



# **MACHINE-LEARNING EXTRACTS AND SEMANTIC GRAPHS: CREATING STRUCTURED DATA TO DRIVE SEARCH, RECOMMENDATIONS AND DISCOVERY**

Lijin Chungapalli, Venkata Babji Perambattu

TiVo, India

## **ABSTRACT**

Effective entertainment discovery solutions require a deeper understanding of content, and one approach to harnessing this knowledge is extracting semantically-relevant metadata. This paper explains how to use a combination of semantic graphs and machine learning to automatically generate structured data, recognise important entities/keywords and create weighted connections for more relevant search results and recommendations. For example, the movie *The Big Short* can automatically produce entities, such as “hedge fund” and “subprime lending,” which are thematically relevant and therefore given a high weight. By inferring relevant entities through these underlying technologies, metadata results are richer and more meaningful, enabling faster decision-making for the consumer and stronger viewership for the content owner.

## **INTRODUCTION**

Today’s consumers have the advantage of choice – but from an ocean of content, including movies, programmes, news and short-form video from an array of linear and streaming services. Because there is so much content, largely lacking structured metadata, viewers are frustrated – they can’t find what they want to watch quickly and easily. Moreover, a 2016 consumer research study by ‘TiVo (1)’ identified a phenomenon called “show-dumping,” where consumers simply give up on programmes due to the challenges involved in accessing them. Show-dumping leaves content owners with a big problem: they heavily invest in producing excellent content, yet struggle to ensure consumers can find it.

A deeper understanding of content is required to create intelligent solutions that can overcome the challenges faced by consumers and content owners alike. Using traditional statistics-driven models for entity extraction will not solve the problem, as they lack semantic understanding. Combining machine-learning methods and semantic graphs is a unique way to add much-needed context and can alleviate consumer frustration, as well as strengthen viewership for content owners.

Historically, semantic graphs have helped a great deal in question-answering ‘Dali et al (2)’ and text summarisation ‘Moawrd and Ared (3)’. In this paper, we delve into ways to leverage the importance of the nodes in a semantic graph to train a machine-learning

model that will automatically determine the relevance of an entity in a given blurb of text, thus serving up better results for consumers to find what they want to watch.

## DATASET

We took the top 10,000 movies (based on popularity) from English Wikipedia, extracted candidates for entities/keywords from the movie plots, and manually verified them to create positive (all accepts) and negative (all rejects) labels in the dataset. The candidates from Wikipedia’s movie article page are:

1. Wiki links in plot section
2. Wiki links from synopsis
3. Wiki categories referenced in plot
4. Noun chunks from plot

We split the dataset into training and test sets in the ratio of 70:30. The training set was used to build the model, and the test set was evaluated and used for benchmarking. The details of how the dataset was used to build the machine-learning model are explained in the next section.

## ARCHITECTURE

Our objective is to take any blurb of text as input and convert it into a semantic graph that identifies key entities and their associations. The features from the semantic graph and the text blurb flow through the machine-learning model to infer the most contextually important entities.

Our process involves four stages:

- Pronoun Resolution
- Candidate Identification
- Creation of Semantic Graph
- Node Score

