

The Unusual Suspects: Deep Learning Based Mining of Interesting Entity Trivia from Knowledge Graphs

Nausheen Fatma

LTRC, IIT-H
Hyderabad, India
nausheen.fatma@research.iiit.ac.in

Manoj K. Chinnakotla

Microsoft
Hyderabad, India
manojc@microsoft.com

Manish Shrivastava

LTRC, IIT-H
Hyderabad, India
m.shrivastava@iiit.ac.in

Abstract

Trivia is any fact about an entity which is *interesting* due to its *unusualness*, *uniqueness* or *unexpectedness*. Trivia could be successfully employed to promote user engagement in various product experiences featuring the given entity. A Knowledge Graph (KG) is a semantic network which encodes various facts about entities and their relationships. In this paper, we propose a novel approach called DBpedia Trivia Miner (DTM) to automatically mine trivia for entities of a given domain in KGs. The essence of DTM lies in learning an Interestingness Model (IM), for a given domain, from human annotated training data provided in the form of interesting facts from the KG. The IM thus learnt is applied to extract trivia for other entities of the same domain in the KG. We propose two different approaches for learning the IM - a) A Convolutional Neural Network (CNN) based approach and b) Fusion Based CNN (F-CNN) approach which combines both hand-crafted and CNN features. Experiments across two different domains - Bollywood Actors and Music Artists reveal that CNN automatically learns features which are relevant to the task and shows competitive performance relative to hand-crafted feature based baselines whereas F-CNN significantly improves the performance over the baseline approaches which use hand-crafted features alone. Overall, DTM achieves an F1 score of 0.81 and 0.65 in Bollywood Actors and Music Artists domains respectively.

1 Introduction

In recent years, significant progress has been made in curating large web-scale semantic networks, popularly known as Knowledge Graphs (KGs), which encode various facts about entities and their relationships (Berners-Lee et al. 2001; Bizer, Heath, and Berners-Lee 2009; Auer et al. 2007). Some of the popular freely available knowledge graphs are DBpedia (Auer et al. 2007), YAGO2 (Suchanek, Kasneci, and Weikum 2007) and Wikidata (Vrandečić and Krötzsch 2014). For example, the English version of DBpedia alone contains more than 4.22 million entities and close to 3 billion relations (RDF triples). Besides, large commercial companies like Google, Microsoft, Yahoo! and Baidu have been curating their own KGs. Although, a significant portion of these facts were extracted from unstructured sources like

Wikipedia, KGs also include data from other rich sources such as - crowd-sourcing, CIA World Factbook *etc.*

A Trivia is any fact about an entity which is *interesting* due to its *unusualness*, *unexpectedness*, *uniqueness* or *weirdness* when compared to other entities of similar type (Prakash et al. 2015). For example, Yuvraj Singh, a popular cricket player from the Indian cricket team, authored a book called ‘*Test of my Life*’ which is an unexpected event for a cricketer since very few of them have authored books.

Trivia, when presented in the form of either a question or a factoid, helps in attracting human attention since it appeals to their innate sense of curiosity, inquisitiveness and appreciating novelty (Attfield et al. 2011; O’Brien and Toms 2010). Due to this, studies have shown that trivia could be used to improve user engagement in various product experiences featuring the entity (Wizard 2013; Prakash et al. 2015). For example, during ICC Cricket World Cup 2015, the Bing search engine offered a unique experience¹ for cricket related queries which included trivia about teams and players. As a result, they observed significant improvement in user engagement with each user watching at least 10 trivia per cricket query. In a pure sense, the notion of human interestingness is a psychological, cognitive and subjective process (Varela, Thompson, and Rosch 1993). However, there are usually some facts for which there would be significant agreement, between a majority of people, regarding their interestingness. In this work, we restrict ourselves to such a majoritarian notion of interestingness and leave the personalized subjective angle for future work.

