Contents

1 Introduction

Deep CBRS Amar is a deep architecture for content based recommendations. This architecture takes in input an user u and an item i and returns a score $s(u, i)$ which represents the probability that the user would like that particular item. The way in which users and items are modeled is essential. An exogenous approach is used, in which an external knowledge base is exploited for representing users and items.

This work is an extension of a previous work. The latter models users and items by mean of embeddings, in particular:

- knowledge graph embedding, which models users and items exploiting the relationships they have in a knowledge graph;
- plot embeddings, which models items exploiting textual information about items. BERT [1] is used for this purpose.

Deep CBRS Amar comes with three variants, using different model of embeddings and combining knowledge graph embeddings and BERT embeddings.

Starting from the previous work, the main objectives of this study are:

1. use two different knowledge graphs for representing users and items;
2. improve the architecture of the deep learning model.

For experimentation, this work has used as dataset the MovieLens dataset, which contains movies ratings. BERT embeddings have been inherited from the previous work.

2 Knowledge Graph Embedding

Knowledge graph (KG) embedding is to embed components of a KG including entities and relations into continuous vector spaces, so as to simplify the manipulation while preserving the inherent structure of the KG [4].

Formally, supposing we are given a KG consisting of n entities and m relations. Facts observed in the KG are stored as a collection of triples $D+ = \{(h, r, t)\}$. Each triple is composed of a head entity $h \in E$, a tail entity $t \in E$, and a relation $r \in R$ between them. Here, E denotes the set of entities, and R the set of relations. KG embedding aims to embed entities and relations into a low-dimensional continuous vector space, so as to simplify computations on the KG.

A typical KG embedding technique generally consists of three steps: (i) representing entities and relations, (ii) defining a scoring function, and (iii) learning entity and relation representations. The first step specifies the form in which entities and relations are represented in a continuous vector space. Entities are usually represented as vectors, i.e., deterministic points in the vector space. Relations are typically taken as operations in the vector space, which can be represented as vectors. Then, in the second step, a scoring function $f_r(h, t)$ is defined on each fact (h, r, t) to measure its plausibility. Facts observed in the KG tend to have higher scores than those that have not been observed. Finally, to learn those entity and relation representations (i.e., embeddings), the third step solves an optimization problem that maximizes the total plausibility of observed facts (i.e., facts contained in $D+$).

2.1 Knowledge Graph Embedding Models

In literature, embedding techniques are categorized into two groups: translational distance models and semantic matching models.

2.1.1 Translational Distance Models

Translational distance models exploit distance-based scoring functions. They measure the plausibility of a fact as the distance between the two entities, usually after a translation carried out by the relation. The translational models used in the work are:

- *TransE,* which represents both entities and relations as vectors in the same space, say $\mathbb{R}^d$. Given a fact (h, r, t), the relation is interpreted as a translation vector r so that the embedded entities h and t can be connected by r with low error, i.e., $h + r \approx t$ when the fact holds. The scoring function is then defined as the (negative) distance between $h + r$ and t, i.e., $f_r(h, t) = -||h + r - t||_{1/2}$.

- *TransH,* which is introduced to overcomes some problems of TransE, such as the problem of dealing with 1-to-N, N-to-1, and N-to-N relations. It models entities again as vectors, but each relation r as a vector r on a hyperplane with w_r as the normal vector. Given a fact (h, r, t), the entity representations h and t are first projected onto the hyperplane, resulting in $h_\perp = h - w_r^\top h w_r$, $t_\perp = t - w_r^\top t w_r$. The projections are then assumed to be connected by r on the hyperplane with low error if (h, r, t) holds, i.e., $h_\perp + r \approx t_\perp$. The scoring function is accordingly defined as $f_r(h, t) = -||h_\perp + r - t_\perp||_2^2$.

- *TransD,* which introduces additional mapping vectors w_h , $w_t \in \mathbb{R}^d$ and $w_r \in \mathbb{R}^k$, along with the entity/relation representations $h, t \in \mathbb{R}^d$ and $r \in \mathbb{R}^k$. Two projection matrices M_r^1 and M_r^2 are accordingly defined as $M_r^1 = w_r w_h^\top + I$, $M_2^r = w_r w_t^\top + I$. These two projection matrices are then applied on the head entity h and the tail entity t respectively to get their projections.

2.1.2 Semantic Matching Models

Semantic matching models exploit similarity-based scoring functions. They measure plausibility of facts by matching latent semantics of entities and relations embodied in their vector space representations. The semantic models used in the work are:

- *RESCAL,* which associates each entity with a vector to capture its latent semantics. Each relation is represented as a matrix which models pairwise interactions between latent factors. The score of a fact (h, r, t) is defined by a bilinear function $f_r(h,t) = h^\top M_r t = \sum_{i=0}^{d-1} \sum_{j=0}^{d-1} [M_r]_{ij} \cdot [h]_i [t]_j$, with $h, t \in \mathbb{R}^d$ are vector representations of the entities and $M_r \in \mathbb{R}^{d \times d}$ is a matrix associated to the relation r.

- *DistMult,* which semplifies RESCAL by restricting M_r to diagonal matrices.

- *HolE,* which combines the expressive power of RESCAL with the efficiency and sim- plicity of DistMult, representing both entities and relations in $\mathbb{R}^d$.

For more informations about each Knowledge Graph Embedding model, in terms of constraints, space complexity and time complexity, and the training process, see [4].

2.2 OpenKE

OpenKE is an open-source Framework for Knowledge Embedding [2]. It comes in two flavours: one implented with TensorFlow and one with PyTorch. This work has used the TensorFlow one. It also uses C++ to implement some underlying operations such as data preprocessing and negative sampling. OpenKE supports the most well-known models both transationals and semantic-based.

For training with OpenKE refers to the documentation on the official repository. For the embedding generated in this work with OpenKE refers to the Results section.

2.3 Embedding Generation

The previous work has used only one type of knowledge graph, where nodes are *item* and *user* whereas relations are *like* and *dislike*. In this work two types of knowledge graph are used: i) which contains items and their properties and ii) which contains user, item and items' properties. i) is a bipartite graph, while ii) is a tripartite graph. Both in i) and ii) the embedding vector of the user is computed as the centroid vector of the embedding of the items he/she likes.

Not all the properties of the items have been injected because not all the properties are useful. A manual approach has been adopted, choosing three subset and then selecting the most promising. The embeddings have been generated on dimension 256, 512 and 768. The knowledge graph embedding techniques used in this work are:

- TransE, TransD and TransH, belonging to the translational category.

- RESCAL, DistMult and HolE, belonging to the semantic category.

For space complexity reason, RESCAL has been generated only on dimension 256, and 512.

3 Deep CBRS Amar

Deep CBRS Amar is a deep learning architecture which jointly learns two embeddings representing the items to be recommended as well as the preferences of the user. It is a variant of AMAR (Ask Me Any Rating), which adopts Long Short Term Memory (LSTM) [3]. In Deep CBRS Amar the LSTM layer is removed, and a more classical approach is used. The new variants allow the use of different types of embeddings, because the classic AMAR learn items embeddings only from item textual description. Deep CBRS Amar comes in three variant, which are explained below.

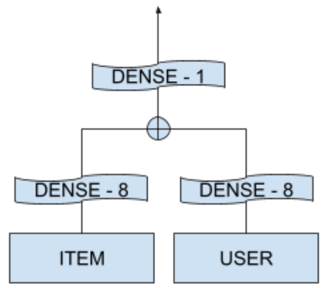

Figure 1: First Deep CBRS Amar variant architecture.

3.1 First Amar Variant

The first proposed AMAR variant involves using AMAR base architecture considering as input:

- KG embeddings
- BERT embeddings
- KG and BERT embeddings (by varying users and items)

The framework takes the user embedding and the item embedding as input and gives them to the neural network to allow the training phase to be performed. The two inputs (uses and item embeddings) are projected to two dense layers (with 8 hidden units and ReLU as activation function) and then they are concatenated through an additional layer. Finally, the last layer uses a sigmoid as activation and computes the final score between that user and that specific item. As loss function a binary crossentropy has been employed, while Adam has been used as optimization method.

3.2 Second Amar Variant

The second proposed AMAR variant has involved two configurations:

1. In the first case the framework takes on one hand KG and BERT item embeddings and, on the other, KG and BERT user embeddings. For each pair, the concatenation of the dense layers of the respective embeddings is performed. This concatenation step is also performed later, on the previous dense layers. Finally, the last layer uses a sigmoid as activation and computes the final score between that user and that specific item.
2. The second configuration works exactly as the first one, but with just one difference: here the inputs are combined in a different way. The framework takes on one hand KG item and user embeddings and, on the other, BERT item and user embeddings.

All the layers but the last, in both variants, consist of 8 hidden units and use a ReLU as activation function. As loss function a binary crossentropy has been employed, while Adam has been used as optimization method.

3.3 Third Amar Variant

The third proposed variant, takes as input on one hand KG item and user embeddings and, on the other, BERT item and user embeddings. This configuration can vary by considering: the number and position of attention blocks and the possibility to have a final dropout. Attention mechanisms represent an attempt to implement the same action of selectively concentrating on a few relevant things, while ignoring others in deep neural networks. Dropout is a regularization technique for reducing overfitting in neural networks by preventing complex adaptations on training data. So, from this variant there are three possible configurations:

- one attention layer after the concatenation.
- three attention layers;
- one attention layer plus a dropout layer.

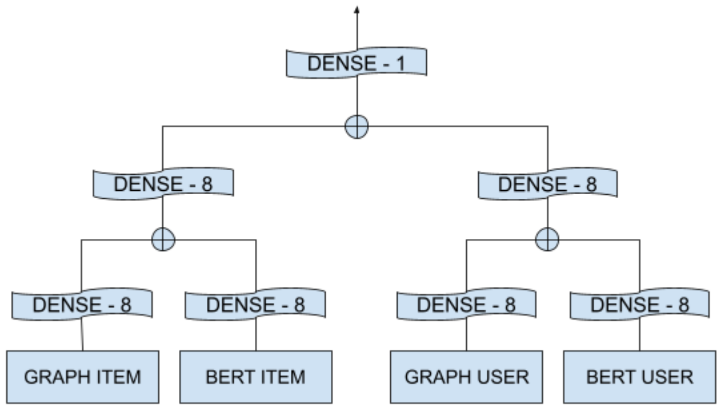

Figure 2: Second Deep CBRS Amar variant architecture.

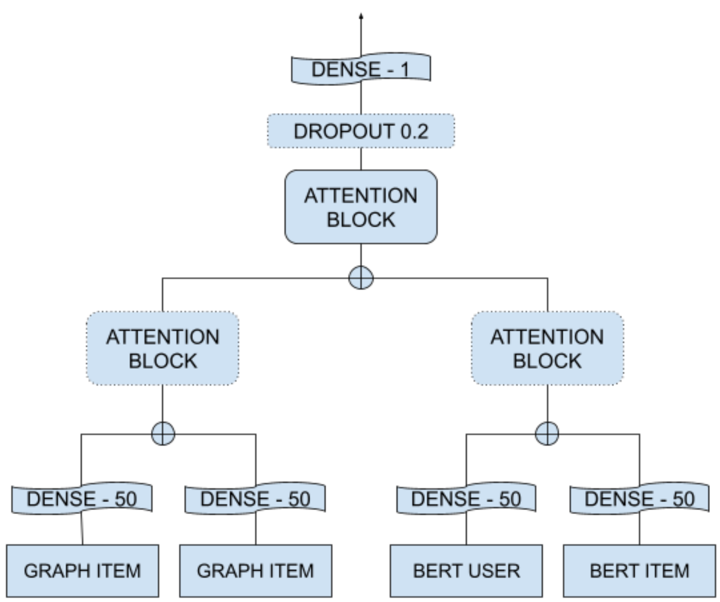

Figure 3: Third Deep CBRS Amar variant.

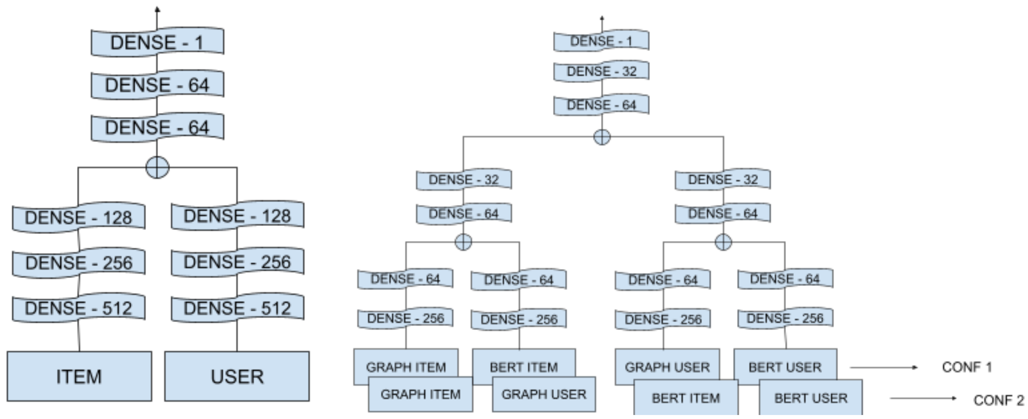

Figure 4: First and second variant revisited.

4 Deep CBRS Amar Revisited

As said in the introduction section, one of the objectives of this work is to improve the Deep CBRS Amar architecture. This work suggests two improvement on the first and second Amar variants, with changes on the number of dense layers, making the overall architecture more deeper, see Fig. 4. Some experiments have been done also for optimizing the EXTENDED architecture. This work ended up with the architecture in Fig. 5.

