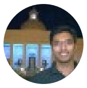
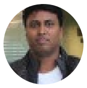
Did you know?-Mining Interesting Trivia for Entities from Wikipedia

4 AUTHORS, INCLUDING:



Abhay Prakash
Indian Institute of Technology Roorkee
3 PUBLICATIONS **0** CITATIONS



Manoj Chinnakotla
Microsoft
13 PUBLICATIONS **46** CITATIONS

Puneet Garg
Microsoft
1 PUBLICATION **0** CITATIONS

Did you know?- Mining Interesting Trivia for Entities from Wikipedia

Abhay Prakash[1], Manoj K. Chinnakotla[2], Dhaval Patel[1], Puneet Garg[2]
[1]Indian Institute of Technology, Roorkee, India
[2]Microsoft, India
abhayprakash@outlook.com, {manojc,puneetga}@microsoft.com, patelfec@iitr.ac.in

Abstract

Trivia is any fact about an entity which is *interesting* due to its *unusualness*, *uniqueness*, *unexpectedness* or *weirdness*. In this paper, we propose a novel approach for mining entity trivia from their Wikipedia pages. Given an entity, our system extracts relevant sentences from its Wikipedia page and produces a list of sentences ranked based on their interestingness as trivia. At the heart of our system lies an interestingness ranker which learns the notion of interestingness, through a rich set of domain-independent linguistic and entity based features. Our ranking model is trained by leveraging existing user-generated trivia data available on the Web instead of creating new labeled data. We evaluated our system on movies domain and observed that the system performs significantly better than the defined baselines. A thorough qualitative analysis of the results revealed that our rich set of features indeed help in surfacing interesting trivia in the top ranks.

1 Introduction

In the Internet age, where user attention span has become ephemeral, designing features and experiences which are not just usable but also *engaging*, has become the holy grail of all products and online applications. Failing to actively engage with the user may result in the user losing interest, getting distracted and finally abandoning or switching to a different application [O'Brien and Toms, 2008]. In view of this, researchers and practitioners have started designing product experiences which focus on the non-utilitarian aspect of the interaction which motivates the user to invest time, attention and emotion [Hassenzahl and Tractinsky, 2006; Attfield *et al.*, 2011]. For example, besides search results, popular search engines surfaced rich experiences[1] such as interactive maps, polls and other statistics for election related queries, during the U.S. Elections in 2014.

Trivia is any fact about an entity which is *interesting* due to its *unusualness*, *uniqueness*, *unexpectedness* or *weirdness*. For example, for the movie *"The Dark Knight (2008)"*, a trivia could be - *"To prepare for Joker's role, Heath Ledger lived alone in a hotel room for a month, formulating the character's posture, voice, and personality"*. The example sentence qualifies as a trivia as per our definition since it is unusual for an actor to seclude himself in a hotel for a month to just prepare for his role. Such kind of facts draw the user to engage more with the entity since it appeals to their sense of appreciating novelty, curiosity and inquisitiveness, thereby promoting repeated engagement [O'Brien and Toms, 2010; Attfield *et al.*, 2011]. A trivia could be presented either as a question-answer or as a single fact depending on the design of the experience and scenario. Although, business case studies [Voice Heard Media Inc., 2013; Pro Marketing Wizard, 2013] have shown that trivia helps in driving more user engagement, the current process of curating trivia involves significant manual effort. Editors curate trivia by searching for interesting facts about the entity from the Web and various other knowledge sources. This makes the entire process both expensive and hard to scale across a large number of entities. In an experiment, we tried collecting trivia for selected entities using professional trivia curators, and observed that the average throughput on a working day was around 50 trivia covering only 10 entities.

In this paper, we introduce the problem of *automatically mining trivia for entities from unstructured text* and propose a novel approach called *"Wikipedia Trivia Miner (WTM)"* to mine them from Wikipedia. Although interestingness is a subjective notion, which may differ from person to person, there are some facts for which there would be a significant agreement about their interestingness between a majority of people. We currently restrict ourselves to such a majority-based view of interestingness and leave the personalized subjective angle for future work. We choose Wikipedia as our knowledge source since factual correctness is an important attribute for trivia. Given an entity, WTM extracts relevant sentences from its Wikipedia page and orders them based on their interestingness using a machine-learning model. The final output of WTM is a list of top k sentences which are *interesting trivia* for those entities. In the current work, we demonstrate the effectiveness of WTM on movie entities. Instead of collecting labeled data for training the Interestingness Ranker, we harness publicly available user-generated trivia data from IMDB. Based on a rich set of entity and language analysis based features, WTM's Interestingness Ranker learns to give preference to sentences which could be interesting trivia over regular sentences, and shows significant improvement over baselines.