In spite of the utility of trivia, the curation process remains mostly manual which is hard to scale across millions of entities on the web. Recently, some attempts (Prakash et al. 2015; Michael Gamon 2014) have been made to automatically mine interesting facts or spans of text from unstructured Wikipedia or normal text. However, such techniques rely heavily on the textual representation of facts or context features to learn about interestingness such as the presence of superlative (greatest, longest, best), contradictory words (although, however), exclusive words (is the only), verbs within the sentence indicating the core activity (performing a stunt) *etc.* On the other hand, in KGs, since the knowledge is

Copyright © 2017, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

¹<http://www.windowcentral.com/bings-cricket-world-cup-coverage-includes-predictions-polls-and-more>

represented in the form of a relation tuple with (subject, relation, object), the above features will not be available. Due to this, the interestingness of a relation needs to be inferred based on other features such as the unusuality of a given relation with respect to other entities in the same domain *etc.*

Recently, deep learning based techniques have shown significant performance gains across various human intelligence tasks (Krizhevsky, Sutskever, and Hinton 2012; Collobert et al. 2011; Dahl 2015). They automatically learn the feature representations required for the task while simultaneously modeling the target function. Inspired by this, in this paper, we propose a novel deep learning based approach called “DBpedia Trivia Miner (DTM)” which mines interesting trivia/facts from structured KGs such as DBpedia. For a chosen domain, given manually annotated facts which are interesting, DTM learns an “*Interestingness Model (IM)*” which captures the notion of interestingness. We propose two different variants of learning an IM - a) A Convolutional Neural Network (CNN) based approach and b) Fusion Based CNN (F-CNN) approach. Inspired by previous work in this direction (Suggu et al. 2016a), our F-CNN leverages the advantages of both hand-crafted features and deep learnt features and is trained end-end as a single model. Experiments across two different domains - Bollywood Actors and Music Artists reveal that CNN automatically learns features which are relevant to the task and shows competitive performance relative to hand-crafted feature based baselines. However, F-CNN significantly improves the performance over the baseline approaches which use hand-crafted features alone. Overall, the proposed DTM system achieves an F1 score of 0.81 and 0.65 in Bollywood Actors and Music Artists domains respectively. To summarize, the following are the main contributions of our paper:

- We introduce the research problem of mining interesting facts for an entity in a chosen domain of KG
- We propose a novel deep learning based approach for the above task which allows direct *end-end learning* of interestingness avoiding design of hand-crafted features while offering competitive performance
- We also release our dataset to the research community interested in the problem of mining interesting trivia from KGs

The rest of the paper is organized as follows: Section 2 describes the related work in this area. Section 3 presents the details of DTM such as architecture, IM and various features used. Section 4 describes the experimental set-up, details of evaluation datasets and evaluation metrics. Section 5 presents the results of our system. Finally, Section 6 concludes the paper.

2 Related Work

Interestingness has been a well studied area of research in knowledge discovery and data mining community (McGarry 2005; Geng and Hamilton 2006; Kontonasios, Spyropoulou, and De Bie 2012). However, it has not been explored much for data represented in the form of a KG. There is also work on uncovering interesting structural

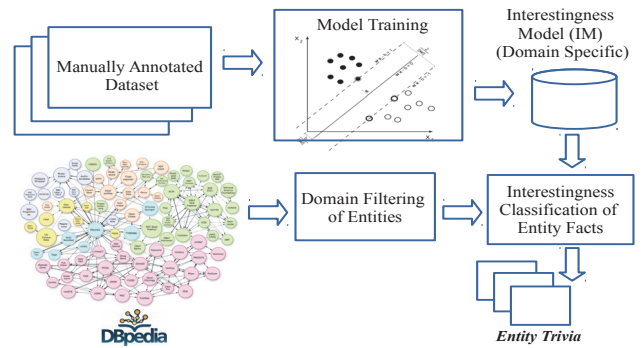


Figure 1: DBpedia Trivia Miner (DTM) System Architecture

anomalies/patterns/clusters from graphs (Akoglu, Tong, and Koutra 2015; Perozzi et al. 2014). Such work is more focused towards identifying clusters of nodes which structurally deviate from the entire graph and does not make use of the textual descriptions for relations (edges), objects usually available for KGs.

Recently, researchers (Prakash et al. 2015; Michael Gannon 2014) have tried to mine interesting phrases or sentences (similar to trivia) from unstructured text of Wikipedia. However, their interestingness model relies a lot on the language features of written text such as presence of superlatives, contradictory words and other contextual hints in the syntax which are not found in KG facts. Also, instead of a pure hand-crafted feature based model, we propose a hybrid model which uses deep learning to automatically learn features which are relevant to the task while also leveraging the hand-crafted features provided by humans.