5 Experiment Settings

5.1 Knowledge Graph Embedding

For each KG model category a specific parameter configuration has been used.

	Transational Model	Semantic Model
Batch size	50	50
Learning rate	0.001	0.0001
Epochs	1000	1000
Optimizer	SGD	Adam

The best configuration has been achieved using a trial and error method and comparing the results.

5.2 Deep CBRS Amar

For Deep CBRS Amar the following parameter settings has been used:

- **Optimization method**: Adam, with alpha equals to 0.9
- **Learning rate**: 0.001
- **Batch size**: 512
- **Epochs**: 25/30

This settings has been used for all variants.

5.3 Dataset

In this study has been used the Movielens 1 Million (ML1M) [movielens citation], that is a dataset which contains 1M+ ratings from 6,000 users on 4,000 movies. The 80/20 training/test split method has been used for training with Deep CBRS Amar.

Also, an augmented dataset has been used, with movies properties linked to each movie. The dataset comes a N-Triples format, with URI from DBpedia, which builds a bipartite graph. Because not all the item properties may be useful, three subset candidates has been selected. They are:

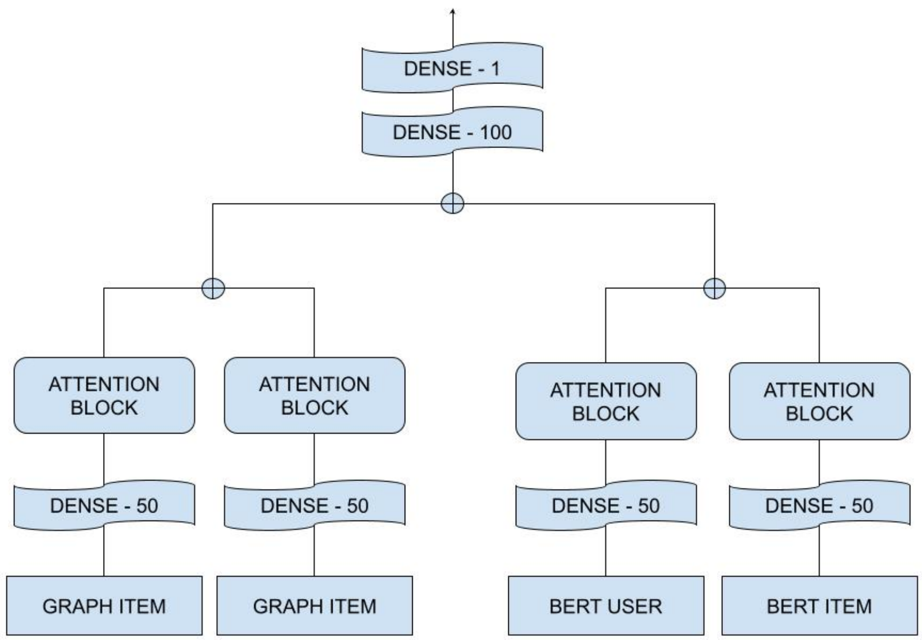

Figure 5: Third variant revisited.

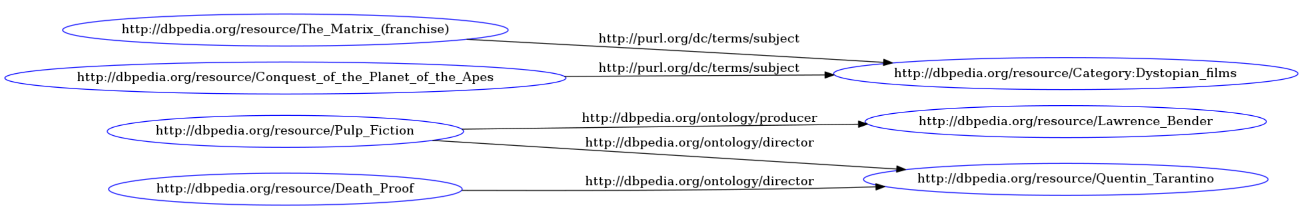

Figure 6: A snapshot from the bipartite graph item-properties.

- $\{subject, starring, director, cinematography, producer, writer, editing, country, musicComposer, language, basedOn\}$
- $\{subject\}$
- $\{subject, starring, writer, editing, director, language, narrator\}$

The best configuration is then chosen to be used in the Deep CBRS Amar architecture.

Starting from this dataset, two knowlede graph has been used for generating the KG embeddings, that is a bipartite graph item-properties, and a tripartite graph user-item-properties.

As said in Section 2.3, the embedding vector of the user is computed as the centroid vector of the embedding of the items he/she like.

6 Results

6.1 Knowledge Graph Embedding Generation

KG embedding has been generating for each model, for each properties subset and for each vector dimensions, that is 256, 512, 768. The embeddings has been evalueted using a qualitative and quantitative analysis. The analysis has been carried out as follow: given three movies, the most four similar movies are retrieved using cosine similarity. For

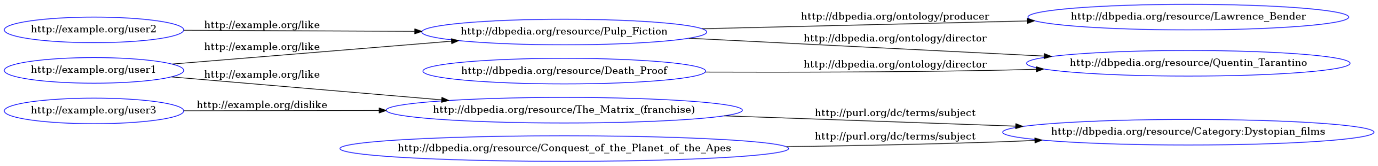

Figure 7: A snapshot from the tripartite graph user-item-properties.

each retrieved movie a score is given, which reflects overlapping between the properties of the movies. Given the movies m_1, m_2,

$$score(m_1, m_2) = \begin{cases} 0 & no\ similarity \\ \frac{1}{2} & weak\ similarity \\ 1 & strong\ similarity \end{cases}$$

The final score is computed as the average score.

On the basis of the following results, some decision has been taken:

- discard RESCAL because it crashs on higher dimension;
- pick the first subset properties configuration;
- use Transational model trained using SGD as optimizer, a learning rate α equals to 0.001, with a batch size of 50 on 1000 epochs;
- use Semantic model trained using Adam as optimizer, a learning rate α equals to 0.0001, with a batch size of 50 on 1000 epochs.

Those embeddings has been used into Deep CBRS Amar architecture.

6.1.1 Embedding Generation on item-properties Graph

The following results has been obtained using SGD as optimizer, a learning rate α equals to 0.001, with a batch size of 50 on 1000 epochs. Because of space complexity, RESCAL has been used only on dimension 256 e 512. It easy to note that TransE works very well, and it is the best model on each configuration. Some models such as DistMult work very poor, for this reason different parameters has been tested. Best results came using Adam as optimizer and setting the learning rate equals to 0.001. However this parameter settings worked well only on Semantic models, making no improvement on Transational ones.

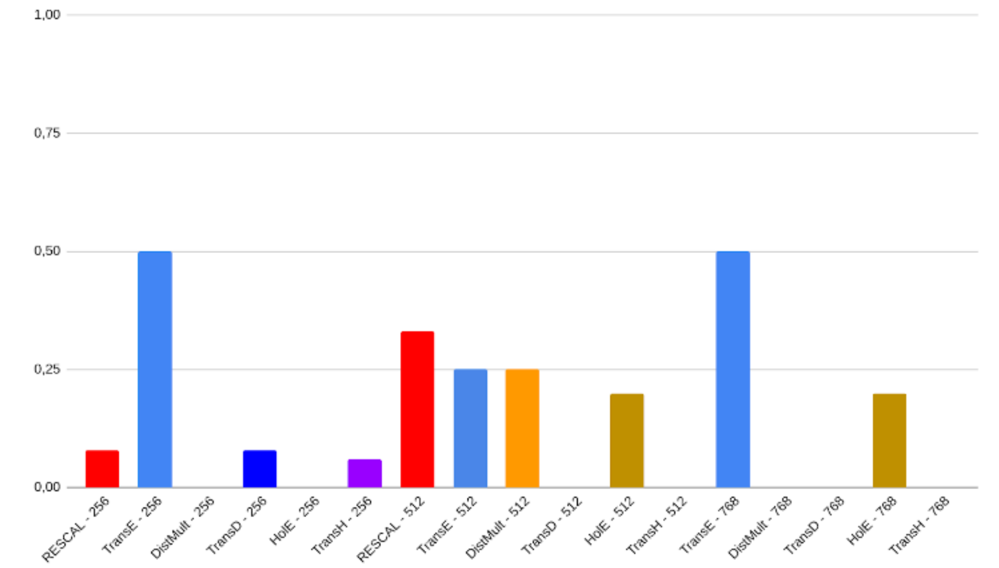

Figure 8: Results on item-properties graph using the properties $\{subject, starring, director, cinematography, producer, writer, editing, country, musicComposer, language, basedOn\}$

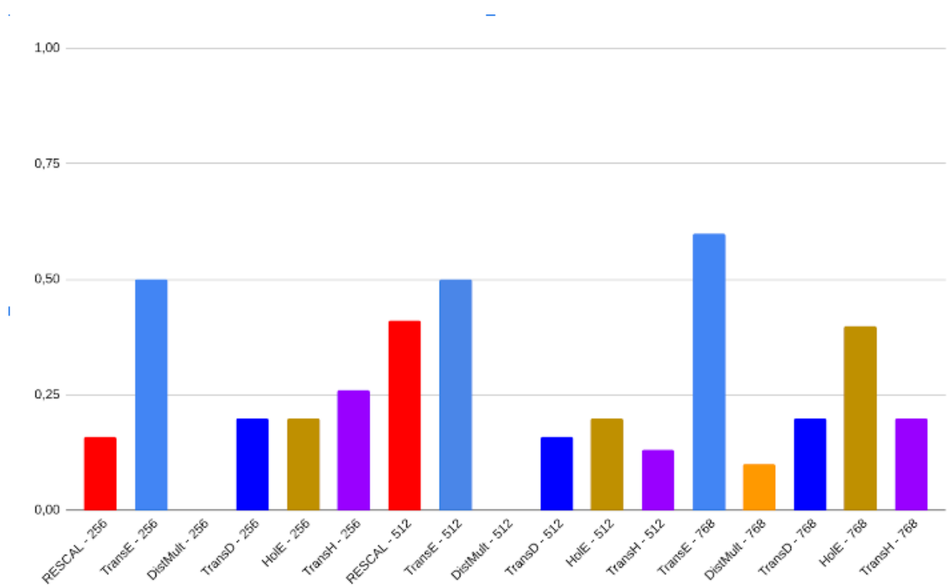

Figure 9: Results on item-properties graph using the properties $\{subject\}$

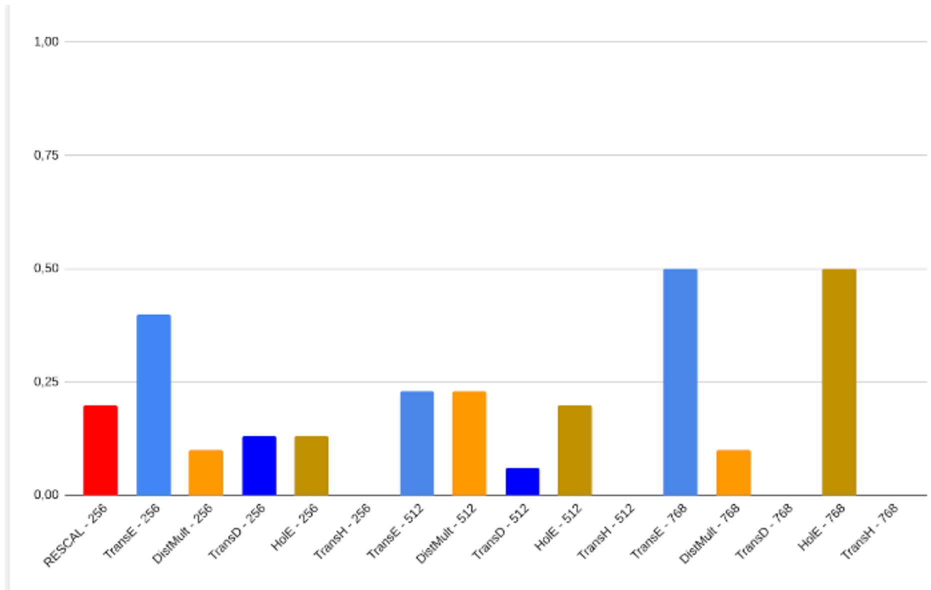

Figure 10: Results on item-properties graph using the properties $\{subject, starring, writer, editing, director, language, narrator\}$

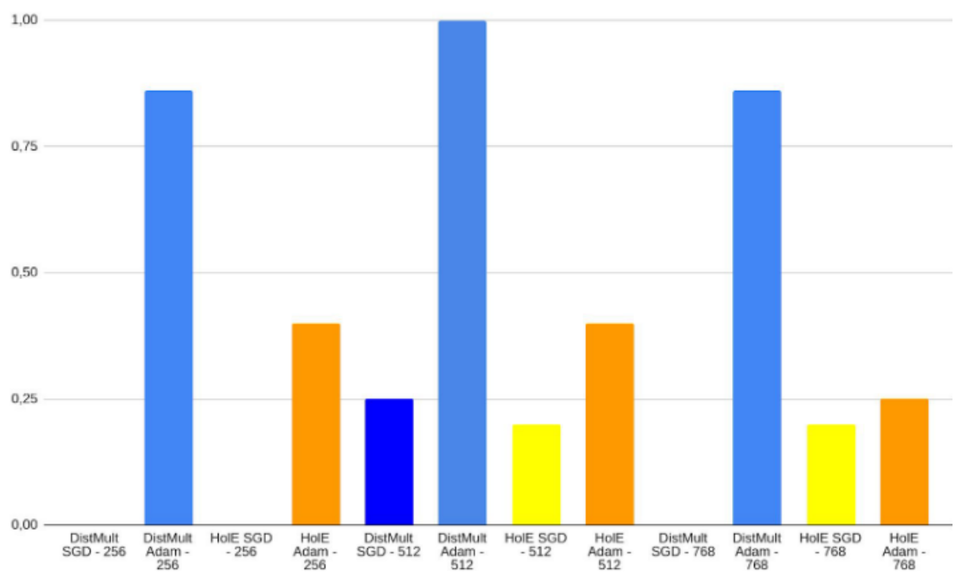

Figure 11: Results on Semantic models using Adams as optimzier and a learning rate of 0.0001.