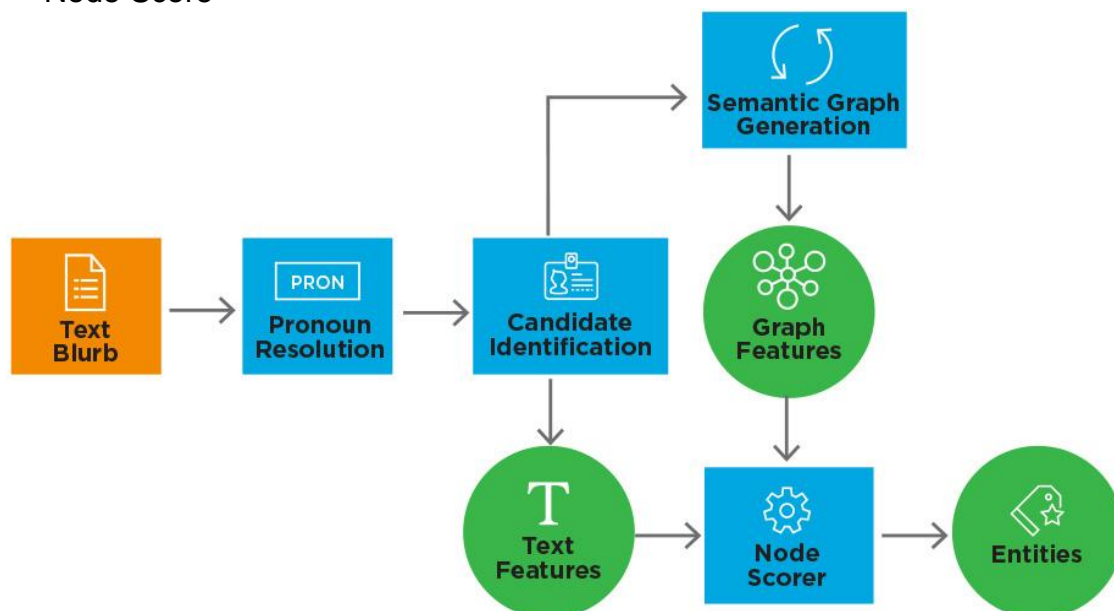


Figure 1 – Architecture

## Pronoun Resolution

Pronoun resolution is crucial for identifying the entity relationships necessary to rich, accurate semantic graphs. In this step of the process, we resolve all the pronouns across sentences in the text blurb by using a Python implementation of end-to-end Neural Coreference Resolution ‘Lee et al (4)’. This action helps determine the noun or proper noun to which the pronoun refers.

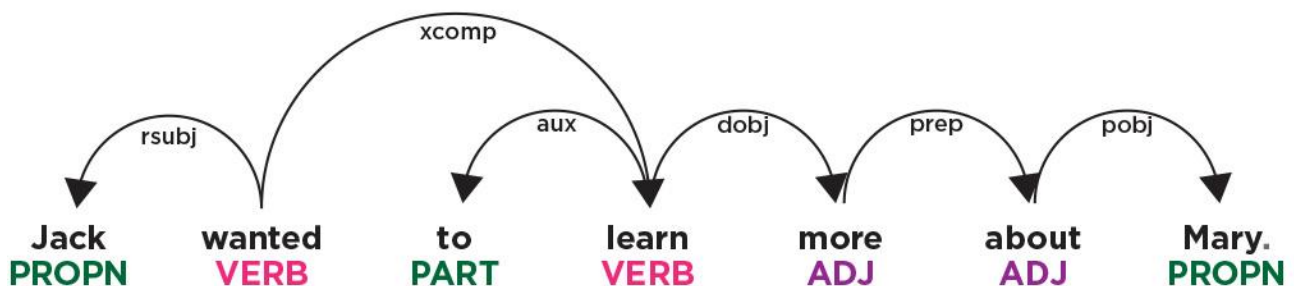
Example: *John* helped Mary. *He* is a doctor.  
By resolving pronouns, we end up with:  
*John* helped Mary. *John* is a doctor.

## Candidate Identification

By applying POS (Part-Of-Speech) tagging on the processed text, we identify all noun chunks as nodes in the semantic graph. SpaCy, a Python library for advanced Natural Language Processing, powers identification through its POS tagging ability. When using Wikipedia, we leverage its rich structure to identify more candidates like links from plot, synopsis and category mentions.

## Creation of Semantic Graph

For each of the candidates appearing in the sentence, we check whether they are connected by traversing the dependency tree, which is created using spaCy. Most connections are via verbs, and an undirected graph is created using these edges.



**Figure 2 – Creation of Semantic Graph**

In Figure 2, “Jack” and “Mary” are connected by the verbs “wanted” and “learn.”