The closest to our work was done by Mahesh et al. (Mahesh and Karanth 2015). They built a system *Smart-Aleck* which generates interesting facts from YAGO. Although, they define six levels of interestingness, they mainly rely on - a) availability of user ratings from public data sources for inferring interesting facts and b) applying various quantifiers, grouping operators and maximum, minimum operators across entity data. They do not share many important details on how these operators are applied, on what kind of data, and what fraction of the applied quantifiers turn out to be truly interesting. Moreover, they didn’t perform any experimental evaluation of their approach and ideas. Due to this, it was hard to reproduce their results.

To the best of our knowledge, we are the first to propose an approach for mining trivia from KGs using machine learning along with detailed experimental evaluation and analysis of results.

3 DBpedia Trivia Miner

In this section, we describe the details of our system DBpedia Trivia Miner (DTM).

Figure 1 shows the architecture of DTM. DTM takes manually annotated training data for a given domain in the form of a fact (entity, relation/predicate, object) and its true label and learns a domain-specific Interestingness Model (IM). Later, for each entity in the given domain, the IM is applied

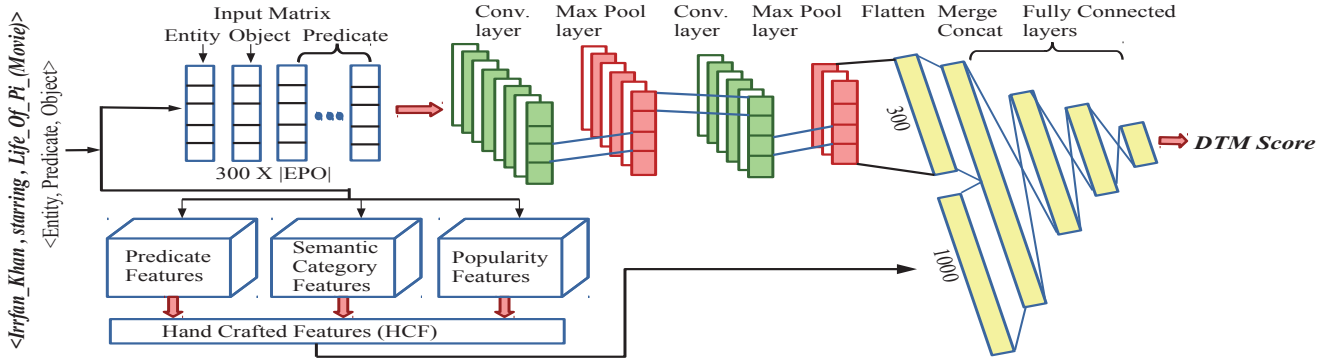


Figure 2: Fusion Based CNN (F-CNN) for learning the Interestingness Model (IM)

which filters the interesting facts as *trivia*. For learning IM, we model the notion of interestingness as a binary classification problem - interesting vs. boring. We show two different ways of learning the IM using a - a) Convolutional Neural Network (CNN) and b) Fusion based CNN (F-CNN) (Suggu et al. 2016a; 2016b) which combines both hand-crafted features and CNN features.

3.1 Convolutional Neural Network for IM

In this approach, we use a CNN to learn the binary classification model from the given training data. The CNN takes a KG fact as input in the form of a triple (entity, object, relation) where the entity, object and relation/predicate are represented using their word embeddings in the same order mentioned above. For entities and objects, instead of words, we use their word2vec DBpedia entity embeddings in 300 dimensions generated using an open source tool². The tool is a slightly modified version of word2vec where instead of treating the entity/object as separate words, they are treated as a single unit while generating word2vec embeddings. For example, the entire phrase “*The Lord of the Rings*” is treated as a single entity and will have a single embedding. For relation/predicate, we use their word2vec embeddings in 300 dimensions. We fix the length of the predicate to 8 words since most predicates are short and have length less than 8 words. If the predicate length is less than 8, we pad the remaining word vectors with zero. If the length is greater than 8, we ignore the remaining words. The input matrix $((8 + 2) \times 300)$ is then convolved through two rounds of convolution, pooling and non-linearity layers to get a 300 dimensional feature representation of the input triple. We use max-pooling for the pooling layer and Rectified Linear Unit (ReLU) (Nair and Hinton 2010) as the non-linearity layer. The 300-dimensional vector representation passes through the second stage Neural Network (NN) consisting of fully connected layers. These layers model the various interactions between the learnt features and finally outputs a probabilistic score indicative of the interestingness of the given fact. Although, the system was described in steps, it is trained as a single end-end differentiable system.