To summarize, the major contributions of this paper are:

- We introduce the problem of automatically mining interesting trivia for entities from Wikipedia.
- We propose a system to mine top k interesting trivia for any given entity based on their interestingness.

[1] `http://bit.ly/1njhlmh`

- For movie entities, we leverage already available user-generated trivia data from IMDB for learning a model of interestingness.

The rest of the paper is organized as follows. Section 2 presents our work in the context of related work. Section 3 describes the details of our system such as architecture, interestingness ranker and features used. Section 4 describes the experimental set-up, details of the evaluation datasets and evaluation metrics. Section 5 presents the quantitative and qualitative results. Finally, Section 6 concludes the paper.

2 Related Work

Earlier work in trivia mining focused on generating questions from structured databases. [Merzbacher, 2002] gave a functional approach to mine trivia questions in form of relational queries. The relational queries thus obtained, are finally decoded by humans to frame natural language questions. However, the work is constrained to the availability of structured databases in the target domain, as well as is limited to discovering trivia which can be represented as structured queries.

Recently, there is a noticeable trend in the text mining community to discover interesting items from unstructured text. For instance, [Gamon *et al.*, 2014] presented a technique to identify interesting anchors from Wikipedia pages. They model the interestingness of an anchor by utilizing the users′ browsing transitions within the Wikipedia domain, and formulate the problem as a click prediction task. However, their work is limited to anchors and there is no natural language understanding involved to discover interesting non-anchor text.

[Ganguly *et al.*, 2014] have tried to identify aesthetically pleasing (beautiful) sentences, using only positive samples obtained from Kindle *"popular highlights"*. In particular, the authors engineered various features, which they used in their proposed one-class classification algorithm. Some of the prominent features were *Topic Diversity*, *Sentiment*, *word repetition* and *Part of Speech (POS)* tags. Whereas for trivia mining, except POS tags, other features are not relevant e.g., a trivia usually has only one topic, sentiment of a trivia may be positive, negative or neutral, and repetition of words in trivia sentences is not usual.

The unusual, weird or surprising element of trivia could also be modeled using standard anomaly or outlier detection techniques. [Byrne and Hunter, 2004] make such an effort in which they try to identify unexpected or surprising news by identifying the violation of expectation, formed using background knowledge and domain facts. However, while mining trivia about an entity, background knowledge is not always obtainable and amenable to be modeled as expectation. e.g. consider the trivia *"The actors sang live on set"* (for movie *Jersey Boys (2014)*), which is an outlier and unexpected fact, but it is not feasible to model the notion that generally actors don't sing live on set. Moreover, the approach assumes that the input knowledge exists in structured format (XML) such as <*action*> *takeoverBid* <*/action*>. However, such rich structured data may not be available for all the facets of entities (such as actors singing live on sets). Hence, we chose unstructured text as the source for trivia where the above approach can't be used.

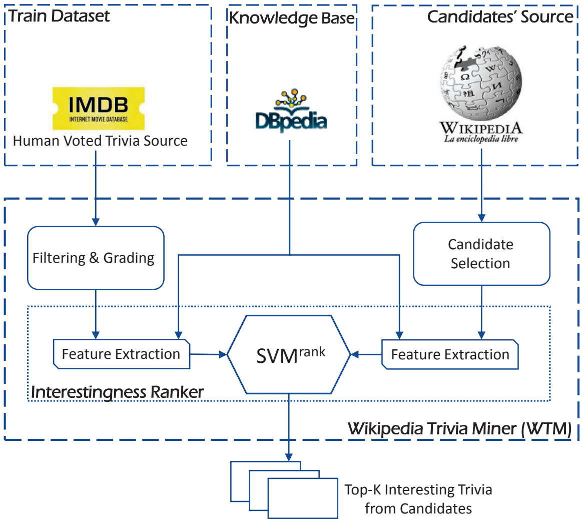


Figure 1: System Architecture of Wikipedia Trivia Miner (WTM)

To the best of our knowledge, we are the first one to introduce the problem and propose a machine-learning based approach for mining trivia from unstructured text using domain-independent linguistic and entity based features.

3 Wikipedia Trivia Miner

In this section, we describe the details of our system.