6.1.2 Embedding Generation on user-item-properties Graph

As before, the following results has been obtained using SGD as optimizer, a learning rate α equals to 0.001, with a batch size of 50 on 1000 epochs. Because of space complexity, RESCAL has been used only on dimension 128. On the user-item-properties graph, TransE is not the best anymore. It work well on low dimension but get worse on higher dimensions. Same behaviour for TransH and TransD. Different paramters has been used and, as before, Semantic models get better with Adam and a learning rate equals to 0.0001.

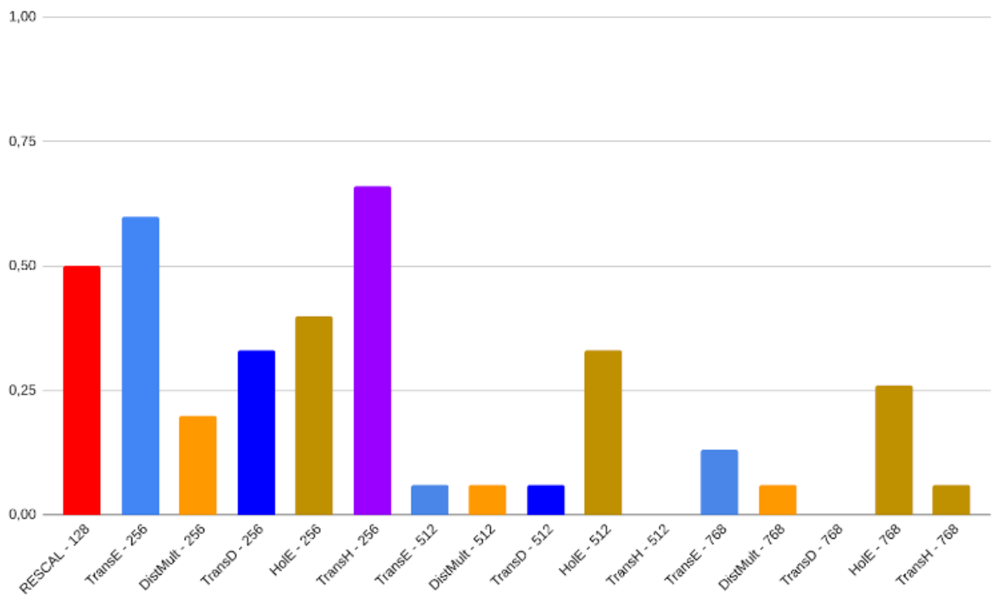

Figure 12: Results on user-item-properties graph using the properties $\{subject, starring, director, cinematography, producer, writer, editing, country, musicComposer, language, basedOn\}$

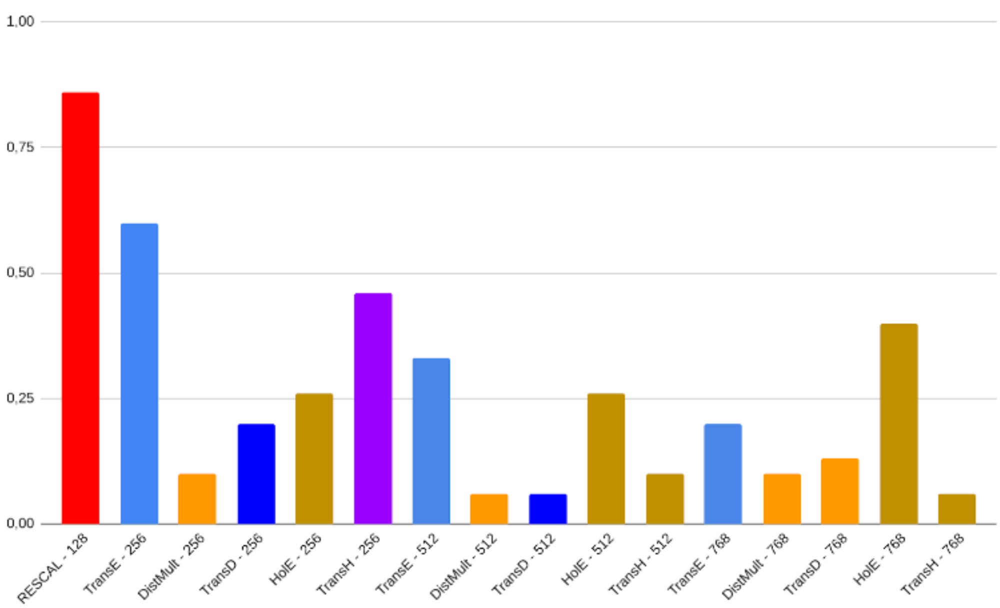

Figure 13: Results on user-item-properties graph using the properties $\{subject\}$

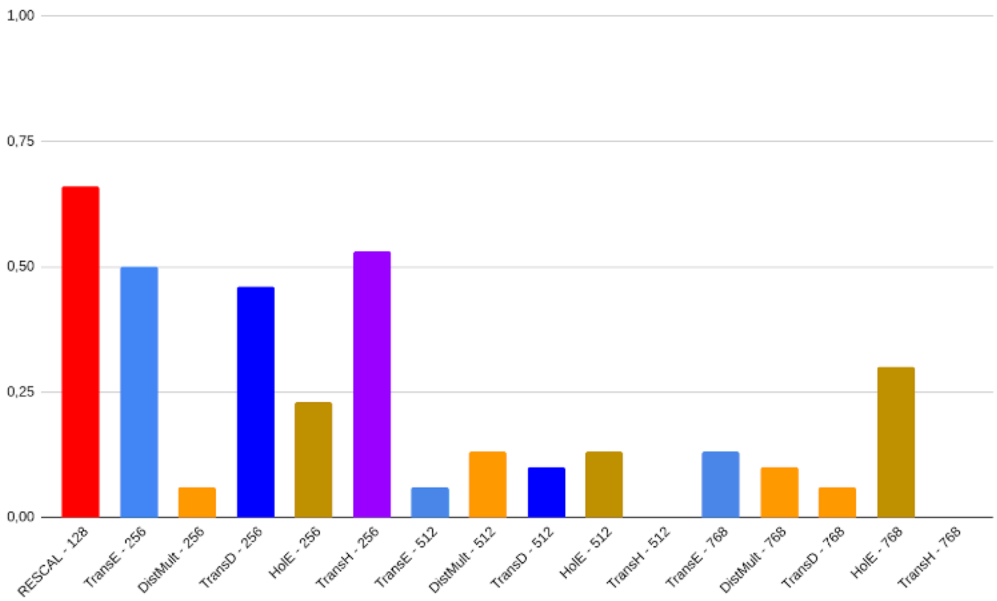

Figure 14: Results on user-item-properties graph using the properties $\{subject, starring, writer, editing, director, language, narrator\}$

6.2 Deep CBRS Amar

As stated in the introduction, one of the objective of this work is to use two different knowledge graphs for representing users and items. The previous work has used another type of graph in which nodes were *user* and *movie* and edges are *like* and *dislike*. Now, let's see how the KG embeddings generated on those two graphs work with Deep CBRS Amar, using as baseline the results obtained from the previous work. See figure 15 for the comparisons.

In the nex subsections all results are show. The best results on each variant are highlighted. In general, the two new graphs used in this work, improve the overall result respect to the previous work. This means that using additional information about items can be beneficial.

Figure 15: Comparison between KGs used on each Deep CBRS Amar architecture. Best performed models have been selected.

6.2.1 First Variant (BASIC)

MOVIELENS			TOP 5									TOP 10								
			256			512			768			256			512			768		
			Precision	*Recall*	*F1*	*Precision*	*Recall*	*F1*	*Precision*	*Recall*	*F1*	*Precision*	*Recall*	*F1*	*Precision*	*Recall*	*F1*	*Precision*	*Recall*	*F1*
BASIC	GRAPH Emb	TransE	0.7683	0.4475	0.5656	0.7714	0.4490	0.5676	0.7722	0.4492	0.5680	0.6573	0.6461	0.6517	0.6611	0.6485	0.6547	0.6622	0.6496	0.6559
		TransH	0.7701	0.4479	0.5664	0.7718	0.4496	0.5682	0.7722	0.4494	0.5682	0.6586	0.6465	0.6525	0.6613	0.6486	0.6549	0.6618	0.6490	0.6554
		TransD	0.7697	0.4481	0.5664	0.7710	0.4492	0.5677	0.7736	0.4501	0.5691	0.6582	0.6464	0.6523	0.6619	0.6491	0.6554	0.6617	0.6485	0.6550
		DistMult	0.7658	0.4460	0.5637	0.7725	0.4495	0.5683	0.7731	0.4494	0.5684	0.6557	0.6448	0.6502	0.6607	0.6482	0.6544	0.6619	0.6490	0.6554
		HolE	0.7102	0.4222	0.5296	0.7392	0.4336	0.5466	0.7526	0.4405	0.5557	0.6128	0.6210	0.6169	0.6361	0.6324	0.6343	0.6455	0.6384	0.6420
	BERT Emb	lastlayer							0.7680	0.4479	0.5658							0.6581	0.6469	0.6525
		lastlayer_no_stopw							0.7692	0.4487	0.5668							0.6568	0.6456	0.6512
	GRAPH Emb user + BERT Emb item	TransE u + lastlayer i							0.7710	0.4491	0.5675							0.6584	0.6473	0.6528
		TransE u + lastlayer_no_stopw i							0.7607	0.4453	0.5617							0.6522	0.6435	0.6478
		TransH u + lastlayer i							0.7686	0.4471	0.5654							0.6577	0.6466	0.6521
		TransH u + lastlayer_no_stopw i							0.7517	0.4417	0.5564							0.6463	0.6399	0.6430
		TransD u + lastlayer i							0.7401	0.4370	0.5495							0.6390	0.6354	0.6372
		TransD u + lastlayer_no_stopw i							0.7720	0.4498	0.5684							0.6604	0.6478	0.6541
		DistMult u + lastlayer i							0.7540	0.4424	0.5576							0.6505	0.6425	0.6465
		DistMult u + lastlayer_no_stopw i							0.7655	0.4468	0.5643							0.6569	0.6460	0.6514
		HolE u + lastlayer i							0.7583	0.4450	0.5608							0.6526	0.6438	0.6482
		HolE u + lastlayer_no_stopw i							0.7643	0.4465	0.5637							0.6563	0.6457	0.6510
	BERT Emb user + GRAPH Emb item	lastlayer u + TransE i							0.7711	0.4488	0.5673							0.6622	0.6493	0.6557
		lastlayer u + TransH i							0.7723	0.4492	0.5680							0.6609	0.6487	0.6547
		lastlayer u + TransD i							0.7735	0.4504	0.5693							0.6621	0.6493	0.6556
		lastlayer u + DistMult i							0.7720	0.4498	0.5684							0.6620	0.6491	0.6555
		lastlayer u + HolE i							0.7468	0.4380	0.5522							0.6412	0.6357	0.6384
		lastlayer_no_stopw u + TransE i							0.7730	0.4497	0.5686							0.6622	0.6490	0.6555
		lastlayer_no_stopw u + TransH i							0.7728	0.4495	0.5684							0.6621	0.6491	0.6555
		lastlayer_no_stopw u + TransD i							0.7740	0.4505	0.5695							0.6613	0.6486	0.6549
		lastlayer_no_stopw u + DistMult i							0.7723	0.4498	0.5685							0.6608	0.6482	0.6545
		lastlayer_no_stopw u + HolE i							0.7500	0.4387	0.5535							0.6436	0.6370	0.6402

Figure 16: Result of the Deep CBRS Amar first variant, from previous work, where the KG embedding are generated from the user-item graph.