²<https://github.com/idio/wiki2vec>

3.2 Hand Crafted Features (HCF)

Given a fact triple (entity, relation/predicate, object), we compute various features which help in capturing unusualness of an entity fact.

• Predicate Features :

Inverse Entity Frequency (IEF) : In Information Retrieval (IR), Inverse Document Frequency (IDF) (Sparck Jones 1972) is usually used for measuring the importance of a term. We apply a similar intuition for our entity relationships. For a given relation/predicate p , we find how rare the predicate is with respect to the domain D . We believe that entities belonging to the same domain will have many predicates in common and hence rarer predicates would be good candidates for interestingness. We define the KG equivalent of IDF for a given predicate p as follows:

$$IEF(p) = \log \frac{|D|}{n_p}$$

where $|D|$ denotes the total number of entities in domain set D , and n_p denotes the the number of entities in domain set D in which the predicate p occurs. For example, in Bollywood actors domain, a predicate like `starring` is present in almost all the entities and hence will have a low IEF value whereas a predicate like `author` will have a high IEF since not many actors in Bollywood have authored a book.

Predicate Frequency-Inverse Entity Frequency (PF-IEF): Another well known term weighting technique in IR is TF-IDF (Salton and Buckley 1988). We define its KG equivalent, for an entity e and predicate p as:

$$PF-IEF(p, e) = PF(p, e) \times IEF(p)$$

$PF(p, e)$ = Number of RDF triples for entity e in which predicate p occurs.

Predicate Bag of Words : This feature converts the text of a predicate into a bag of words and then makes features out of them. Some predicates are potentially more interesting than the rest. For example, `owner`,

founder, leader.

- **Semantic Category Features:**

Object Type : We find out the semantic type of the relation object. Given an RDF triple $\langle e, p, o \rangle$, we find the semantic type of the object o such as Person, Place, Organization, Work, EducationalInstitution, Film, *etc.* from DBpedia. This could be a useful feature in capturing unusual associations in some predicates. For example, consider a triple $\langle \text{Jerry Garcia, named after of, 4442 Garcia} \rangle$ which conveys that the planet was named after him. We find the type information for o from DBpedia as Planet which is interesting as the type Planet is a very rare occurrence for a Music domain.

Object Categories : For an RDF triple $\langle e, p, o \rangle$, we get the various semantic tags associated with the object o . Such tags are usually defined using the “subject” predicate. For example, consider the triple $\langle \text{Farhan Akhtar, owner, Mumbai Tennis Masters} \rangle$. The entity Mumbai Tennis Masters falls into the category like Tennis Teams, and Sports Teams, which is interesting as most of the entities in Bollywood domain with predicate owner has categories like Companies based in Mumbai, Film production companies, *etc.*

Average Category and Type Entropy : In a given domain, most of the objects are associated with some common categories and types which are semantic descriptive tags. For example, the most common object types in the Music domain are MusicalWork, MusicGroup, Album, *etc.* Any unusual or rare object category or type may turn a fact into a trivia. For example, for the triple $\langle \text{Celine Dion, key person, Cystic Fibrosis Canada} \rangle$, the types of the object are Organization, Non-ProfitOrganisation, CharitiesBasedInCanada, FinancialInstitution, SocialGroup. Since, some of these types are rarely associated with Music domain, it is a potential trivia.

We estimate the distribution of category and type tags for each domain and measure the extent of surprise of an object category tag or type set C of a given triple using “Average Category/Type Entropy” which is defined as follows:

$$E(C|M) = \frac{1}{|C|} \sum_{c \in C} -\log_2 Pr(c|M)$$

where $Pr(c|M)$ is the probability of the category/type tag c across the entire domain M . E is high if there are many rare categories/types associated with the object.

- **Popularity :** Trivia of an entity with higher popularity may generate greater surprise as opposed to a trivia about a relatively obscure entity. Number of in-links

Domain	#Entities	#Triples	#Interesting Triples
Bollywood	100	11322	464
Music	100	30514	392

Table 1: Dataset Statistics

to a web page is usually indicative of its popularity. For example, the triple $\langle \text{Michael Jackson, owner, Bubbles(chimpanzee)} \rangle$. Since, Michael Jackson is popular, people may find unusual facts related to him as interesting. Hence, for each RDF triple $\langle e, p, o \rangle$, we estimate the relative popularity of the entity e and object o using the number of in-links to their wikipedia pages. We calculate the number of in-links to the Wikipedia page using the Mediawiki Awk API³.