3.1 System Architecture

Figure 1 depicts the architecture of our entire system, in which our primary contribution lies in designing *"Wikipedia Trivia Miner (WTM)"*. WTM takes trivia from reliable sources to train a ranking model and applies it on Wikipedia documents of target entity to mine interesting trivia. In this paper, we demonstrate WTM for movie segments, where we used user-generated trivia, available at IMDB, as training dataset. The IMDB trivia data also has a voting related interestingness measure in the form of *"X of Y found this interesting"*, where *X* is the positive votes and *Y* is the total votes for the trivia. The WTM Filtering & Grading module filters and labels this data into grades, based on their interestingness votes, as required to train our Interestingness Ranker.

For any given movie entity, its Wikipedia page covers different aspects of the target entity, using more than one adjoining sentences. But, not all of them could be understood independently without requiring appropriate context. For example, a sentence like *"It really reminds me of my childhood."* (from the movie *Let Me In (2010)*), is out of context and can't be understood independently. The Candidate Selection (CS) module selects sentences which are independently comprehensible. The Interestingness Ranker takes these candidates as input and ranks them in the order of interestingness. In the following sub-sections, we discuss each of the mentioned modules in more detail.

3.2 Filtering & Grading

In this module, we prepare the training dataset for learning interestingness, by classifying each trivia into one of the

Grade	No. of Trivia	Sample Trivia	Movie Name	LR
4 (Very Interesting)	706	Luc Besson wrote the original screenplay when he was in high school.	The Fifth Element (1997)	1.00
3 (Interesting)	1091	Tom Cruise did all of his own stunt driving.	Jack Reacher (2012)	0.98
2 (Ambiguous)	2880	Emily Blunt's character is named Rita, a possible nod to the love interest Rita from Groundhog Day (1993).	Edge of Tomorrow (2014)	0.75
1 (Boring)	945	Andrea Riseborough was considered to play the role of Henley.	Now You See Me (2013)	0.27
0 (Very Boring)	541	The first time Portia Doubleday and Rooney Mara are in the same movie since Youth in Revolt (2009).	Her (2013)	0.20

Table 1: Sample trivia from each grade of IMDB data after the Filtering and Grading step.

five grades (ranging from 'Very Interesting' to 'Very Boring'). We selected 5000 most popular movie entities from Microsoft's internal Knowledge Graph, and crawled trivia for those entities from IMDB. In total, we obtain 99185 trivia along with their interestingness votes data. Using the votes data, we calculate Likeness Ratio (LR) for each trivia as $LR = \frac{No.\ of\ Interesting\ Votes}{Total\ Votes}$. Since, LR would be unreliable when computed only on a few total votes, we only consider trivia which have at least 5 total votes (minimum support). We observed that the distribution of trivia LR was highly skewed and followed a power law which is in agreement with earlier observations [Kostakos, 2009]. For instance, trivia with an LR of 1 were around 39.5% of the total. Due to this, we changed the minimum support only for higher LR ranges (greater than 0.6) to be 100 votes. We sort the remaining trivia based on their LR and assign grades to each of them by defining percentile cut-offs. The percentile cut-offs were placed at 90 (Very Interesting), 90-75 (Interesting), 75-25 (Ambiguous), 25-10 (Boring) and 10 (Very Boring). As a result of the above transformation, we ended up with 6163 trivia spanning across 846 movies. Table 1 gives the detailed statistics for each grade along with sample trivia. Note that there is a marked difference in the degree of interestingness across each grade which also gets reflected by the respective LR obtained.

3.3 Candidate Selection

As a first step in Candidate Selection (CS), we pre-process the target entity's Wikipedia page and extract the sentences contained in paragraph HTML elements. We ignore the text present in other environments such as infobox, tables, images, categories, references, links and itemized lists. We denote the resultant text as *"Core Content Text (CCT)"*. We use only CCT in all the downstream processing modules.

CCT, which is a set of sentences, also includes out of context sentences, as discussed in second paragraph of Section 3.1. To deal with this, there could be two solutions either provide context by including adjoining sentences or just drop such sentences for further processing. In the current work, we focus on mining trivia which are independently comprehensible and hence chose the alternative of dropping out-of-context sentences.