MOVIELENS			TOP 5									TOP 10								
			256			512			768			256			512			768		
			Precision	*Recall*	*F1*	*Precision*	*Recall*	*F1*	*Precision*	*Recall*	*F1*	*Precision*	*Recall*	*F1*	*Precision*	*Recall*	*F1*	*Precision*	*Recall*	*F1*
BASIC	GRAPH Emb	TransE	0.7663	0.4476	0.5651	0.7732	0.4502	0.5690	0.7737	0.4501	0.5691	0.6574	0.6460	0.6517	0.6622	0.6489	0.6555	0.6624	0.6494	0.6559
		TransH	0.7700	0.4491	0.5673	0.7742	0.4508	0.5698	0.7741	0.4501	0.5692	0.6598	0.6475	0.6536	0.6622	0.6491	0.6556	0.6630	0.6493	0.6561
		TransD	0.7677	0.4485	0.5662	0.7737	0.4504	0.5694	0.7750	0.4506	0.5699	0.6593	0.6475	0.6534	0.6621	0.6491	0.6556	0.6623	0.6495	0.6558
		DistMult	0.7671	0.4470	0.5649	0.7723	0.4497	0.5685	0.7715	0.4501	0.5685	0.6569	0.6455	0.6512	0.6605	0.6481	0.6542	0.6606	0.6479	0.6542
		HolE	0.7396	0.4359	0.5485	0.7487	0.4379	0.5526	0.7617	0.4447	0.5616	0.6349	0.6328	0.6338	0.6414	0.6364	0.6389	0.6536	0.6433	0.6484
	BERT Emb	lastlayer							0.7709	0.4498	0.5682							0.6593	0.6472	0.6532
		lastlayer_no_stopw							0.7727	0.4508	0.5694							0.6605	0.6480	0.6542
	GRAPH Emb user + BERT Emb item	TransE u + lastlayer i							0.7623	0.4456	0.5624							0.6535	0.6441	0.6488
		TransE u + lastlayer_no_stopw i							0.7714	0.4494	0.5679							0.6605	0.6478	0.6541
		TransH u + lastlayer i							0.7697	0.4484	0.5667							0.6588	0.6472	0.6529
		TransH u + lastlayer_no_stopw i							0.7715	0.4498	0.5683							0.6606	0.6481	0.6543
		TransD u + lastlayer i							0.7636	0.4466	0.5636							0.6560	0.6455	0.6507
		TransD u + lastlayer_no_stopw i							0.7705	0.4485	0.5670							0.6599	0.6477	0.6537
		DistMult u + lastlayer i							0.7683	0.4471	0.5652							0.6595	0.6472	0.6533
		DistMult u + lastlayer_no_stopw i							0.7689	0.4481	0.5663							0.6602	0.6482	0.6542
		HolE u + lastlayer i							0.7699	0.4486	0.5669							0.6590	0.6475	0.6532
		HolE u + lastlayer_no_stopw i							0.7669	0.4470	0.5648							0.6570	0.6456	0.6512
	BERT Emb user + GRAPH Emb item	lastlayer u + TransE i							0.7736	0.4505	0.5694							0.6631	0.6499	0.6564
		lastlayer u + TransH i							0.7750	0.4507	0.5699							0.6621	0.6490	0.6555
		lastlayer u + TransD i							0.7748	0.4506	0.5702							0.6624	0.6493	0.6558
		lastlayer u + DistMult i							0.7743	0.4513	0.5702							0.6613	0.6487	0.6549
		lastlayer u + HolE i							0.7638	0.4454	0.5627							0.6553	0.6446	0.6499
		lastlayer_no_stopw u + TransE i							0.7754	0.4513	0.5705							0.6626	0.6495	0.6560
		lastlayer_no_stopw u + TransH i							0.7734	0.4507	0.5695							0.6630	0.6501	0.6565
		lastlayer_no_stopw u + TransD i							0.7736	0.4506	0.5695							0.6620	0.6485	0.6552
		lastlayer_no_stopw u + DistMult i							0.7711	0.4493	0.5678							0.6607	0.6486	0.6546
		lastlayer_no_stopw u + HolE i							0.7676	0.4477	0.5655							0.6567	0.6458	0.6512

Figure 17: Result of the Deep CBRS Amar first variant, where the KG embedding are generated from the item-properties graph.

MOVIELENS			TOP 5									TOP 10								
			256			512			768			256			512			768		
			Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
BASIC	GRAPH Emb	TransE	0.7742	0.4506	0.5696	0.7749	0.4510	0.5701	0.7753	0.4513	0.5705	0.6629	0.6495	0.6561	0.6635	0.6502	0.6568	0.6633	0.6501	0.6567
		TransH	0.7746	0.4508	0.5699	0.7756	0.4517	0.5709	0.7754	0.4515	0.5707	0.6637	0.6502	0.6569	0.6640	0.6503	0.6571	0.6635	0.6499	0.6566
		TransD	0.7747	0.4510	0.5701	0.7766	0.4518	0.5713	0.7759	0.4515	0.5708	0.6638	0.6503	0.6569	0.6637	0.6500	0.6568	0.6636	0.6501	0.6568
		DistMult	0.7649	0.4464	0.5638	0.7734	0.4500	0.5690	0.7736	0.4499	0.5689	0.6561	0.6458	0.6509	0.6612	0.6490	0.6551	0.6617	0.6489	0.6552
		HolE	0.7480	0.4381	0.5526	0.7548	0.4405	0.5563	0.7578	0.4428	0.5590	0.6427	0.6363	0.6395	0.6478	0.6392	0.6435	0.6484	0.6396	0.6439
	BERT Emb	lastlayer							0.7709	0.4498	0.5682							0.6593	0.6472	0.6532
		lastlayer_no_stopw							0.7727	0.4508	0.5694							0.6605	0.6480	0.6542
	GRAPH Emb user + BERT Emb item	TransE u + lastlayer i							0.7359	0.4347	0.5465							0.6325	0.6320	0.6322
		TransE u + lastlayer_no_stopw i							0.7726	0.4495	0.5684							0.6593	0.6475	0.6534
		TransH u + lastlayer i							0.7281	0.4325	0.5427							0.6286	0.6302	0.6294
		TransH u + lastlayer_no_stopw i							0.7728	0.4505	0.5692							0.6606	0.6481	0.6543
		TransD u + lastlayer i							0.7326	0.4341	0.5451							0.6312	0.6321	0.6317
		TransD u + lastlayer_no_stopw i							0.7703	0.4489	0.5672							0.6602	0.6483	0.6542
		DistMult u + lastlayer i							0.7341	0.4344	0.5458							0.6339	0.6331	0.6335
		DistMult u + lastlayer_no_stopw i							0.7680	0.4480	0.5659							0.6597	0.6476	0.6536
		HolE u + lastlayer i							0.7384	0.4364	0.5486							0.6352	0.6333	0.6342
		HolE u + lastlayer_no_stopw i							0.7661	0.4475	0.5650							0.6577	0.6460	0.6518
	BERT Emb user + GRAPH Emb item	lastlayer u + TransE i							0.7742	0.4505	0.5696							0.6631	0.6498	0.6564
		lastlayer u + TransH i							0.7754	0.4510	0.5703							0.6637	0.6504	0.6569
		lastlayer u + TransD i							0.7754	0.4513	0.5705							0.6633	0.6500	0.6566
		lastlayer u + DistMult i							0.7743	0.4499	0.5691							0.6624	0.6491	0.6557
		lastlayer u + HolE i							0.7592	0.4436	0.5600							0.6507	0.6413	0.6460
		lastlayer_no_stopw u + TransE i							0.7755	0.4510	0.5703							0.6638	0.6503	0.6570
		lastlayer_no_stopw u + TransH i							0.7753	0.4512	0.5704							0.6639	0.6504	0.6571
		lastlayer_no_stopw u + TransD i							0.7752	0.4511	0.5703							0.6633	0.6500	0.6565
		lastlayer_no_stopw u + DistMult i							0.7728	0.4502	0.5689							0.6609	0.6484	0.6546
		lastlayer_no_stopw u + HolE i							0.7572	0.4423	0.5584							0.6483	0.6397	0.6440

Figure 18: Result of the Deep CBRS Amar first variant, where the KG embedding are generated from the user-item-properties graph.

6.2.2 Second Variant (MIXED)

MOVIELENS			TOP 5									TOP 10								
			256			512			768			256			512			768		
			Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
MIXED	CONF 1: (GRAPH Emb item + BERT Emb item) + (GRAPH Emb user + BERT Emb user)	(TransE i + lastlayer i) + (TransE u + lastlayer u)							0.7741	0.4505	0.5696							0.6623	0.6434	0.6558
		(TransE i + lastlayer_no_stopw i) + (TransE u + lastlayer_no_stopw u)							0.7744	0.4512	0.5702							0.6624	0.6492	0.6557
		(TransH i + lastlayer i) + (TransH u + lastlayer u)							0.7749	0.4511	0.5702							0.6621	0.6491	0.6555
		(TransH i + lastlayer_no_stopw i) + (TransH u + lastlayer_no_stopw u)							0.7747	0.4511	0.5702							0.6617	0.6485	0.6550
		(TransD i + lastlayer i) + (TransD u + lastlayer u)							0.7744	0.4510	0.5700							0.6625	0.6494	0.6559
		(TransD i + lastlayer_no_stopw i) + (TransD u + lastlayer_no_stopw u)							0.7752	0.4511	0.5704							0.6628	0.6495	0.6560
		(DistMult i + lastlayer i) + (DistMult u + lastlayer u)							0.7748	0.4509	0.5701							0.6627	0.6497	0.6561
		(DistMult i + lastlayer_no_stopw i) + (DistMult u + lastlayer_no_stopw u)							0.7747	0.4511	0.5702							0.6627	0.6495	0.6561
		(HolE i + lastlayer i) + (HolE u + lastlayer u)							0.7722	0.4499	0.5685							0.6602	0.6475	0.6538
		(HolE i + lastlayer_no_stopw i) + (HolE u + lastlayer no stopw u)							0.7726	0.4501	0.5689							0.6606	0.6485	0.6545
	CONF 2: (GRAPH Emb item + GRAPH Emb user) + (BERT Emb item + BERT Emb user)	(TransE i + TransE u) + (lastlayer i + lastlayer u)	0.7817	0.4534	0.5739	0.7850	0.4551	0.5762	0.7825	0.4536	0.5743	0.6667	0.6517	0.6591	0.6714	0.6539	0.6625	0.6684	0.6522	0.6602
		(TransE i + TransE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7840	0.4550	0.5759	0.7806	0.4524	0.5728				0.6698	0.6531	0.6614	0.6676	0.6523	0.6598
		(TransH i + TransH u) + (lastlayer i + lastlayer u)				0.7830	0.4538	0.5746	0.7793	0.4528	0.5728				0.6686	0.6522	0.6603	0.6667	0.6512	0.6589
		(TransH i + TransH u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7690	0.4473	0.5656	0.7858	0.4551	0.5764				0.6596	0.6471	0.6533	0.6705	0.6537	0.6620
		(TransD i + TransD u) + (lastlayer i + lastlayer u)	0.7749	0.4512	0.5703	0.7864	0.4555	0.5769	0.7791	0.4524	0.5724	0.6645	0.6502	0.6573	0.6717	0.6535	0.6625	0.6677	0.6521	0.6598
		(TransD i + TransD u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7721	0.4494	0.5681	0.7838	0.4540	0.5750				0.6621	0.6489	0.6554	0.6700	0.6531	0.6614
		(DistMult i + DistMult u) + (lastlayer i + lastlayer u)				0.7721	0.4501	0.5687	0.7798	0.4534	0.5734				0.6612	0.6488	0.6550	0.6662	0.6507	0.6583
		(DistMult i + DistMult u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7721	0.4497	0.5684	0.7804	0.4530	0.5732				0.6617	0.6489	0.6553	0.6666	0.6513	0.6589
		(HolE i + HolE u) + (lastlayer i + lastlayer u)				0.7289	0.4300	0.5409	0.7679	0.4476	0.5656				0.6280	0.6286	0.6283	0.6587	0.6469	0.6527
		(HolE i + HolE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7833	0.4547	0.5754	0.7732	0.4509	0.5696				0.6686	0.6521	0.6603	0.6631	0.6491	0.6560

Figure 19: Result of the Deep CBRS Amar second variant, from previous work, where the KG embedding are generated from the user-item graph.

MOVIELENS			TOP 5									TOP 10								
			256			512			768			256			512			768		
			Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
MIXED	CONF 1: (GRAPH Emb item + BERT Emb item) + (GRAPH Emb user + BERT Emb user)	(TransE i + lastlayer i) + (TransE u + lastlayer u)							0.7761	0.4509	0.5704							0.6625	0.6493	0.6558
		(TransE i + lastlayer_no_stopw i) + (TransE u + lastlayer_no_stopw u)							0.7740	0.4505	0.5695							0.6629	0.6494	0.6561
		(TransH i + lastlayer i) + (TransH u + lastlayer u)							0.7744	0.4514	0.5703							0.6634	0.6501	0.6567
		(TransH i + lastlayer_no_stopw i) + (TransH u + lastlayer_no_stopw u)							0.7745	0.4505	0.5697							0.6632	0.6496	0.6564
		(TransD i + lastlayer i) + (TransD u + lastlayer u)							0.7756	0.4512	0.5705							0.6628	0.6493	0.6560
		(TransD i + lastlayer_no_stopw i) + (TransD u + lastlayer_no_stopw u)							0.7746	0.4510	0.5701							0.6624	0.6494	0.6558
		(DistMult i + lastlayer i) + (DistMult u + lastlayer u)							0.7753	0.4517	0.5708							0.6632	0.6498	0.6564
		(DistMult i + lastlayer_no_stopw i) + (DistMult u + lastlayer_no_stopw u)							0.7749	0.4516	0.5706							0.6624	0.6493	0.6558
		(HolE i + lastlayer i) + (HolE u + lastlayer u)							0.7706	0.4491	0.5674							0.6603	0.6476	0.6539
		(HolE i + lastlayer_no_stopw i) + (HolE u + lastlayer_no_stopw u)							0.7745	0.4505	0.5697							0.6632	0.6496	0.6564
	CONF 2: (GRAPH Emb item + GRAPH Emb user) + (BERT Emb item + BERT Emb user)	(TransE i + TransE u) + (lastlayer i + lastlayer u)	0.7612	0.4449	0.5616	0.7744	0.4512	0.5702	0.7837	0.4546	0.5754	0.6536	0.6441	0.6488	0.6627	0.6494	0.6560	0.6714	0.6537	0.6624
		(TransE i + TransE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7847	0.4541	0.5753	0.7853	0.4543	0.5756				0.6717	0.6538	0.6626	0.6714	0.6541	0.6626
		(TransH i + TransH u) + (lastlayer i + lastlayer u)				0.7820	0.4547	0.5750	0.7822	0.4537	0.5743				0.6691	0.6529	0.6609	0.6701	0.6530	0.6614
		(TransH i + TransH u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7876	0.4561	0.5777	0.7874	0.4555	0.5772				0.6738	0.6551	0.6643	0.6721	0.6544	0.6632
		(TransD i + TransD u) + (lastlayer i + lastlayer u)	0.7783	0.4522	0.5721	0.7734	0.4504	0.5693	0.7860	0.4546	0.5760	0.6656	0.6509	0.6582	0.6622	0.6494	0.6558	0.6729	0.6547	0.6637
		(TransD i + TransD u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7780	0.4509	0.5709	0.7839	0.4551	0.5759				0.6673	0.6510	0.6591	0.6688	0.6527	0.6606
		(DistMult i + DistMult u) + (lastlayer i + lastlayer u)				0.7792	0.4530	0.5729	0.7828	0.4535	0.5743				0.6648	0.6491	0.6569	0.6704	0.6532	0.6617
		(DistMult i + DistMult u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7850	0.4544	0.5756	0.7828	0.4535	0.5743				0.6703	0.6528	0.6614	0.6685	0.6528	0.6605
		(HolE i + HolE u) + (lastlayer i + lastlayer u)				0.7405	0.4352	0.5482	0.7672	0.4476	0.5654				0.6345	0.6324	0.6334	0.6586	0.6465	0.6525
		(HolE i + HolE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7739	0.4503	0.5693	0.7770	0.4513	0.5710				0.6607	0.6480	0.6543	0.6655	0.6503	0.6578

Figure 20: Result of the Deep CBRS Amar second variant, where the KG embedding are generated from the item-properties graph.