3.3 Fusion Based CNN for IM

Figure 2 shows the architecture of F-CNN for IM. As described in Suggu et al. (Suggu et al. 2016a; 2016b), a F-CNN combines both HCF and automatic features learnt using the CNN in a single model. The F-CNN is similar to a CNN in the first stage. It takes the KG fact as input in the same representation format as prescribed for the CNN. Later, the input fact passes through two rounds of convolution, max-pooling and non-linearity layers to produce a 300-dimensional feature representation (lets call this CNN-FR). In parallel, using the input fact, F-CNN also computes the hand-crafted features described earlier (lets call this HC-FR). In the second stage, these two feature representations are combined together (CNN-FR + HC-FR) and fed as input to the second stage NN which consists of fully connected layers in which the final output node gives the probability of the target class. To avoid unnecessarily processing a large number of non-informative HCF features through the second stage NN, we perform basic feature selection and then only pick the top 1000 most informative features amongst them. For basic feature selection, we choose an embedded model with regularization approach (Tang, Alelyani, and Liu 2014; Ma and Huang 2008) in which we train a Linear SVM (SVM-L) with L1 regularization and choose the top 1000 features sorted as per their feature importance. For each domain, the SVM-L training for feature selection was performed using a random sample of 5000 facts sampled from the dataset of corresponding domain.

3.4 Training

We train the parameters of CNN and F-CNN with an objective to maximize their predication accuracy given the target classes. We perform 10-fold cross validation on the evaluation dataset. Each random fold was divided into training, validation and test sets. The training set consisted of the entity triples along with their true binary label. We train the CNN and F-CNN using the training set and tune the hyper-parameters of the network using the validation set. Since, the

³<https://github.com/greencardamom/MediaWikiAwkAPI>

Hyperparameter	CNN	F-CNN
Batch size	128	256
Convolution Layers	2	2
Convolution Region Size	3	3
Max Pooling Units	2	2
Stride	1	1
Non-Linearity	ReLU	ReLU
Optimizer	SGD	SGD
Momentum	0.6	0.8
Dropout Rate	0.2	0.1
Fully Connected Layers	2	4

Table 2: Tuned Hyper-parameters which were used for training CNN and F-CNN Models

Model	Bollywood Actors			Music Artists		
	Prec.	Rec.	F1	Prec.	Rec.	F1
F-CNN	0.83*	0.79	0.81*	0.72*	0.59	0.65*
CNN	0.79	0.72	0.75*	0.69	0.55	0.59*
GBC	0.77	0.67	0.71	0.69	0.41	0.48
SVC-RBF	0.71	0.71	0.71	0.53	0.58	0.54
SVC-L	0.66	0.81	0.72	0.5	0.68	0.55

Table 3: 10-Fold Cross Validation Results: DTM variants (CNN and F-CNN) in comparison with other baseline approaches. Results marked with a * were found to be statistically significant with respect to the nearest baseline at 95% confidence level ($\alpha = 0.05$) when tested using a two-tailed paired t-test. All the models were trained sensitive to class imbalance.

number of hyper-parameters and their combinations is high for CNN and F-CNN, we tuned their hyper-parameters using the validation set of the first fold alone and then use the same combination across the remaining folds. The optimal hyper-parameter configuration thus found is shown in Table 2. If t is the true label and o is the output of the network with the current weight configuration, we use Binary Cross Entropy (BCE) as the loss function which is calculated as follows:

$$BCE(t, o) = -(t \cdot \log(o) + (1 - t) \cdot \log(1 - o))$$

We use Stochastic Gradient Descent (SGD) as the optimization routine and the models were trained by minimizing the above loss function in a batch size of n which was tuned differently for CNN and F-CNN and given in Table 2.

4 Experimental Setup

In this section, we describe our experimental setup which includes details related to our dataset, evaluation metrics and baseline approaches.