Given the CCT of a target entity, we use Sentence Detector [Manning *et al.*, 2014] to identify individual sentences. Next, we use Co-Reference Resolution [Manning *et al.*, 2014] to find out links between sentences in a given paragraph, and remove those sentences which have mentions outside the current sentence. However, sentences which refer to the target entity are retained. For instance, for the movie *"Forrest Gump (1994)"*, the following are sample sentences from Wikipedia *"Hanks revealed that he signed onto the film after an hour and a half of reading the script. He initially wanted to ease Forrest's pronounced Southern accent, but was eventually persuaded by director Bob Zemeckis to portray the heavy accent stressed in the novel."*. The first sentence has an out link as "the film" which refers to the target entity. The second sentence refers to the hero "Tom Hanks" as "He". As stated earlier, in CS phase, we drop the second sentence and retain the first one.

3.4 Interestingness Ranker

The objective of the Interestingness Ranker (IR) is to rank candidate sentences in the decreasing order of their interestingness as trivia. We use the Rank SVM [Joachims, 2006] based formulation to automatically *learn* the ordering function from training data. This enables WTM to adapt to a new domain of entities by just changing the training data accordingly. The training input to IR is the graded interestingness data, in the form of (Movie, Trivia, Grade), as prepared in Section 3.2. The featurization step converts this triple into a - (Movie, Features, Grade) where each trivia is transformed into a feature vector. During training, the Rank SVM model learns the feature weights which will result in the best ranking performance at rank 10 measured using NDCG@10 [Järvelin and Kekäläinen, 2002]. At test time, the feature weights are used to compute a score which will be used to order the trivia. The features extracted by the featurizer could be divided into three classes: Unigram, Linguistic and Entity based.

Unigram Features

Using unigram features, we try to identify important words which make the trivia interesting e.g., words like *"improvise"*, *"award"* etc. might bring interestingness to the trivia for an entity from movie domain. For example, in Table 1, the trivia from movie Jack Reacher (2012) is interesting because of the word *"stunt"*. We do some basic pre-processing before computing features: case conversion, stemming, stop word removal and punctuation symbol removal. We use TF-IDF weight of each unigram as feature.

Feature Type	No. of Features	NDCG@10
Unigram (U)	18025	0.934
Linguistic (L)	5	0.919
Entity Features (E)	4686	0.929
U + L	18029	0.942
U + E	22711	0.944
WTM (U + L + E)	22716	**0.951**

Table 2: Results of five-fold cross validation with the best model parameters. Also, shows the incremental contribution by each feature group.

Linguistic Features

Mere unigram features are not enough to capture the semantics of sentences. Hence, we perform deeper language analysis on the trivia - POS tagging and dependency parsing to extract the following five different types of language-oriented features:

- Superlative Words: Words of superlative degree such as first, best, longest, shortest etc. express the extremeness or uniqueness of the entity attribute in focus and could be interesting. We use the Stanford Core-NLP POS tagger [Toutanova *et al.*, 2003] to detect the presence of superlative adjectives (JJS) and superlative adverbs (RBS) in the sentence and fire a binary feature based on its presence.
- Contradictory Words: Presence of contradictory words suggests the presence of opposing ideas which could spark intrigue and interest. We used a list of such words online[2]. Some of these words are *"but, although, unlike"* etc. We fire a binary feature based on its presence. An example trivia with such words is *"Although a very modest hit in theaters, it became one of the highest grossing video rentals of all time."*
- Root Word of Sentence: We use the Stanford Dependency Parser [Chen and Manning, 2014] to obtain dependency parse of the sentence. The root word from a dependency parse helps in capturing the core activity being discussed in the sentence. For example, in the sentence *"Gravity grossed $274,092,705 in North America."*, with *gross* as the root word, we can infer that the sentence is talking about some revenue related stuff. We lemmatize the verb at root and mark its presence in the form of a boolean feature *root_X*, where X is the lemmatized form of the word.
- Subject of Sentence: We extract the subject of the sentence from the dependency parse as *subj_X*, where X is the lemmatized form of the word.
- Readability Score: Complex and lengthy trivia are hardly interesting. Hence, we use Fog Index as a feature - which is a measure of readability [Gunning, 1969]

Entity Features

In order to learn entity and attribute-level generalizations in the model, we include named entities (using Stanford NER [Manning *et al.*, 2014]) and entity-linking features. For example, from the trivia *"De Niro was so anxious, he didn't attend Oscars"*, we would like to learn that - *Entity.Actor not attending Oscars may be interesting*. We include the following features:

Dataset Name	Source	No. of Sent.	No. of Movies	No. of Pos. Sent. (Trivia)
Train Set	IMDB	6163	846	-
Test Set	Wikipedia	2928	20	791

Table 3: Dataset Details

- Presence of generic Named Entities(NEs): Presence of NEs like MONEY, ORGANIZATION etc.
- Entity-Linking Features: We link NEs to entity attributes using knowledge from the DBPedia knowledge base [Auer *et al.*, 2007]. For example, if a trivia from movie *"The Fifth Element (1997)"* contains NE *Luc Besson*, then it is linked to *entity_Director* as well as *entity_Writer*.
- Focus NE of Sentence: For this feature, we resolve any NE present directly under the root. We mark its presence by a feature such as *underRoot_entity_Director*.