MOVIELENS			TOP 5									TOP 10								
			256			512			768			256			512			768		
			Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
MIXED	CONF 1: (GRAPH Emb item + BERT Emb item) + (GRAPH Emb user + BERT Emb user)	(TransE i + lastlayer i) + (TransE u + lastlayer u)							0.7756	0.4511	0.5704							0.6629	0.6496	0.6562
		(TransE i + lastlayer_no_stopw i) + (TransE u + lastlayer_no_stopw u)							0.7756	0.4514	0.5707							0.6632	0.6500	0.6565
		(TransH i + lastlayer i) + (TransH u + lastlayer u)							0.7757	0.4514	0.5707							0.6634	0.6499	0.6566
		(TransH i + lastlayer_no_stopw i) + (TransH u + lastlayer_no_stopw u)							0.7758	0.4515	0.5708							0.6633	0.6500	0.6566
		(TransD i + lastlayer i) + (TransD u + lastlayer u)							0.7759	0.4517	0.5710							0.6629	0.6498	0.6563
		(TransD i + lastlayer_no_stopw i) + (TransD u + lastlayer_no_stopw u)							0.7768	0.4521	0.5716							0.6635	0.6502	0.6568
		(DistMult i + lastlayer i) + (DistMult u + lastlayer u)							0.7740	0.4505	0.5695							0.6619	0.6492	0.6555
		(DistMult i + lastlayer_no_stopw i) + (DistMult u + lastlayer_no_stopw u)							0.7739	0.4508	0.5697							0.6622	0.6496	0.6558
		(HolE i + lastlayer i) + (HolE u + lastlayer u)							0.7715	0.4491	0.5677							0.6616	0.6486	0.6550
		(HolE i + lastlayer_no_stopw i) + (HolE u + lastlayer_no_stopw u)							0.7652	0.4474	0.5646							0.6559	0.6456	0.6507
	CONF 2: (GRAPH Emb item + GRAPH Emb user) + (BERT Emb item + BERT Emb user)	(TransE i + TransE u) + (lastlayer i + lastlayer u)	0.7922	0.4579	0.5803	0.7948	0.4588	0.5818	0.7750	0.4509	0.5701	0.6755	0.6563	0.6658	0.6775	0.6576	0.6674	0.6636	0.6500	0.6568
		(TransE i + TransE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7919	0.4579	0.5803	0.7889	0.4572	0.5789				0.6744	0.6556	0.6648	0.6737	0.6549	0.6642
		(TransH i + TransH u) + (lastlayer i + lastlayer u)				0.7911	0.4570	0.5793	0.7906	0.4569	0.5791				0.6761	0.6564	0.6661	0.6729	0.6544	0.6635
		(TransH i + TransH u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7929	0.4574	0.5802	0.7907	0.4573	0.5795				0.6768	0.6571	0.6668	0.6738	0.6552	0.6643
		(TransD i + TransD u) + (lastlayer i + lastlayer u)	0.7893	0.4567	0.5786	0.7914	0.4570	0.5794	0.7919	0.4570	0.5796	0.6728	0.6551	0.6638	0.6752	0.6559	0.6654	0.6759	0.6563	0.6659
		(TransD i + TransD u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7917	0.4572	0.5796	0.7910	0.4573	0.5795				0.6752	0.6562	0.6656	0.6763	0.6569	0.6664
		(DistMult i + DistMult u) + (lastlayer i + lastlayer u)				0.7727	0.4498	0.5686	0.7808	0.4534	0.5737				0.6615	0.6488	0.6551	0.6682	0.6523	0.6601
		(DistMult i + DistMult u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7802	0.4531	0.5733	0.7825	0.4538	0.5744				0.6685	0.6523	0.6603	0.6698	0.6529	0.6612
		(HolE i + HolE u) + (lastlayer i + lastlayer u)				0.7496	0.4389	0.5536	0.7726	0.4497	0.5685				0.6454	0.6377	0.6415	0.6617	0.6485	0.6551
		(HolE i + HolE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7821	0.4540	0.5745	0.7718	0.4496	0.5682				0.6688	0.6524	0.6605	0.6615	0.6486	0.6550

Figure 21: Result of the Deep CBRS Amar second variant, where the KG embedding are generated from the user-item-properties graph.

6.2.3 Third Variant (EXTENDED)

MOVIELENS			TOP 5									TOP 10								
			256			512			768			256			512			768		
			Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
EXTENDED	CONF 2 with 1 Attention layer: (GRAPH Emb item + GRAPH Emb user) + (BERT Emb item + BERT Emb user)	(TransE i + TransE u) + (lastlayer i + lastlayer u)				0.7648	0.4456	0.5631	0.7971	0.4596	0.5830				0.6551	0.6437	0.6493	0.6810	0.6590	0.6698
		(TransE i + TransE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7694	0.4494	0.5674	0.7973	0.4600	0.5834				0.6562	0.6447	0.6504	0.6813	0.6587	0.6698
		(TransH i + TransH u) + (lastlayer i + lastlayer u)				0.7648	0.4456	0.5631	0.7958	0.4587	0.5819				0.6551	0.6437	0.6493	0.6791	0.6580	0.6684
		(TransH i + TransH u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7793	0.4529	0.5729	0.7972	0.4595	0.5830				0.6638	0.6487	0.6562	0.6805	0.6585	0.6693
		(TransD i + TransD u) + (lastlayer i + lastlayer u)				0.7713	0.4492	0.5678	0.7966	0.4591	0.5825				0.6602	0.6468	0.6534	0.6802	0.6585	0.6692
		(TransD i + TransD u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7741	0.4504	0.5694	0.7942	0.4582	0.5811				0.6608	0.6472	0.6540	0.6795	0.6582	0.6687
		(DistMult i + DistMult u) + (lastlayer i + lastlayer u)				0.7654	0.4463	0.5638	0.7921	0.4573	0.5798				0.6567	0.6452	0.6509	0.6767	0.6564	0.6664
		(DistMult i + DistMult u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7630	0.4455	0.5625	0.7938	0.4580	0.5808				0.6532	0.6437	0.6484	0.6772	0.6566	0.6667
		(HolE i + HolE u) + (lastlayer i + lastlayer u)				0.7870	0.4539	0.5757	0.7851	0.4532	0.5747				0.6741	0.6541	0.6640	0.6728	0.6542	0.6634
		(HolE i + HolE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7895	0.4538	0.5763	0.7877	0.4555	0.5772				0.6741	0.6544	0.6641	0.6725	0.6536	0.6629
	CONF 2 with 3 Attention layers: (GRAPH Emb item + GRAPH Emb user) + (BERT Emb item + BERT Emb user)	(TransE i + TransE u) + (lastlayer i + lastlayer u)							0.7284	0.4336	0.5436							0.6322	0.6338	0.6330
		(TransE i + TransE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7934	0.4596	0.5821							0.6764	0.6567	0.6664
		(TransH i + TransH u) + (lastlayer i + lastlayer u)							0.6224	0.3865	0.4769							0.5591	0.5934	0.5757
		(TransH i + TransH u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7878	0.4556	0.5773							0.6715	0.6539	0.6626
		(TransD i + TransD u) + (lastlayer i + lastlayer u)							0.7712	0.4491	0.5676							0.6623	0.6490	0.6555
		(TransD i + TransD u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7897	0.4571	0.5791							0.6728	0.6546	0.6636
		(DistMult i + DistMult u) + (lastlayer i + lastlayer u)							0.6651	0.4055	0.5038							0.5835	0.6063	0.5947
		(DistMult i + DistMult u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7756	0.4511	0.5704							0.6630	0.6499	0.6564
		(HolE i + HolE u) + (lastlayer i + lastlayer u)							0.7645	0.4479	0.5648							0.6570	0.6466	0.6518
		(HolE i + HolE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7718	0.4493	0.5680							0.6613	0.6487	0.6549
	CONF 2 with 1 Attention layer and dropout: (GRAPH Emb item + GRAPH Emb user) + (BERT Emb item + BERT Emb user)	(TransE i + TransE u) + (lastlayer i + lastlayer u)							0.7974	0.4596	0.5831							0.6808	0.6584	0.6694
		(TransE i + TransE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7907	0.4564	0.5787							0.6774	0.6558	0.6664

Figure 22: Result of the Deep CBRS Amar third variant, from previous work, where the KG embedding are generated from the user-item graph.

MOVIELENS			TOP 5									TOP 10								
			256			512			768			256			512			768		
			Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
EXTENDED	CONF 2 with 1 Attention layer: (GRAPH Emb item + GRAPH Emb user) + (BERT Emb item + BERT Emb user)	(TransE i + TransE u) + (lastlayer i + lastlayer u)				0.7848	0.4530	0.5744	0.7921	0.4568	0.5795				0.6694	0.6507	0.6599	0.6777	0.6575	0.6675
		(TransE i + TransE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7853	0.4526	0.5742	0.7947	0.4583	0.5814				0.6726	0.6529	0.6626	0.6780	0.6564	0.6670
		(TransH i + TransH u) + (lastlayer i + lastlayer u)				0.7715	0.4451	0.5645	0.7956	0.4586	0.5818				0.6618	0.6460	0.6538	0.6790	0.6578	0.6683
		(TransH i + TransH u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7799	0.4495	0.5703	0.7925	0.4575	0.5801				0.6667	0.6488	0.6576	0.6770	0.6564	0.6665
		(TransD i + TransD u) + (lastlayer i + lastlayer u)				0.7720	0.4469	0.5661	0.7943	0.4585	0.5814				0.6614	0.6456	0.6534	0.6768	0.6569	0.6667
		(TransD i + TransD u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7760	0.4472	0.5674	0.7921	0.4570	0.5796				0.6652	0.6478	0.6564	0.6768	0.6565	0.6665
		(DistMult i + DistMult u) + (lastlayer i + lastlayer u)				0.7790	0.4497	0.5702	0.7950	0.4576	0.5809				0.6686	0.6498	0.6590	0.6779	0.6570	0.6673
		(DistMult i + DistMult u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7828	0.4505	0.5719	0.7940	0.4572	0.5803				0.6694	0.6499	0.6595	0.6769	0.6558	0.6662
		(HolE i + HolE u) + (lastlayer i + lastlayer u)				0.7872	0.4535	0.5755	0.7851	0.4533	0.5747				0.6728	0.6536	0.6630	0.6714	0.6526	0.6618
		(HolE i + HolE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7857	0.4522	0.5741	0.7925	0.4575	0.5801				0.6722	0.6528	0.6623	0.6770	0.6564	0.6665
	CONF 2 with 3 Attention layers: (GRAPH Emb item + GRAPH Emb user) + (BERT Emb item + BERT Emb user)	(TransE i + TransE u) + (lastlayer i + lastlayer u)							0.7785	0.4525	0.5723							0.6674	0.6520	0.6596
		(TransE i + TransE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7880	0.4557	0.5775							0.6744	0.6559	0.6650
		(TransH i + TransH u) + (lastlayer i + lastlayer u)							0.7832	0.4546	0.5753							0.6705	0.6531	0.6617
		(TransH i + TransH u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7908	0.4567	0.5790							0.6705	0.6531	0.6617
		(TransD i + TransD u) + (lastlayer i + lastlayer u)							0.7483	0.4402	0.5543							0.6451	0.6384	0.6417
		(TransD i + TransD u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7795	0.4528	0.5728							0.6648	0.6502	0.6574
		(DistMult i + DistMult u) + (lastlayer i + lastlayer u)							0.7915	0.4578	0.5801							0.6735	0.6546	0.6639
		(DistMult i + DistMult u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7427	0.4404	0.5529							0.6450	0.6411	0.6431
		(HolE i + HolE u) + (lastlayer i + lastlayer u)							0.7763	0.4518	0.5712							0.6643	0.6499	0.6570
		(HolE i + HolE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7789	0.4517	0.5718							0.6666	0.6512	0.6588
	CONF 2 with 1 Attention layer and dropout: (GRAPH Emb item + GRAPH Emb user) + (BERT Emb item + BERT Emb user)	(TransE i + TransE u) + (lastlayer i + lastlayer u)							0.7940	0.4577	0.5807							0.6779	0.6564	0.6670
		(TransE i + TransE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7934	0.4571	0.5800							0.6772	0.6562	0.6665

Figure 23: Result of the Deep CBRS Amar third variant, where the KG embedding are generated from the item-properties graph.