4.1 Dataset

For our experiments, we chose two popular domains within DBpedia - a) Bollywood Actors (Indian Movie Actors) and

Rank	Bollywood Actors		Music Artists	
	GBC	SVC-L	GBC	SVC-L
1	IEF	IEF	IEF	IEF
2	Ent.Inlinks	<i>distributor</i>	Ent.Inlinks	<i>business</i>
3	Cat.Entropy	<i>writer</i>	<i>players</i>	<i>birthplace</i>
4	PF-IEF	PF-IEF	<i>activism</i>	<i>recipients</i>
5	<i>lead</i>	<i>people</i>	<i>official</i>	<i>multi</i>
6	<i>writer</i>	<i>bollywood</i>	PF-IEF	<i>lebanese</i>
7	Obj.Inlinks	<i>culture</i>	<i>drowning</i>	<i>activism</i>
8	<i>beauty</i>	<i>men</i>	<i>armenian</i>	<i>australia</i>
9	TypeEntropy	<i>playback</i>	<i>canadian</i>	<i>drowning</i>
10	<i>ancient</i>	<i>child</i>	<i>multi</i>	<i>musician</i>

Table 4: Top 10 Informative Features for the SVM-L and GBC Baselines.

b) Music Artists. The above domains were chosen since we believe an average human annotator available to us would have sufficient knowledge to judge the interestingness of the facts in these domains. For each of the above domains, we extracted all the entities and their RDF relationships from DBpedia - 1113 entities, 58633 RDF relationships (Bollywood Actors) and 9232 entities, 665631 RDF relationships (Musical Artists) respectively. Since, an exhaustive manual evaluation of all the above relations across entities was not practically feasible, we chose a smaller subset of entities for our evaluation purpose. For the Bollywood Actors domain, we chose the most popular 100 actors from the above list who were either - a) listed in the Forbes top 100 celebrity list in the past four years⁴ or b) listed in the IMDB top 100 Bollywood actors of all time⁵. For Musical Artists, we selected the top 100 artists from the Billboard⁶ list for the past five years.

For getting the gold standard label for the interestingness of a fact, we employed five independent human annotators who were exclusively used for this task. The annotators were given clear guidelines and training on how to judge the interestingness of facts along with some sample judgments from each domain. For each domain, from the selected 100 entities, we extracted all their associated facts (triples) and asked judges to rate them for interestingness on a binary class scale: *Interesting* or *Boring*. The majority judgment from the judges was then taken as the true interestingness label for each fact. The inter-annotator agreement using the Kappa statistic (Fleiss 1971) was found to be 0.52 and 0.48 for the Bollywood and the Music domain respectively, which is a moderate agreement.

The final statistics regarding the dataset along with the annotated judgments for each domain is given in Table 1. In the interest of reproducibility of our results and promoting research in this area, we are making our entire dataset public for research purposes. It can be downloaded from the

⁴<http://forbesindia.com/lists/2015-celebrity-100/1519/1>

⁵<http://www.imdb.com/list/ls074758327/>

⁶<http://www.billboard.com/artists/top-100>

DTM Variant	Entity	Trivia	Description (from Wikipedia Page)
F-CNN	Marvin Gaye (<i>Music Artist</i>)	<Marvin Gaye, subject, United States Air Force airmen>	A 17-year-old Marvin ran away from home and joined the US Air Force as a Basic Airman.
	Sanjay Dutt (<i>Bollywood Actor</i>)	<Sanjay Dutt, founder, Super Fight League>	The Super Fight League is an India based mixed martial arts promotion which is also biggest in Asia.
	Akon (<i>Music Artist</i>)	<Akon, owner, Akon Lighting Africa>	Akon Lighting Africa was started in 2014 and aims to provide electricity by solar energy in Africa.
CNN	Farhan Akhtar (<i>Bollywood Actor</i>)	<Farhan Akhtar, owner, Mumbai Tennis Masters>	Farhan Akhtar is the owner of Mumbai Tennis Masters Club.
	Tupak Shakur (<i>Music Artist</i>)	<Tupak Shakur, subject, Murdered Rappers>	On September 7, 1996, Shakur was fatally shot in a drive-by shooting at Las Vegas, Nevada.

Table 5: Representative Samples of Interesting Trivia mined using two variants of DTM.

following location⁷.

4.2 Baseline Approaches and Evaluation Metrics

To demonstrate the effectiveness of DTM, we compare it with three purely hand-crafted feature based baselines: Linear SVM (SVC-L), SVM with RBF Kernel (SVC-RBF) and Gradient Boosting Decision Tree Classifier (GBC) (Cortes and Vapnik 1995; Friedman 2001). We used the handcrafted features, as described in Section 3.2, to train these models. As mentioned in Section 3.4, we use 10-fold cross validation during evaluation. Each fold was divided into train, validation and test. For each fold, we use the train set for training the models and the validation set for tuning the baseline model parameters.