Note that all the Entity Features are *generic* and not domain-specific as while resolving the entities, we just lookup the attribute-value pair from DBPedia. If any NE matches with the value in attribute-value pair of entity, we replace the NE with *entity_Attribute*. For example, in case of country domain, while processing a sentence about the USA, all the occurrences of president's name, will be replaced by *entity_President*. If the word is a NE, but still not resolved then, it is tagged as a feature *entity_unlinked_(NamedEntityType)* like *entity_unlinked_PERSON*.

To summarize, all our current features are generated *automatically* through: linking entity attributes using knowledge-base (DBPedia) and language analysis (Parsing, POS-tagging, NER etc.) on target sentence. In case of a new domain like Celebrities, the entity linking phase will automatically generate a different subset of domain-specific features such as entity_birthPlace, entity_Spouse etc. Based on the celebrity domain training data, the Rank SVM model may assign higher weights to a different set of features than the ones in movie domain.

4 Experimental Setup

In this section, we describe the details of our evaluation dataset, evaluation metrics and experimental set-up.

We used the SVM^{rank} package [Joachims, 2006] for implementing the interestingness ranker. We tuned the kernel, model parameters C and e using five-fold cross validation and rest of the values were set to default. The best parameters were found to be a linear kernel with $C = 17$ and $e = 0.21$. Table 2 shows the final results of the five-fold cross validation with the best parameters of the model.

4.1 Test Dataset Creation

In order to evaluate the effectiveness of our system, we created a dataset of sentences from the Wikipedia pages of movie entities. As shown in Table 3, we downloaded the Wikipedia pages for 20 movies randomly sampled from the top 5000

[2] http://bit.ly/1kOMshx

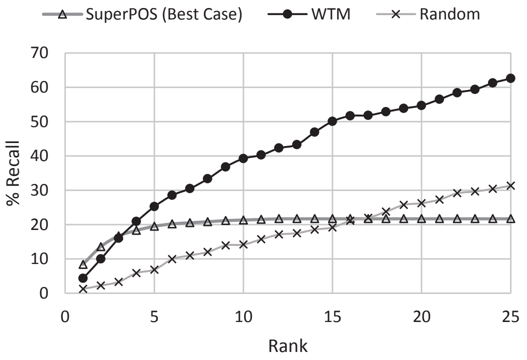


Figure 2: Comparison of Recall at various ranks of WTM with baselines on Wikipedia Test Set.

popular movies of IMDB. As described in Section 3.3, we pre-processed and extracted the CCT for all the movies. Later, we divided them into sentences and obtained crowd-source judgments on their interestingness. In order to avoid subjective bias, each movie-sentence pair was judged by five judges on a two-point scale (Interesting, Boring). The judges were given detailed guidelines along with sample judgments. The consensus (majority judgment) was then marked as the final label for the trivia. In our entire experiments, to be fair, the input to all the approaches (Baselines and WTM) was given as pre-processed Wikipedia text (CCT).

Since interestingness is a subjective notion, getting as many judgments as possible would be ideal for making reliable conclusion. However, we chose the number of judges as five due to the following reasons. Each Wikipedia page has 100 sentences on an average. So, due to practical budgetary limitations, we could not go beyond the current number of judgments. We performed an experiment to validate the extent to which our crowd-sourced judgments with five judgments per trivia match the general wisdom of the crowd. Since the IMDB training data has many votes for each trivia, we randomly sampled 100 trivia from it and labeled those with Likeliness Ratio (LR) greater than 0.5 as class 1 and LR less than 0.5 as class 0. This is essentially a majority vote on the total votes polled. We call these labels as IMDB vote based labels. Later, we got 5 judgments for each of these 100 trivia through our crowd-sourcing platform and assigned labels based on majority vote. We call these labels as Crowd-Source labels. We calculated the agreement between these two labeling mechanisms (IMDB vote based labels and Crowd-Source labels) using the Kappa Statistic [Viera and Garrett, 2005]. We found the Kappa value to be 0.618 which means the agreement is “substantial”. This shows that five judgments, although not ideal, are sufficient enough to adequately reflect the general wisdom of the crowd.

