward networks with 300, 128, and 32 neurons to transform a target product into a vector of 32 dimensions.
  - Compute a Hadamard product between the outfit and target product.
- **Lexical matching component**: Capture **rare** fashion concepts
  - Concatenate all product descriptions in the outfit and represent it as a TFIDF vector.
  - Represent a target product description as a TFIDF vector.
  - Compute a cosine distance of the outfit and target product TFIDF vectors.
- **Compatibility score function**
  - Concatenate the output vectors generated by the semantic and lexical matching component. Use a 1-layer feedforward with a sigmoid function to output a compatibility score.

The Model Architecture of Neural Compatibility Ranking



Model Training

- Train the model as a **point-wise** ranking model by minimizing a **binary cross entropy** loss.
- Use **Polyvore** fashion outfit dataset, which contains **21,889 fashion outfits** (**17,316** for **training**, **1,497** for **validation**, and **3,076** for **testing**).
- A **positive outfit-product** pair is created by sampling one product from the outfit and remove it from the same outfit while a **negative outfit-product** pair is created by picking a product randomly from other outfits within the same training set.

Top-n Recommendation Task

Model	Data	MRR	HR@1	HR@5	HR@10
QL [1]	Text	0.1762	0.1191	0.2073	0.2610
BM25 [2]	Text	0.1649	0.1135	0.2032	0.2594
DSSM [3]	Text	0.1743	0.0846	0.2285	0.3378
K-NRM [4]	Text	0.1707	0.1150	0.1975	0.2584
SiameseNet [5]	Text+Image	0.2559	0.1320	**0.3667**	0.5167
BiLSTM+VSE [6]	Text+Image	0.2256	0.0975	0.3424	**0.5224**
NCR (Ours)	Text	**0.2678**	**0.1547**	0.3626	0.5137

The table above shows the ranking performance of our model and baseline methods on the top-n recommendation task. For each testing outfit, we leave one product out and use it as a positive item. Then, we sample 100 products from other outfits as negative items.

Ranking Performance for Each Component

Model	MRR	HR@1	HR@5	HR@10
Semantic Matching	0.1708	0.0603	0.2584	0.4296
Lexical Matching	0.1794	0.1212	0.2104	0.2610
NCR (Combine)	**0.2678**	**0.1547**	**0.3626**	**0.5137**

The table above shows the ranking performances of all components in NCR.

Conclusions

We propose a neural compatibility ranking model for fashion matching. The model takes the advantages of **semantic matching** and **lexical matching** as it significantly **outperforms** the text-based ad-hoc retrieval models while achieves **comparable** results with the baselines that use both product **image** and **text descriptions**. A **combination** of **semantic** and **lexical matching** is a crucial component for extracting product compatibility information from text data.

References

[1] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to information retrieval. SIGIR, 2001.
[2] Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. Foundations and Trends in Information Retrieval, 2009.
[3] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. CIKM 2013.
[4] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. End-to-end neural ad-hoc ranking with kernel pooling. SIGIR 2017.
[5] Andreas Veit, Balazs Kovacs, Sean Bell, Julian McAuley, Kavita Bala, and Serge Belongie. Learning visual clothing style with heterogeneous dyadic co-occurrences. ICCV 2015.
[6] Xintong Han, Zuxuan Wu, Yu-Gang Jiang, and Larry S Davis. Learning fashion compatibility with bidirectional lstms. ACM on Multimedia Conference, 2017.