We use the standard evaluation metrics for classification tasks - Precision, Recall and F1 score (Raghavan, Bollmann, and Jung 1989).

5 Results and Discussion

Table 3 shows the results of 10-fold cross validation on the evaluation dataset and compares the performance of DTM with other baselines approaches. Table 4 shows the top 10 informative features across SVM-L and GBC baseline approaches. Since, in a non-linear RBF kernel, it won't be possible to compute the weight vector explicitly, we can't directly get the feature importance for SVM-RBF. From the table, we could notice that IEF is the most informative feature across all baselines since it offers an important hint about the rarity (and hence unusualness) of a predicate. Similarly, there are other common features which were found important such as PF-IEF, Ent_Inlinks (Popularity) and Entropy of Category.

Based on F1 score, we can observe that both the variants of DTM - CNN and F-CNN perform better than the other hand-crafted feature based baselines. The combination of hand-crafted and automatically learnt features helps the

F-CNN in achieving a significant improvement in F1 score when compared to the baselines. The CNN based approach which relies on automatically learnt features alone is usually better in improving precision. It is interesting to observe that without the aid of any hand-crafted features, the CNN approach performs competitively with the best hand-crafted baselines with some minor improvements in overall F1 scores. In Bollywood Actors domain, DTM using F-CNN performs best and achieves a precision of 0.83 which is an improvement of 7.7% with respect to the nearest baseline result (given by GBC) and an F1 score of 0.81 which is an improvement of 12% with respect to the nearest baseline result (given by SVC-L). Similarly, in Music Artists domain, DTM using F-CNN performs best and achieves a precision of 0.72 which is an improvement of 4.3% with respect to the nearest baseline result (given by GBC) and an F1 score of 0.65 which is an improvement of 12% with respect to the nearest baseline result (given by SVC-L).

Table 5 shows some sample trivia mined from the datasets using both the variants of DTM. For example, from Bollywood Actors domain, DTM tags the following fact as interesting: "*Sanjay Dutt*" was the *founder* of the popular martial arts club called "*Super Fight League*" which was also found to be interesting by majority of our judges. Similarly, from Music Artists domain, DTM tags the following fact as interesting: "*Akon*" is the *owner* of the organization called "*Akon Lighting Africa*" which is also in agreement with the majority of our judges.

6 Conclusion

We proposed a novel approach called DBpedia Trivia Miner (DTM) for mining interesting trivia from structured KGs such as DBpedia. For a given domain, with the help of supervised data about interesting entity facts, DTM *learns* an Interestingness Model (IM) and uses it to identify more potentially interesting facts (trivia) for other entities in the same domain. We modeled IM as a binary classification problem and proposed two different variants for learning it - a) Con-

⁷<http://lrc.iit.ac.in/download.html>

volutional Neural Network (CNN) based approach and b) Fusion Based CNN which combined both hand-crafted features and automatically learnt CNN features. Experiments across two different domains - Bollywood Actors and Music Artists revealed that the CNN based approach automatically learns features which are relevant to the task and achieves competitive performance relative to the hand-crafted feature based baselines. Fusion Based CNN (F-CNN) significantly improves the performance over baseline approaches which use hand-crafted features alone. As part of future work, we would like to extend our work to include user feedback signals regarding interestingness so that the notion of interestingness could be gradually personalized to a given user.