4.2 Evaluation Metrics

Given a Wikipedia page for an entity, our system produces a ranked list of top *‘k’* most interesting trivia sentences. Hence, we use $Precision@k$ as our evaluation metric. However, to demonstrate the effectiveness of our system in terms of its ability to bring in diverse kinds of trivia, we also report $Recall@k$. Since we learn a ranking model on the graded IMDB training data, we report $NDCG@k$ for all our ranking model building and feature evaluation experiments.

Model		**Avg. P@10**	**% Improv. (Baseline I)**	**% Improv. (Baseline II)**
Random Pick from Wikipedia Page **(Baseline I)**		0.25	-	-
After Cand. Selection (CS)	Random Picked	0.30	**19.61**	-
CS + Rank by	# of Sup. POS (Worst Case)	0.32	**27.45**	-
	# of Sup. POS (Random)	0.33	**29.41**	-
	# of Sup. POS (Best Case) **(Baseline II)**	0.34	**33.33**‡	-
	WTM with (U) Features	0.34	**33.33**	0
	WTM with (U+L+E) Features	0.45	**78.43**‡	**33.82**‡

Table 4: Results comparing the performance of WTM approach over Random and Sup. POS baselines on Wikipedia Test Set. Results marked as ‡ indicate that improvement was statistically significant at 95% confidence level ($\alpha = 0.05$) when tested using a paired two-tailed t-test.

4.3 Baseline Approaches

We define two baselines against which we will compare the performance of our system. The first baseline is a “random” which means randomly picking any sentence from the Wikipedia page of the entity. Also, superlative words such as *best, most, highest, largest* are often used in many of the interesting trivia. For example, for the movie *“The Matrix (1999)”*, the following sentence is a trivia - *“In 2007, Entertainment Weekly called The Matrix the best science-fiction piece of media for the past 25 years.”*. We define the second baseline which ranks sentences based on the number of superlative words in them. To be uniform in comparison, we consider sentences obtained by candidate selection. We use the Stanford POS Tagger [Toutanova *et al.*, 2003] for identifying these superlative words. If sentences have the same number of superlative words - we report the precision corresponding to the best case (i.e., where we get maximum precision).

5 Results and Discussion

Table 4 shows the overall results of our system in comparison with the baseline approaches. In Precision@10 (P@10) metric, WTM system performs significantly better than both the baselines with an improvement of 78.43% over Random and 33.82% over Superlative POS (Best) techniques. Figure 2 compares recall of WTM with other baselines. Although the Superlative POS fares well initially vis-a-vis recall, it reaches saturation very soon. This is due to the fact that Superlative POS baseline lacks diversity and can only retrieve a single type of trivia - those which contain superlative descriptions.

Result	Movie	Trivia	Description
WTM Wins (Sup. POS Misses)	Interstellar (2014)	Paramount is providing a virtual reality walkthrough of the Endurance spacecraft using Oculus Rift technology.	Due to Organization being subject, and (U) features (technology, reality, virtual)
	Gravity (2013)	When the script was finalized, Cuarón assumed it would take about a year to complete the film, but it took four and a half years.	Due to Entity.Director, Subject (the script), Root word (assume) and (U) features (film, years)
	The Deer Hunter (1978)	De Niro was so anxious that he did not attend the Oscars ceremony.	WTM gets it right due to Entity.Actor, Root word(anxious) and (U) features (oscars, ceremony)
WTM’s Bad	Elf (2003)	Stop motion animation was also used.	Candidate Selection failed
	Rio 2 (2014)	Rio 2 received mixed reviews from critics.	Root verb "receive" has high weightage in model
Sup. POS Wins (WTM misses)	The Incredibles (2004)	Humans are widely considered to be the most difficult thing to execute in animation.	Presence of ‘most’, absence of any Entity, vague Root word (consider)
Sup. POS's Bad	Lone Survivor (2013)	Most critics praised Berg's direction, as well as the acting, story, visuals and battle sequences.	Here 'most' is not to show degree but instead to show genericity.

Table 5: Qualitative comparison of trivia mined using WTM and Sup. POS techniques. This table presents representative samples of both interesting and non-interesting trivia mined by each technique along with an explanation.