MOVIELENS			TOP 5									TOP 10								
			256			512			768			256			512			768		
			Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
EXTENDED	CONF 2 with 1 Attention layer: (GRAPH Emb item + GRAPH Emb user) + (BERT Emb item + BERT Emb user)	(TransE i + TransE u) + (lastlayer i + lastlayer u)				0.7963	0.4584	0.5818	0.7989	0.4602	0.5840				0.6776	0.6555	0.6664	0.6808	0.6582	0.6693
		(TransE i + TransE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7948	0.4570	0.5804	0.7979	0.4593	0.5830				0.6761	0.6546	0.6651	0.6809	0.6579	0.6692
		(TransH i + TransH u) + (lastlayer i + lastlayer u)				0.7975	0.4586	0.5824	0.8004	0.4609	0.5849				0.6786	0.6563	0.6673	0.6810	0.6585	0.6696
		(TransH i + TransH u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7938	0.4569	0.5799	0.8018	0.4613	0.5856				0.6766	0.6554	0.6658	0.6809	0.6584	0.6695
		(TransD i + TransD u) + (lastlayer i + lastlayer u)				0.7929	0.4571	0.5799	0.7973	0.4579	0.5817				0.6750	0.6546	0.6646	0.6814	0.6580	0.6695
		(TransD i + TransD u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7940	0.4562	0.5795	0.8013	0.4611	0.5853				0.6784	0.6561	0.6670	0.6821	0.6588	0.6702
		(DistMult i + DistMult u) + (lastlayer i + lastlayer u)				0.7435	0.4336	0.5478	0.7922	0.4575	0.5800				0.6452	0.6356	0.6404	0.6764	0.6570	0.6666
		(DistMult i + DistMult u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7467	0.4339	0.5488	0.7925	0.4566	0.5794				0.6473	0.6371	0.6421	0.6768	0.6566	0.6665
		(HolE i + HolE u) + (lastlayer i + lastlayer u)				0.7908	0.4569	0.5791	0.7869	0.4544	0.5761				0.6753	0.6559	0.6655	0.6729	0.6537	0.6632
		(HolE i + HolE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7903	0.4571	0.5792	0.7904	0.4568	0.5789				0.6728	0.6541	0.6634	0.6751	0.6556	0.6652
	CONF 2 with 3 Attention layers: (GRAPH Emb item + GRAPH Emb user) + (BERT Emb item + BERT Emb user)	(TransE i + TransE u) + (lastlayer i + lastlayer u)							0.7733	0.4503	0.5692							0.6630	0.6494	0.6561
		(TransE i + TransE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7943	0.4586	0.5815							0.6767	0.6572	0.6668
		(TransH i + TransH u) + (lastlayer i + lastlayer u)							0.7939	0.4582	0.5810							0.6764	0.6572	0.6667
		(TransH i + TransH u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7922	0.4581	0.5805							0.6761	0.6567	0.6662
		(TransD i + TransD u) + (lastlayer i + lastlayer u)							0.7926	0.4584	0.5809							0.6759	0.6570	0.6664
		(TransD i + TransD u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7293	0.4363	0.5460							0.6759	0.6570	0.6664
		(DistMult i + DistMult u) + (lastlayer i + lastlayer u)							0.7816	0.4536	0.5741							0.6685	0.6519	0.6601
		(DistMult i + DistMult u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7940	0.4574	0.5804							0.6777	0.6574	0.6674
		(HolE i + HolE u) + (lastlayer i + lastlayer u)							0.7779	0.4525	0.5722							0.6651	0.6501	0.6575
		(HolE i + HolE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7424	0.4377	0.5507							0.6425	0.6373	0.6399
	CONF 2 with 1 Attention layer and dropout: (GRAPH Emb item + GRAPH Emb user) + (BERT Emb item + BERT Emb user)	(TransE i + TransE u) + (lastlayer i + lastlayer u)							0.7964	0.4584	0.5819							0.6809	0.6581	0.6693
		(TransE i + TransE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7952	0.4578	0.5810							0.6795	0.6573	0.6682

Figure 24: Result of the Deep CBRS Amar third variant, where the KG embedding are generated from the user-item-properties graph.

6.3 Deep CBRS Amar Rivisited

The last objective of this work is to improve Deep CBRS Amar architecture. The effort has been employed on the first two variants, the BASIC architecture and the MIXED architecture. An alternative architecture is given also for the EXTENDED architecture. The problem of the old architecture is that it is too shallow. Feature vectors are of dimension up to 768 and the denser layer consists of just 8 hidden units. In this way it is very difficult for the neural network to learn useful features. So, the idea is to make a deeper architecture: more dense layers and more denser layer. From the following results, if comparing the old architecture with the new, on the same knowledge graph, the BASIC architecture gain from 0.5 to 2 percentage point. Same for the MIXED architecture, even though the improvement is less marked. See figures 25,26 and 27 for direct comparison between the old and the proposed architectures.

The EXTENDED architecture proposed outperforms only the second configuration of the old EXTENDED variant on user-item and user-item-properties graph. Instead it outperforms the old architecture using the item-properties graph. Anyway, the idea of putting the attention module before the concatenation could be a way to improve the overall architecture and could be continued in future works. This because, an attention module for a single embedding may have more sense of an attention module for a concatenations of embedding, considering that the intuition behind the attention strategy is to create dynamic weights which compute the importance of a given information sequence. See the comparison on Fig. 28.

Figure 25: Comparison between the old and the new architectures using user-item graph for KG embeddings. Best performed models have been selected.

Figure 26: Comparison between the old and the new architectures using item-properties graph for KG embeddings. Best performed models have been selected.

Figure 27: Comparison between the old and the new architectures using user-item-properties graph for KG embeddings.
Best performed models have been selected.

Figure 28: Comparison between the old and the new architectures using user-item-properties graph for KG embeddings.
Best performed models have been selected.

6.3.1 First Variant (BASIC)

MOVIELENS			TOP 5									TOP 10								
			256			512			768			256			512			768		
			Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
BASIC	GRAPH Emb	TransE	0.7879	0.4543	0.5763	0.7903	0.4554	0.5778	0.7886	0.4551	0.5771	0.6750	0.6552	0.6650	0.6760	0.6555	0.6656	0.6752	0.6551	0.6650
		TransH	0.7875	0.4548	0.5766	0.7863	0.4556	0.5769	0.7868	0.4554	0.5769	0.6746	0.6552	0.6647	0.6730	0.6541	0.6634	0.6730	0.6550	0.6639
		TransD	0.7891	0.4558	0.5779	0.7877	0.4547	0.5765	0.7876	0.4555	0.5772	0.6755	0.6557	0.6655	0.6737	0.6544	0.6639	0.6755	0.6561	0.6656
		DistMult	0.7863	0.4556	0.5769	0.7845	0.4543	0.5754	0.7807	0.4534	0.5737	0.6740	0.6551	0.6644	0.6716	0.6533	0.6623	0.6694	0.6525	0.6609
		HolE	0.7624	0.4456	0.5624	0.7698	0.4485	0.5668	0.7692	0.4484	0.5665	0.6545	0.6448	0.6496	0.6598	0.6472	0.6534	0.6595	0.6474	0.6534
	BERT Emb	lastlayer							0.7924	0.4573	0.5799							0.6759	0.6559	0.6658
		lastlayer_no_stopw							0.7901	0.4559	0.5782							0.6746	0.6547	0.6645
	GRAPH Emb user + BERT Emb item	TransE u + lastlayer i							0.7928	0.4573	0.5800							0.6778	0.6573	0.6674
		TransE u + lastlayer_no_stopw i							0.7917	0.4579	0.5802							0.6746	0.6554	0.6649
		TransH u + lastlayer i							0.7906	0.4572	0.5794							0.6750	0.6553	0.6650
		TransH u + lastlayer_no_stopw i							0.7963	0.4587	0.5821							0.6797	0.6579	0.6686
		TransD u + lastlayer i							0.7950	0.4591	0.5821							0.6779	0.6568	0.6672
		TransD u + lastlayer_no_stopw i							0.7911	0.4565	0.5789							0.6774	0.6569	0.6670
		DistMult u + lastlayer i							0.7849	0.4552	0.5762							0.6718	0.6536	0.6626
		DistMult u + lastlayer_no_stopw i							0.7857	0.4547	0.5760							0.6715	0.6536	0.6624
		HolE u + lastlayer i							0.7789	0.4522	0.5722							0.6656	0.6504	0.6579
		HolE u + lastlayer_no_stopw i							0.7933	0.4571	0.5800							0.6773	0.6570	0.6670
	BERT Emb user + GRAPH Emb item	lastlayer u + TransE i							0.7853	0.4544	0.5757							0.6730	0.6548	0.6638
		lastlayer u + TransH i							0.7897	0.4566	0.5787							0.6738	0.6549	0.6642
		lastlayer u + TransD i							0.7889	0.4562	0.5781							0.6748	0.6552	0.6649
		lastlayer u + DistMult i							0.7881	0.4553	0.5772							0.6731	0.6544	0.6636
		lastlayer u + HolE i							0.7716	0.4491	0.5677							0.6610	0.6489	0.6549
		lastlayer_no_stopw u + TransE i							0.7909	0.4573	0.5796							0.6757	0.6561	0.6657
		lastlayer_no_stopw u + TransH i							0.7907	0.4565	0.5788							0.6748	0.6553	0.6649
		lastlayer_no_stopw u + TransD i							0.7892	0.4556	0.5777							0.6745	0.6548	0.6645
		lastlayer_no_stopw u + DistMult i							0.7912	0.4562	0.5788							0.6763	0.6562	0.6661
		lastlayer_no_stopw u + HolE i							0.7719	0.4497	0.5683							0.6601	0.6477	0.6539

Figure 29: Result of the Deep CBRS Amar Revisited first variant, where the KG embedding are generated from the user-item graph.

MOVIELENS			TOP 5									TOP 10								
			256			512			768			256			512			768		
			Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
BASIC	GRAPH Emb	TransE	0.7963	0.4611	0.5840	0.7950	0.4595	0.5824	0.7962	0.4590	0.5823	0.6786	0.6580	0.6682	0.6793	0.6581	0.6685	0.6790	0.6579	0.6683
		TransH	0.7974	0.4597	0.5832	0.7949	0.4587	0.5817	0.7948	0.4583	0.5814	0.6809	0.6592	0.6699	0.6781	0.6579	0.6678	0.5814	0.6780	0.6572
		TransD	0.7967	0.4604	0.5835	0.7959	0.4598	0.5829	0.7948	0.4596	0.5824	0.6804	0.6587	0.6694	0.6805	0.6591	0.6696	0.5824	0.6787	0.6578
		DistMult	0.7847	0.4551	0.5761	0.7910	0.4568	0.5792	0.7908	0.4567	0.5790	0.6712	0.6543	0.6627	0.6760	0.6560	0.6659	0.6759	0.6561	0.6659
		HolE	0.7611	0.4447	0.5614	0.7666	0.4473	0.5650	0.7724	0.4500	0.5687	0.6517	0.6426	0.6471	0.6569	0.6449	0.6508	0.6602	0.6475	0.6538
	BERT Emb	lastlayer							0.7924	0.4573	0.5799							0.6759	0.6559	0.6658
		lastlayer_no_stopw							0.7901	0.4559	0.5782							0.6746	0.6547	0.6645
	GRAPH Emb user + BERT Emb item	TransE u + lastlayer i							0.7951	0.4591	0.5821							0.6798	0.6581	0.6688
		TransE u + lastlayer_no_stopw i							0.7972	0.4597	0.5831							0.6792	0.6577	0.6683
		TransH u + lastlayer i							0.7949	0.4591	0.5820							0.6786	0.6575	0.6679
		TransH u + lastlayer_no_stopw i							0.7963	0.4587	0.5821							0.6797	0.6579	0.6686
		TransD u + lastlayer i							0.7969	0.4594	0.5828							0.6811	0.6587	0.6697
		TransD u + lastlayer_no_stopw i							0.7966	0.4586	0.5821							0.6804	0.6583	0.6692
		DistMult u + lastlayer i							0.7876	0.4563	0.5778							0.6720	0.6539	0.6628
		DistMult u + lastlayer_no_stopw i							0.7890	0.4567	0.5785							0.6751	0.6562	0.6655
		HolE u + lastlayer i							0.7763	0.4523	0.5715							0.6643	0.6504	0.6573
		HolE u + lastlayer_no_stopw i							0.7780	0.4520	0.5718							0.6661	0.6514	0.6586
	BERT Emb user + GRAPH Emb item	lastlayer u + TransE i							0.7915	0.4572	0.5796							0.6763	0.6565	0.6662
		lastlayer u + TransH i							0.7909	0.4566	0.5789							0.6764	0.6559	0.6660
		lastlayer u + TransD i							0.7901	0.4555	0.5779							0.6763	0.6562	0.6661
		lastlayer u + DistMult i							0.7904	0.4567	0.5789							0.6757	0.6555	0.6655
		lastlayer u + HolE i							0.7777	0.4518	0.5716							0.6647	0.6503	0.6574
		lastlayer_no_stopw u + TransE i							0.7915	0.4572	0.5796							0.6768	0.6569	0.6667
		lastlayer_no_stopw u + TransH i							0.7944	0.4583	0.5813							0.6766	0.6565	0.6664
		lastlayer_no_stopw u + TransD i							0.7911	0.4572	0.5795							0.6757	0.6560	0.6657
		lastlayer_no_stopw u + DistMult i							0.7917	0.4571	0.5795							0.6750	0.6560	0.6653
		lastlayer_no_stopw u + HolE i							0.7774	0.4510	0.5708							0.6667	0.6508	0.6586

Figure 30: Result of the Deep CBRS Amar Revisited first variant, where the KG embedding are generated from the item-properties graph.