## Node Scorer

Identifying key features is critical to any machine-learning model. In this model, we have two sets of features: text features and graph features.

Text features:

1. POS tag of the candidate we extracted using spaCy
2. TF-IDF (Term Frequency-Inverse Document Frequency) value of the candidate calculated over the plot of the top 10,000 movies in Wikipedia
3. Capitalisation of the candidate in the text blurb.

4. Whether the candidate has a link to another Wikipedia page in the Wikipedia movie plot (set to false for non-Wiki articles)
5. Whether the candidate is mentioned as a Wikipedia category for the movie (set to false for non-Wiki articles)
6. Whether the candidate is mentioned in Wikipedia first paragraph (set to false for non-Wiki articles)
7. Wikipage type of candidate; the type of page is tagged using Wikipedia first line and Wikipedia categories into seven types – programmes, people, fictional, place, organisation, sports and phrase (default type is phrase for any candidate)

### Graph features

We utilised two best centralities that reflect how important a node is with respect to the graph.

Closeness centrality 'Freeman (5)'

Betweenness centrality 'Freeman (6)'

### Closeness centrality

In a connected graph, closeness centrality (or closeness) of a node measures centrality in a network, calculated as the sum of the length of the shortest paths between the node and all other nodes in the graph. Thus, the more central a node is, the closer it is to all other nodes. The closeness centrality of a node  $C(x)$  is denoted by

$$C(x) = \frac{N}{\sum_y d(y, x)}$$

Where  $d(y, x)$  is the distance between vertex  $x$  and  $y$  and  $N$  is the number of nodes.

### Betweenness centrality

In graph theory, "betweenness" centrality is a measure of centrality in a graph based on shortest paths. For every pair of vertices in a connected graph, there exists at least one shortest path between the vertices, such that either the number of edges that the path passes through (for unweighted graphs) or the sum of the weights of the edges (for weighted graphs) is minimised. The betweenness centrality for each vertex is the number of these shortest paths that pass through the vertex. Betweenness centrality  $g(v)$  is denoted by

$$g(v) = \sum_{s, t \in V} \frac{\sigma(s, t/v)}{\sigma(s, t)}$$

Where  $V$  is the set of nodes,  $\sigma(s, t)$  is the number of shortest  $(s, t)$ -paths, and  $\sigma(s, t/v)$  is the number of those paths passing through some node  $v$  other than  $s, t$ . where if  $s = t$ ,  $\sigma(s, t) = 1$ , and if  $v \notin s, t$ ,  $\sigma(s, t/v) = 0$

If there are many connected components, we compute these features on each connected component separately.

We take all nine (seven text features and two graph features) of the features listed above, normalise them, train a classifier over the manually-curated data and use this model to

predict entities. We evaluated both the Decision Tree Classifier and Random Forest Classifier as they work well with categorical data.

Among the classifiers, the Decision Tree Classifier performed the best.

### EVALUATION MEASURE

We measure the precision and recall of the model by comparing our results with a manually-curated list of entities for movies.

We define precision as the proportion of the number of machine-generated entities that match the manually-curated list(N) to the total number of machine-generated entities(K).

$$precision = \frac{N}{K}$$

Recall is measured as the proportion of manually-curated entities that are extracted by the model(N) to the number of manually-curated entities(M).

$$recall = \frac{N}{M}$$

Many previous studies have used these metrics to calculate precision and recall for entities extraction 'Kerner (7)'.

### RESULTS

We tested the model with the test split of our manually-curated list of the top 10,000 movies. We ran the Decision Tree Classifier with and without graph features.