References

- Akoglu, L.; Tong, H.; and Koutra, D. 2015. Graph Based Anomaly Detection and Description: A Survey. *Data Mining and Knowledge Discovery* 29(3):626–688.
- Attfield, S.; Kazai, G.; Lalmas, M.; and Piwowarski, B. 2011. Towards a Science of User Engagement. In *WSDM Workshop on User Modelling for Web Applications*.
- Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 2007. Dbpedia: A Nucleus for a Web of Open Data. In *The Semantic Web*. Springer. 722–735.
- Berners-Lee, T.; Hendler, J.; Lassila, O.; et al. 2001. The Semantic Web. *Scientific American* 284(5):28–37.
- Bizer, C.; Heath, T.; and Berners-Lee, T. 2009. Linked Data-The Story So Far. *Semantic Services, Interoperability and Web Applications: Emerging Concepts* 205–227.
- Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. P. 2011. Natural Language Processing (almost) from Scratch. *CoRR* abs/1103.0398.
- Cortes, C., and Vapnik, V. 1995. Support Vector Networks. *Machine Learning* 20(3):273–297.
- Dahl, G. E. 2015. *Deep Learning Approaches to Problems in Speech Recognition, Computational Chemistry, and Natural Language Text Processing*. Ph.D. Dissertation, University of Toronto.
- Fleiss, J. L. 1971. Measuring Nominal Scale Agreement Among Many Raters. *Psychological bulletin* 76(5):378.
- Friedman, J. H. 2001. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics* 1189–1232.
- Geng, L., and Hamilton, H. J. 2006. Interestingness Measures for Data Mining: A Survey. *ACM Computing Surveys (CSUR)* 38(3):9.
- Kontonasiou, K.-N.; Spyropoulou, E.; and De Bie, T. 2012. Knowledge Discovery Interestingness Measures Based On Unexpectedness. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2(5):386–399.
- Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, 1097–1105.
- Ma, S., and Huang, J. 2008. Penalized Feature Selection and Classification in Bioinformatics. *Briefings in Bioinformatics* 9(5):392–403.
- Mahesh, K., and Karanth, P. 2015. Smart-aleck: An Interestingness Algorithm for Large Semantic Datasets. *MOEKA-Memoranda in Ontological Engineering and Knowledge Analytics MOEKA. 2015.6 Recent Papers on Ontological Engineering* 103.
- McGarry, K. 2005. A Survey of Interestingness Measures for Knowledge Discovery. *The Knowledge Engineering Review* 20(1):39–61.
- Michael Gamon, Arjun Mukherjee, P. P. 2014. Predicting Interesting Things in Text. In *COLING '14. ACL*.
- Nair, V., and Hinton, G. E. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 807–814.
- O'Brien, H. L., and Toms, E. G. 2010. The Development and Evaluation of a Survey to Measure User Engagement. *Journal of the American Society for Information Science and Technology* 61(1):50–69.
- Perozzi, B.; Akoglu, L.; Iglesias Sánchez, P.; and Müller, E. 2014. Focused Clustering and Outlier Detection in Large Attributed Graphs. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1346–1355. ACM.
- Prakash, A.; Chinnakotla, M. K.; Patel, D.; and Garg, P. 2015. Did You Know? Mining Interesting Trivia for Entities from Wikipedia. In *Proceedings of the 24th International Conference on Artificial Intelligence*, 3164–3170. AAAI Press.
- Raghavan, V.; Bollmann, P.; and Jung, G. S. 1989. A Critical Investigation of Recall and Precision as Measures of Retrieval System Performance. *ACM Transactions on Information Systems (TOIS)* 7(3):205–229.
- Salton, G., and Buckley, C. 1988. Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing & Management* 24(5):513–523.
- Sparck Jones, K. 1972. A Statistical Interpretation of Term Specificity and its Application in Retrieval. *Journal of Documentation* 28(1):11–21.
- Suchanek, F. M.; Kasneci, G.; and Weikum, G. 2007. Yago: A Core of Semantic Knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, 697–706. ACM.
- Suggu, S. P.; Goutham, K. N.; Chinnakotla, M. K.; and Shrivastava, M. 2016a. Deep Feature Fusion Network for Answer Quality Prediction in Community Question Answering. In *SIGIR '16*. ACM.
- Suggu, S. P.; Goutham, K. N.; Chinnakotla, M. K.; and Shrivastava, M. 2016b. *Hand In Glove*: Deep Feature Fusion Network Architectures for Answer Quality Prediction in Community Question Answering. In *COLING '16. ACL*.
- Tang, J.; Alelyani, S.; and Liu, H. 2014. Feature Selection for Classification: A Review. *Data Classification: Algorithms and Applications* 37.
- Varela, F. J.; Thompson, E.; and Rosch, E. 1993. *The Embodied Mind: Cognitive Science and Human Experience*. Cambridge/MA: The MIT Press.
- Vrandečić, D., and Krötzsch, M. 2014. Wikidata: A Free Collaborative Knowledge Base. *Communications of the ACM* 57(10):78–85.
- Wizard, P. M. 2013. Trivia Game on Facebook.