Rank	Feature	Group	Weight
1	subj_scene	Linguistic	0.327141
2	subj_entity_cast	Linguistic + Entity	0.305082
3	entity_produced_by	Entity	0.225021
4	underroot_unlinked_organization	Linguistic + Entity	0.215818
6	root_improvise	Linguistic	0.194474
7	entity_character	Entity	0.190772
8	MONEY	Entity (NER)	0.188747
14	stunt	Unigram	0.162217
16	superPOS	Linguistic	0.158272
17	subj_actor	Linguistic	0.157705

Table 6: Most informative features (by weight) from Interestingness Ranker Model along with their Feature Group.

On the other hand, besides superlatives, WTM can retrieve a variety of other trivia due to its rich feature based on language analysis and entity understanding. At rank 25, WTM has an average recall of around 60%. This is indeed a valuable proposition which enables mining of entity trivia at scale.

5.1 Feature Contribution

Table 2 shows the contribution of each feature class in improving the model accuracy. The Entity Based Features such as Entity Attributes (Entity.Director, Entity.Producer, Entity.Actor), Named Entities (MONEY, ORGANIZATION etc.) result in the highest improvement followed by language analysis based features such as contradictions, root words and subject of dependency parse. The combined feature rich model outperforms all individual models. Table 6 shows few of top 20 most weighted features of our interestingness ranking model. It gives a peek into the trivia interestingness related hints learnt by our model. These correlate well with our general observations regarding movie trivia. For example, subject of a sentence being “scene”, main verb in the sentence is about “improvise”, words like “flop”, “real”, presence of entities such as Entity.Producer, Entity.Character etc.

5.2 Effect of Candidate Selection

Table 7 shows the effect of adding CS module on WTM accuracy for both uni-gram and the final WTM models. CS improves the precision of the final WTM model by more than 16%. These results prove that CS is indeed helping in eliminating out-of-context sentences. Human annotation was done before CS and we missed 51% of trivia candidates during CS. However, although human annotators marked them as interesting, most of the dropped sentences required context of previous sentences and paragraph.

5.3 Qualitative Comparison

Table 5 shows the qualitative comparison of trivia mined from WTM and Superlative POS approaches. We discuss a few examples from this table here. The first section, WTM Wins (Sup. POS Misses), represents those samples where WTM retrieved the interesting trivia in top-10 and the baseline Sup. POS completely missed it. In the second example of this section, from *Gravity (2013)*, WTM gets the trivia correctly due to: a) resolving the NE *Cuaron* to Entity.Director b) getting the subject of the sentence as *the script*, c) important unigram features such as *film* and *years*, and finally d) the ranker understanding that this combination of features is interesting. Sup. POS doesn’t have any such sophisticated features and hence totally misses it. The second section of table, WTM’s Bad, shows some weaknesses of the model due to which non-interesting trivia show up. The first example shows a failure of CS that allows an out-of-context sentence. The second example shows a case where the trivia was ranked important due to the root verb *”receive”* getting undue importance.

There are some examples where the baseline Sup. POS got some good trivia which WTM missed (top-10). The third section presents an example from *The Incredibles (2004)* which consists of a superlative word *most*. However, WTM could not get it due to absence of any prominent entity (no entities), language (vague root word: consider) or unigram related features. The fourth section presents an example from

Model	P@10 before CS	P@10 after CS	% Improv.
WTM (U)	0.28	0.34	21.43
WTM (U+L+E)	0.39	0.45	16.67

Table 7: Effect of CS on WTM Precision. WTM (U) is only Unigram and WTM (U+L+E) is final system.

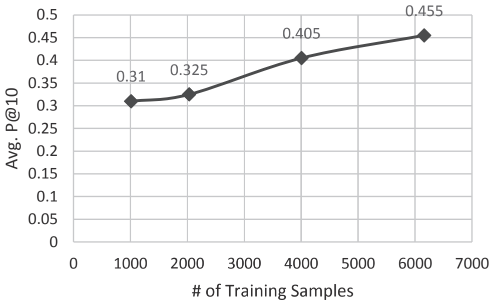


Figure 3: Training Data Size vs. WTM Accuracy

Lone Survivor (2013) where Sup. POS fails and retrieves a non-interesting trivia. This is due to misclassifying the word *Most* occurring at the beginning of the sentence as superlative.

5.4 Sensitivity to Training Size

As given in Table 3, in our current system, we use around 6K trivia training samples filtered from around 846 movies. However, we also studied the effect of varying training data size on the precision of our system and report it in Figure 3. Results show that the precision of WTM increases with the size of training data. This is a desirable property as it allows us to further improve the precision of WTM by including more training data (for instance, by expanding the initial movies list beyond 5K).