MOVIELENS			TOP 5									TOP 10								
			256			512			768			256			512			768		
			Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
BASIC	GRAPH Emb	TransE	0.7963	0.4611	0.5840	0.7950	0.4595	0.5824	0.7962	0.4590	0.5823	0.6786	0.6580	0.6682	0.6793	0.6581	0.6685	0.6790	0.6579	0.6683
		TransH	0.7974	0.4597	0.5832	0.7949	0.4587	0.5817	0.7948	0.4583	0.5814	0.6809	0.6592	0.6699	0.6781	0.6579	0.6678	0.5814	0.6780	0.6572
		TransD	0.7967	0.4604	0.5835	0.7959	0.4598	0.5829	0.7948	0.4596	0.5824	0.6804	0.6587	0.6694	0.6805	0.6591	0.6696	0.5824	0.6787	0.6578
		DistMult	0.7847	0.4551	0.5761	0.7910	0.4568	0.5792	0.7908	0.4567	0.5790	0.6712	0.6543	0.6627	0.6760	0.6560	0.6659	0.6759	0.6561	0.6659
		HolE	0.7611	0.4447	0.5614	0.7666	0.4473	0.5650	0.7724	0.4500	0.5687	0.6517	0.6426	0.6471	0.6569	0.6449	0.6508	0.6602	0.6475	0.6538
	BERT Emb	lastlayer							0.7924	0.4573	0.5799							0.6759	0.6559	0.6658
		lastlayer_no_stopw							0.7901	0.4559	0.5782							0.6746	0.6547	0.6645
	GRAPH Emb user + BERT Emb item	TransE u + lastlayer i							0.7951	0.4591	0.5821							0.6798	0.6581	0.6688
		TransE u + lastlayer_no_stopw i							0.7972	0.4597	0.5831							0.6792	0.6577	0.6683
		TransH u + lastlayer i							0.7949	0.4591	0.5820							0.6786	0.6575	0.6679
		TransH u + lastlayer_no_stopw i							0.7963	0.4587	0.5821							0.6797	0.6579	0.6686
		TransD u + lastlayer i							0.7969	0.4594	0.5828							0.6811	0.6587	0.6697
		TransD u + lastlayer_no_stopw i							0.7966	0.4586	0.5821							0.6804	0.6583	0.6692
		DistMult u + lastlayer i							0.7876	0.4563	0.5778							0.6720	0.6539	0.6628
		DistMult u + lastlayer_no_stopw i							0.7890	0.4567	0.5785							0.6751	0.6562	0.6655
		HolE u + lastlayer i							0.7763	0.4523	0.5715							0.6643	0.6504	0.6573
		HolE u + lastlayer_no_stopw i							0.7780	0.4520	0.5718							0.6661	0.6514	0.6586
	BERT Emb user + GRAPH Emb item	lastlayer u + TransE i							0.7915	0.4572	0.5796							0.6763	0.6565	0.6662
		lastlayer u + TransH i							0.7909	0.4566	0.5789							0.6764	0.6559	0.6660
		lastlayer u + TransD i							0.7901	0.4555	0.5779							0.6763	0.6562	0.6661
		lastlayer u + DistMult i							0.7904	0.4567	0.5789							0.6757	0.6555	0.6655
		lastlayer u + HolE i							0.7777	0.4518	0.5716							0.6647	0.6503	0.6574
		lastlayer_no_stopw u + TransE i							0.7915	0.4572	0.5796							0.6768	0.6569	0.6667
		lastlayer_no_stopw u + TransH i							0.7944	0.4583	0.5813							0.6766	0.6565	0.6664
		lastlayer_no_stopw u + TransD i							0.7911	0.4572	0.5795							0.6757	0.6560	0.6657
		lastlayer_no_stopw u + DistMult i							0.7917	0.4571	0.5795							0.6750	0.6560	0.6653
		lastlayer_no_stopw u + HolE i							0.7774	0.4510	0.5708							0.6667	0.6508	0.6586

Figure 31: Result of the Deep CBRS Amar Revisited first variant, where the KG embedding are generated from the user-item-properties graph.

6.3.2 Second Variant (MIXED)

MOVIELENS			TOP 5									TOP 10								
			256			512			768			256			512			768		
			Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
MIXED	CONF 1: (GRAPH Emb item + BERT Emb item) + (GRAPH Emb user + BERT Emb user)	(TransE i + lastlayer i) + (TransE u + lastlayer u)							0.7917	0.4580	0.5803							0.6766	0.6564	0.6663
		(TransE i + lastlayer_no_stopw i) + (TransE u + lastlayer_no_stopw u)							0.7906	0.4566	0.5789							0.6762	0.6561	0.6660
		(TransH i + lastlayer i) + (TransH u + lastlayer u)							0.7903	0.4559	0.5782							0.6758	0.6563	0.6659
		(TransH i + lastlayer_no_stopw i) + (TransH u + lastlayer_no_stopw u)							0.7923	0.4576	0.5801							0.6767	0.6563	0.6663
		(TransD i + lastlayer i) + (TransD u + lastlayer u)							0.7930	0.4578	0.5805							0.6778	0.6573	0.6674
		(TransD i + lastlayer_no_stopw i) + (TransD u + lastlayer_no_stopw u)							0.7913	0.4581	0.5803							0.6768	0.6566	0.6665
		(DistMult i + lastlayer i) + (DistMult u + lastlayer u)							0.7922	0.4580	0.5800							0.6755	0.6557	0.6654
		(DistMult i + lastlayer_no_stopw i) + (DistMult u + lastlayer_no_stopw u)							0.7919	0.4575	0.5800							0.6753	0.6560	0.6655
		(HolE i + lastlayer i) + (HolE u + lastlayer u)							0.7851	0.4547	0.5759							0.6716	0.6533	0.6623
		(HolE i + lastlayer_no_stopw i) + (HolE u + lastlayer_no_stopw u)							0.7875	0.4552	0.5769							0.6742	0.6549	0.6644
	CONF 2: (GRAPH Emb item + GRAPH Emb user) + (BERT Emb item + BERT Emb user)	(TransE i + TransE u) + (lastlayer i + lastlayer u)	0.7784	0.4521	0.5720	0.7776	0.4530	0.5725	0.7885	0.4555	0.5774	0.6677	0.6521	0.6598	0.6653	0.6509	0.6580	0.6755	0.6554	0.6653
		(TransE i + TransE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7760	0.4514	0.5708	0.7794	0.4520	0.5722				0.6654	0.6507	0.6580	0.6686	0.6513	0.6598
		(TransH i + TransH u) + (lastlayer i + lastlayer u)				0.7772	0.4516	0.5713	0.7794	0.4512	0.5715				0.6651	0.6504	0.6577	0.6687	0.6514	0.6599
		(TransH i + TransH u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7769	0.4523	0.5718	0.7811	0.4521	0.5727				0.6639	0.6497	0.6567	0.6709	0.6528	0.6617
		(TransD i + TransD u) + (lastlayer i + lastlayer u)	0.7754	0.4510	0.5703	0.7788	0.4529	0.5727	0.7939	0.4579	0.5808	0.6644	0.6503	0.6573	0.6652	0.6507	0.6579	0.6775	0.6564	0.6668
		(TransD i + TransD u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7797	0.4526	0.5727	0.7859	0.4529	0.5747				0.6671	0.6516	0.6593	0.6716	0.6524	0.6619
		(DistMult i + DistMult u) + (lastlayer i + lastlayer u)				0.7582	0.4441	0.5602	0.7821	0.4521	0.5729				0.6505	0.6420	0.6462	0.6707	0.6528	0.6616
		(DistMult i + DistMult u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7617	0.4452	0.5620	0.7780	0.4521	0.5718				0.6521	0.6428	0.6474	0.6656	0.6503	0.6578
		(HolE i + HolE u) + (lastlayer i + lastlayer u)				0.7715	0.4502	0.5686	0.7899	0.4569	0.5790				0.6617	0.6487	0.6551	0.6748	0.6557	0.6651
		(HolE i + HolE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7309	0.4302	0.5416	0.7915	0.4571	0.5795				0.6296	0.6298	0.6297	0.6758	0.6560	0.6658

Figure 32: Result of the Deep CBRS Amar Revisited second variant, where the KG embedding are generated from the user-item graph.

MOVIELENS			TOP 5									TOP 10								
			256			512			768			256			512			768		
			Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
MIXED	CONF 1: (GRAPH Emb item + BERT Emb item) + (GRAPH Emb user + BERT Emb user)	(TransE i + lastlayer i) + (TransE u + lastlayer u)							0.7905	0.4567	0.5790							0.6756	0.6559	0.6656
		(TransE i + lastlayer_no_stopw i) + (TransE u + lastlayer_no_stopw u)							0.7918	0.4579	0.5803							0.6760	0.6565	0.6661
		(TransH i + lastlayer i) + (TransH u + lastlayer u)							0.7920	0.4582	0.5806							0.6754	0.6559	0.6655
		(TransH i + lastlayer_no_stopw i) + (TransH u + lastlayer_no_stopw u)							0.7923	0.4576	0.5801							0.6767	0.6563	0.6663
		(TransD i + lastlayer i) + (TransD u + lastlayer u)							0.7930	0.4578	0.5805							0.6778	0.6573	0.6674
		(TransD i + lastlayer_no_stopw i) + (TransD u + lastlayer_no_stopw u)							0.7913	0.4581	0.5803							0.6768	0.6566	0.6665
		(DistMult i + lastlayer i) + (DistMult u + lastlayer u)							0.7922	0.4580	0.5805							0.6755	0.6557	0.6654
		(DistMult i + lastlayer_no_stopw i) + (DistMult u + lastlayer_no_stopw u)							0.7920	0.4567	0.5793							0.6759	0.6557	0.6657
		(HolE i + lastlayer i) + (HolE u + lastlayer u)							0.7855	0.4545	0.5758							0.6722	0.6539	0.6629
		(HolE i + lastlayer_no_stopw i) + (HolE u + lastlayer_no_stopw u)							0.7895	0.4565	0.5785							0.6736	0.6545	0.6639
	CONF 2: (GRAPH Emb item + GRAPH Emb user) + (BERT Emb item + BERT Emb user)	(TransE i + TransE u) + (lastlayer i + lastlayer u)	0.7784	0.4521	0.5720	0.7769	0.4512	0.5709	0.7943	0.4581	0.5810	0.6677	0.6521	0.6598	0.6644	0.6500	0.6571	0.6783	0.6571	0.6675
		(TransE i + TransE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7937	0.4580	0.5808	0.7956	0.4581	0.5814				0.6763	0.6563	0.6661	0.6786	0.6570	0.6676
		(TransH i + TransH u) + (lastlayer i + lastlayer u)				0.7741	0.4512	0.5701	0.7953	0.4582	0.5814		0.5812		0.6626	0.6496	0.6560	0.6771	0.6567	0.6668
		(TransH i + TransH u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7921	0.4578	0.5803	0.7951	0.4579	0.5812				0.6751	0.6563	0.6656	0.6794	0.6575	0.6683
		(TransD i + TransD u) + (lastlayer i + lastlayer u)	0.7754	0.4510	0.5703	0.7843	0.4549	0.5758	0.7939	0.4579	0.5808	0.6644	0.6503	0.6573	0.6704	0.6528	0.6615	0.6775	0.6564	0.6668
		(TransD i + TransD u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7914	0.4575	0.5798	0.7859	0.4529	0.5747				0.6765	0.6569	0.6666	0.6716	0.6524	0.6619
		(DistMult i + DistMult u) + (lastlayer i + lastlayer u)				0.7841	0.4547	0.5756	0.7904	0.4561	0.5784				0.6705	0.6531	0.6617	0.6756	0.6553	0.6653
		(DistMult i + DistMult u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7727	0.4502	0.5689	0.7927	0.4566	0.5794				0.6630	0.6495	0.6562	0.6759	0.6555	0.6656
		(HolE i + HolE u) + (lastlayer i + lastlayer u)				0.7767	0.4515	0.5711	0.7862	0.4542	0.5758				0.6650	0.6507	0.6577	0.6715	0.6527	0.6620
		(HolE i + HolE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7734	0.4507	0.5695	0.7868	0.4542	0.5759				0.6618	0.6493	0.6555	0.6723	0.6533	0.6627

Figure 33: Result of the Deep CBRS Amar Revisited second variant, where the KG embedding are generated from the item-properties graph.