6 Conclusion and Future Work

We introduced the problem of automatically mining trivia for entities from the unstructured Wikipedia text. We proposed a novel system called *"Wikipedia Trivia Miner (WTM)"* for tackling the problem and demonstrated its performance on movie entities. Experiments on movie entities and Wikipedia dataset reveal that the proposed system performs significantly better than the baseline approaches and indeed succeeds in surfacing interesting trivia sentences for the entity in focus. The success of the system could be mainly attributed to the sophisticated domain-independent ranking features which are based on language analysis and entity understanding. As part of future work, we would like to extend our current work to Celebrities domain and investigate on unsupervised methods for entity trivia mining.

References

[Attfield *et al.*, 2011] Simon Attfield, Gabriella Kazai, Mounia Lalmas, and Benjamin Piwowarski. Towards a Science of User Engagement. In *WSDM Workshop on User Modelling for Web Applications, 2011*. ACM.

[Auer *et al.*, 2007] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, and Zachary Ives. Dbpedia: A Nucleus for a Web of Open Data. In *In 6th Intl Semantic Web Conference, Busan, Korea*, pages 11–15. Springer, 2007.

[Byrne and Hunter, 2004] Emma Byrne and Anthony Hunter. Man Bites Dog: Looking for Interesting Inconsistencies in Structured News Reports. *Data and Knowledge Engineering*, 48(3):265 – 295, 2004.

[Chen and Manning, 2014] Danqi Chen and Christopher Manning. A Fast and Accurate Dependency Parser using Neural Networks. In *EMNLP 2014*, pages 740–750. ACL, 2014.

[Gamon *et al.*, 2014] Michael Gamon, Arjun Mukherjee, and Patrick Pantel. Predicting Interesting Things in Text. In *COLING 2014*, pages 1477–1488. Dublin City University and ACL.

[Ganguly *et al.*, 2014] Debasis Ganguly, Johannes Leveling, and Gareth Jones. Automatic Prediction of Text Aesthetics and Interestingness. In *COLING 2014*. Dublin City University and ACL.

[Gunning, 1969] Robert Gunning. The Fog Index after Twenty Years. *Journal of Business Communication*, 6(2):3–13, 1969.

[Hassenzahl and Tractinsky, 2006] Marc Hassenzahl and Noam Tractinsky. User Experience - A Research Agenda. *Behaviour and Information Technology*, 25(2):91–97, 2006.

[Järvelin and Kekäläinen, 2002] Kalervo Järvelin and Jaana Kekäläinen. Cumulated Gain-Based Evaluation of IR Techniques. *ACM TOIS*, 20(4):422–446, 2002.

[Joachims, 2006] Thorsten Joachims. Training Linear SVMs in Linear Time. In *KDD '06*, pages 217–226, 2006. ACM.

[Kostakos, 2009] Vassilis Kostakos. Is the Crowd's Wisdom Biased? A Quantitative Asessment of Three Online Communities. *arXiv preprint arXiv:0909.0237*, 2009.

[Manning *et al.*, 2014] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP Natural Language Processing Toolkit. In *ACL 2014: System Demonstrations*, pages 55–60.

[Merzbacher, 2002] Matthew Merzbacher. Automatic Generation of Trivia Questions. In Mohand-Sad Hacid, ZbigniewW. Ra, DjamelA. Zighed, and Yves Kodratoff, editors, *Foundations of Intelligent Systems*, vol. 2366 of *LNCS*, pages 123–130. Springer Berlin Heidelberg, 2002.

[O'Brien and Toms, 2008] Heather L. O'Brien and Elaine G. Toms. What is User Engagement? A Conceptual Framework for Defining User Engagement with Technology. *Journal of the American Society for Information Science and Technology*, 59(6):938–955.

[O'Brien and Toms, 2010] Heather L. O'Brien and Elaine G. Toms. 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, 2010.

[Pro Marketing Wizard, 2013] Pro Marketing Wizard. Trivia Game on Facebook. `http://bit.ly/1HUhlRx`, 2013.

[Toutanova *et al.*, 2003] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *NAACL '03*, pages 173–180, Stroudsburg, PA, USA, 2003. ACL.

[Viera and Garrett, 2005] A. J. Viera and J. M. Garrett. Understanding Interobserver Agreement: The Kappa Statistic. *Fam Med*, 37(5):360–363, 2005.

[Voice Heard Media Inc., 2013] Voice Heard Media Inc. Using Trivia and Quiz Products to Engage your Customer. `http://bit.ly/1Htiv5V`, 2013.