MOVIELENS			TOP 5									TOP 10								
			256			512			768			256			512			768		
			Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
MIXED	CONF 1: (GRAPH Emb item + BERT Emb item) + (GRAPH Emb user + BERT Emb user)	(TransE i + lastlayer i) + (TransE u + lastlayer u)							0.7930	0.4577	0.5804							0.6779	0.6574	0.6675
		(TransE i + lastlayer_no_stopw i) + (TransE u + lastlayer_no_stopw u)							0.7934	0.4578	0.5806							0.6769	0.6568	0.6667
		(TransH i + lastlayer i) + (TransH u + lastlayer u)							0.7945	0.4593	0.5821							0.6782	0.6581	0.6680
		(TransH i + lastlayer_no_stopw i) + (TransH u + lastlayer_no_stopw u)							0.7933	0.4588	0.5814							0.6765	0.6570	0.6666
		(TransD i + lastlayer i) + (TransD u + lastlayer u)							0.7930	0.4583	0.5809							0.6780	0.6576	0.6677
		(TransD i + lastlayer_no_stopw i) + (TransD u + lastlayer_no_stopw u)							0.7945	0.4586	0.5815							0.6796	0.6586	0.6689
		(DistMult i + lastlayer i) + (DistMult u + lastlayer u)							0.7898	0.4561	0.5782							0.6763	0.6569	0.6664
		(DistMult i + lastlayer_no_stopw i) + (DistMult u + lastlayer_no_stopw u)							0.7930	0.4568	0.5797							0.6779	0.6572	0.6674
		(HolE i + lastlayer i) + (HolE u + lastlayer u)							0.7875	0.4550	0.5768							0.6729	0.6546	0.6636
		(HolE i + lastlayer_no_stopw i) + (HolE u + lastlayer_no_stopw u)							0.7876	0.4556	0.5772							0.6740	0.6555	0.6646
	CONF 2: (GRAPH Emb item + GRAPH Emb user) + (BERT Emb item + BERT Emb user)	(TransE i + TransE u) + (lastlayer i + lastlayer u)	0.7919	0.4570	0.5796	0.7913	0.4573	0.5796	0.7956	0.4587	0.5819	0.6752	0.6561	0.6655	0.6751	0.6557	0.6653	0.6778	0.6568	0.6672
		(TransE i + TransE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7937	0.4580	0.5808	0.7960	0.4595	0.5827				0.6763	0.6563	0.6661	0.6788	0.6570	0.6677
		(TransH i + TransH u) + (lastlayer i + lastlayer u)				0.7884	0.4552	0.5772	0.7980	0.4606	0.5841				0.6739	0.6555	0.6645	0.6802	0.6581	0.6690
		(TransH i + TransH u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7921	0.4579	0.5803	0.7990	0.4611	0.5847				0.6751	0.6563	0.6656	0.6807	0.6587	0.6695
		(TransD i + TransD u) + (lastlayer i + lastlayer u)	0.7827	0.4537	0.5744	0.7956	0.4588	0.5820	0.7987	0.4605	0.5842	0.6701	0.6531	0.6615	0.6777	0.6575	0.6674	0.6811	0.6583	0.6695
		(TransD i + TransD u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7914	0.4575	0.5798	0.7964	0.4596	0.5828				0.6765	0.6569	0.6666	0.6797	0.6575	0.6684
		(DistMult i + DistMult u) + (lastlayer i + lastlayer u)				0.7789	0.4521	0.5721	0.7929	0.4564	0.5793				0.6660	0.6511	0.6584	0.6769	0.6563	0.6664
		(DistMult i + DistMult u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7727	0.4502	0.5689	0.7903	0.4562	0.5784				0.6630	0.6495	0.6562	0.6748	0.6555	0.6650
		(HolE i + HolE u) + (lastlayer i + lastlayer u)				0.7790	0.4519	0.5720	0.7940	0.4584	0.5812				0.6659	0.6510	0.6584	0.6766	0.6567	0.6665
		(HolE i + HolE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)				0.7734	0.4507	0.5695	0.7908	0.4569	0.5792				0.6618	0.6493	0.6555	0.6747	0.6553	0.6648

Figure 34: Result of the Deep CBRS Amar Revisited second variant, where the KG embedding are generated from the user-item-properties graph.

6.3.3 Third Variant (EXTENDED)

MOVIELENS			TOP 5									TOP 10								
			256			512			768			256			512			768		
			Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
EXTENDED	CONF 2: (GRAPH Emb item + GRAPH Emb user) + (BERT Emb item + BERT Emb user)	(TransE i + TransE u) + (lastlayer i + lastlayer u)							0.7945	0.4583	0.5813							0.6794	0.6580	0.6685
		(TransE i + TransE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7957	0.4594	0.5825							0.6791	0.6584	0.6686
		(TransH i + TransH u) + (lastlayer i + lastlayer u)							0.7933	0.4580	0.5808							0.6779	0.6568	0.6672
		(TransH i + TransH u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7934	0.4574	0.5802							0.6774	0.6568	0.6670
		(TransD i + TransD u) + (lastlayer i + lastlayer u)							0.7951	0.4590	0.5820							0.6805	0.6585	0.6693
		(TransD i + TransD u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7945	0.4582	0.5812							0.6799	0.6580	0.6688
		(DistMult i + DistMult u) + (lastlayer i + lastlayer u)							0.7933	0.4582	0.5809							0.6770	0.6570	0.6669
		(DistMult i + DistMult u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7945	0.4585	0.5814							0.6778	0.6575	0.6675
		(HolE i + HolE u) + (lastlayer i + lastlayer u)							0.7835	0.4532	0.5742							0.6702	0.6524	0.6612
		(HolE i + HolE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7866	0.4550	0.5765							0.6727	0.6544	0.6634

Figure 35: Result of the Deep CBRS Amar Revisited third variant, where the KG embedding are generated from the user-item graph.

MOVIELENS			TOP 5									TOP 10								
			256			512			768			256			512			768		
			Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
EXTENDED	CONF 2: (GRAPH Emb item + GRAPH Emb user) + (BERT Emb item + BERT Emb user)	(TransE i + TransE u) + (lastlayer i + lastlayer u)							0.7937	0.4583	0.5811							0.6777	0.6569	0.6671
		(TransE i + TransE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7957	0.4592	0.5823							0.6780	0.6572	0.6675
		(TransH i + TransH u) + (lastlayer i + lastlayer u)							0.7939	0.4585	0.5813							0.6768	0.6562	0.6663
		(TransH i + TransH u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7945	0.4577	0.5808							0.6788	0.6577	0.6681
		(TransD i + TransD u) + (lastlayer i + lastlayer u)							0.7930	0.4569	0.5798							0.6781	0.6573	0.6675
		(TransD i + TransD u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7928	0.4581	0.5807							0.6767	0.6568	0.6666
		(DistMult i + DistMult u) + (lastlayer i + lastlayer u)							0.7954	0.4582	0.5814							0.6766	0.6559	0.6661
		(DistMult i + DistMult u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7940	0.4569	0.5800							0.6776	0.6565	0.6669
		(HolE i + HolE u) + (lastlayer i + lastlayer u)							0.7845	0.4533	0.5746							0.6702	0.6521	0.6611
		(HolE i + HolE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7877	0.4558	0.5774							0.6722	0.6534	0.6627

Figure 36: Result of the Deep CBRS Amar Revisited third variant, where the KG embedding are generated from the item-properties graph.

MOVIELENS			TOP 5									TOP 10								
			256			512			768			256			512			768		
			Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
EXTENDED	CONF 2: (GRAPH Emb item + GRAPH Emb user) + (BERT Emb item + BERT Emb user)	(TransE i + TransE u) + (lastlayer i + lastlayer u)							0.7980	0.4596	0.5833							0.6804	0.6583	0.6692
		(TransE i + TransE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7992	0.4600	0.5839							0.6805	0.6579	0.6690
		(TransH i + TransH u) + (lastlayer i + lastlayer u)							0.7994	0.4611	0.5849							0.6816	0.6585	0.6698
		(TransH i + TransH u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.8000	0.4610	0.5849							0.6821	0.6588	0.6702
		(TransD i + TransD u) + (lastlayer i + lastlayer u)							0.8002	0.4598	0.5840							0.6811	0.6583	0.6695
		(TransD i + TransD u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.8003	0.4589	0.5833							0.6828	0.6586	0.6705
		(DistMult i + DistMult u) + (lastlayer i + lastlayer u)							0.7909	0.4566	0.5789							0.6746	0.6557	0.6650
		(DistMult i + DistMult u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7923	0.4573	0.5799							0.6782	0.6571	0.6674
		(HolE i + HolE u) + (lastlayer i + lastlayer u)							0.7931	0.4574	0.5802							0.6759	0.6561	0.6658
		(HolE i + HolE u) + (lastlayer_no_stopw i + lastlayer_no_stopw u)							0.7908	0.4557	0.5782							0.6743	0.6545	0.6643

Figure 37: Result of the Deep CBRS Amar Revisited third variant, where the KG embedding are generated from the user-item-properties graph.

7 Reproducibility

For KG embedding generation see the README of the swapUniba's OpenKE fork [1]. The dataset used for user-item-properties and item-properties graphs can be download here: `https://drive.google.com/drive/folders/1ug44hplNbxGT3B7kaWAJJG6-qLcUreGX?usp=sharing`. For training with Deep CBRS Amar see the README of the swapUniba's Deep CBRS Amar repository [2].

[1] https://github.com/swapUniba/OpenKE
[2] https://github.com/swapUniba/Deep_CBRS_Amar

In this section will be only shown the training and testing phase of all Deep CBRS AMAR Revisited architectures, that are performed through some Python scripts which exploits a Google Colab notebook and the Tensorflow libraries. A python notebook file has been employed to call the train and test files. In this way, the results can be easily replicated or integrated.

Before executing some experiments, clone the repository end move into the project folder:

```
!git clone https://github.com/cenzy/Deep_CBRS_Amar.git
cd Deep_CBRS_Amar
```

7.1 BASIC Architecture

BASIC architecture consists of four sub-architectures. The following scripts show how to train and test each BASIC sub-architecture.

1. user and item as KG embedding

```
#train
!python train-predict-BASIC-graphemb.py
        path/to/embeddings.json
        result_path/
        prediction_path/
#test
!python top-predictions-BASIC-graphemb.py
        path/to/embeddings.json
        result_path/
        prediction_path/
```

2. user and item as BERT embedding

```
#train
!python train-predict-BASIC-bertemb.py
        path/to/user_embeddings.json
        path/to/item_embeddings.json
        result_path/
        prediction_path/
#test
!python top-predictions-BASIC-bertemb.py
        path/to/user_embeddings.json
        path/to/item_embeddings.json
        result_path/
        prediction_path/
```

3. user as KG embedding and item as BERT embedding

```
#train
!python train-predict-BASIC-bertandgraphemb.py
        path/to/KG_embeddings.json
        path/to/BERT_item_embeddings.json
        result_path/
        prediction_path/
        1 #first configuration where with user as KG embedding
#test
!python top-predictions-BASIC-bertandgraphemb.py
        path/to/KG_embeddings.json
        path/to/BERT_item_embeddings.json
        result_path/
        prediction_path/
        1 #first configuration where with user as KG embedding
```

4. user as BERT embegging and item as KG embedding

```
#train
!python train-predict-BASIC-bertandgraphemb.py
        path/to/user_embeddings.json
        path/to/item_embeddings.json
        result_path/
        prediction_path/
        2 #second configuration where with user as BERT embedding
#test
!python top-predictions-BASIC-bertandgraphemb.py
        path/to/BERT_user_embeddings.json
        path/to/KG_embeddings.json
        result_path/
        prediction_path/
        2 #second configuration where with user as BERT embedding
```

7.2 MIXED Architecture

MIXED architecture consists of twe sub-architectures. The following scripts show how to train and test each BASIC sub-architecture.

1. (KG embeddings item + BERT embeddings item) + (KG embeddings user + BERT embeddings user)

```
#train
!python train-predict-MIXED.py
        path/to/BERT_user_embeddings.json
        path/to/BERT_user_embeddings.json
        path/to/KG_embeddings.json
        result_path/
        prediction_path/
        1 #first configuration
#test
!python top-predictions-MIXED.py
        path/to/BERT_user_embeddings.json
        path/to/BERT_user_embeddings.json
        path/to/KG_embeddings.json
        result_path/
        prediction_path/
        1 #first configuration
```

2. (KG embeddings user + KG embeddings item) + (BERT embeddings user + BERT embeddings user)

```
#train
!python train-predict-MIXED.py
        path/to/BERT_user_embeddings.json
        path/to/BERT_user_embeddings.json
        path/to/KG_embeddings.json
        result_path/
        prediction_path/
        2 #second configuration
#test
!python top-predictions-MIXED.py
        path/to/BERT_user_embeddings.json
        path/to/BERT_user_embeddings.json
        path/to/KG_embeddings.json
        result_path/
```

```
        prediction_path/
        2 #second configuration
```

3. (KG embeddings user + KG embeddings item) + (BERT embeddings user + BERT embeddings user) #tron KG embeddings dimension lower than 768

```
#train
!python train-predict-MIXED-strategy.py
        path/to/BERT_user_embeddings.json
        path/to/BERT_user_embeddings.json
        path/to/KG_embeddings.json
        result_path/
        prediction_path/
#test
!python top-predictions-MIXED-strategy.py
        path/to/BERT_user_embeddings.json
        path/to/BERT_user_embeddings.json
        path/to/KG_embeddings.json
        result_path/
        prediction_path/
```

For evaluating the top five and the top 10 movies for each user:

```
!python evaluate_results.py path/to/test2id_all_pred.txt prediction_path/
```

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[2] Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. Openke: An open toolkit for knowledge embedding. In *Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations*, pages 139–144, 2018.

[3] Cataldo Musto, Claudio Greco, Alessandro Suglia, and Giovanni Semeraro. Ask me any rating: A content-based recommender system based on recurrent neural networks. In *IIR*, 2016.

[4] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.