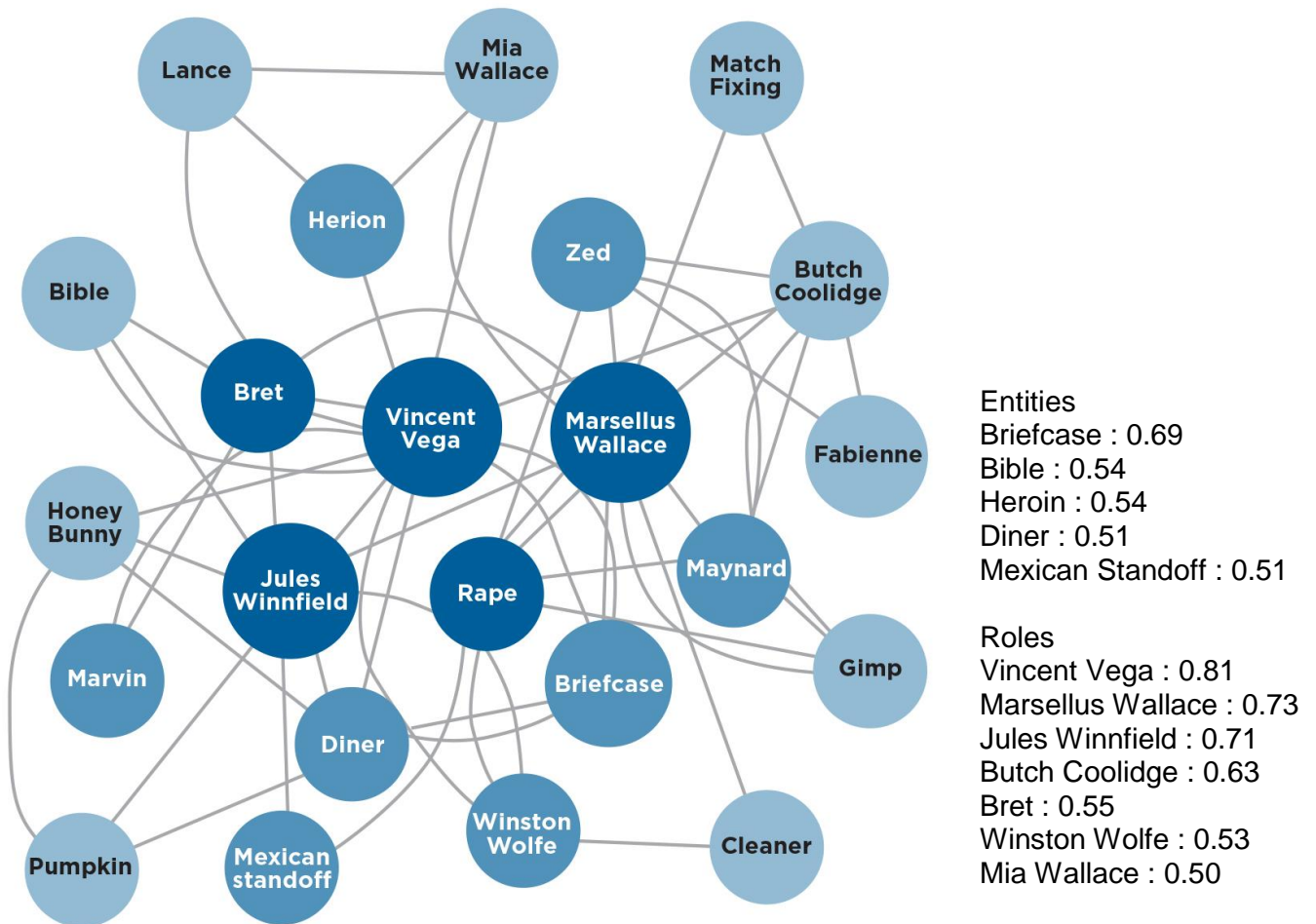
Features	Precision	Recall	F1-Score
Without Graph Features	0.611	0.887	0.723
With Graph Features	0.879	0.82	0.848

**Table 1 – Results on Test Data**

The recall is higher in the model without graph features, and precision is low as expected, because the model without graph feature is unable to distinguish between high-quality and low-quality entities.

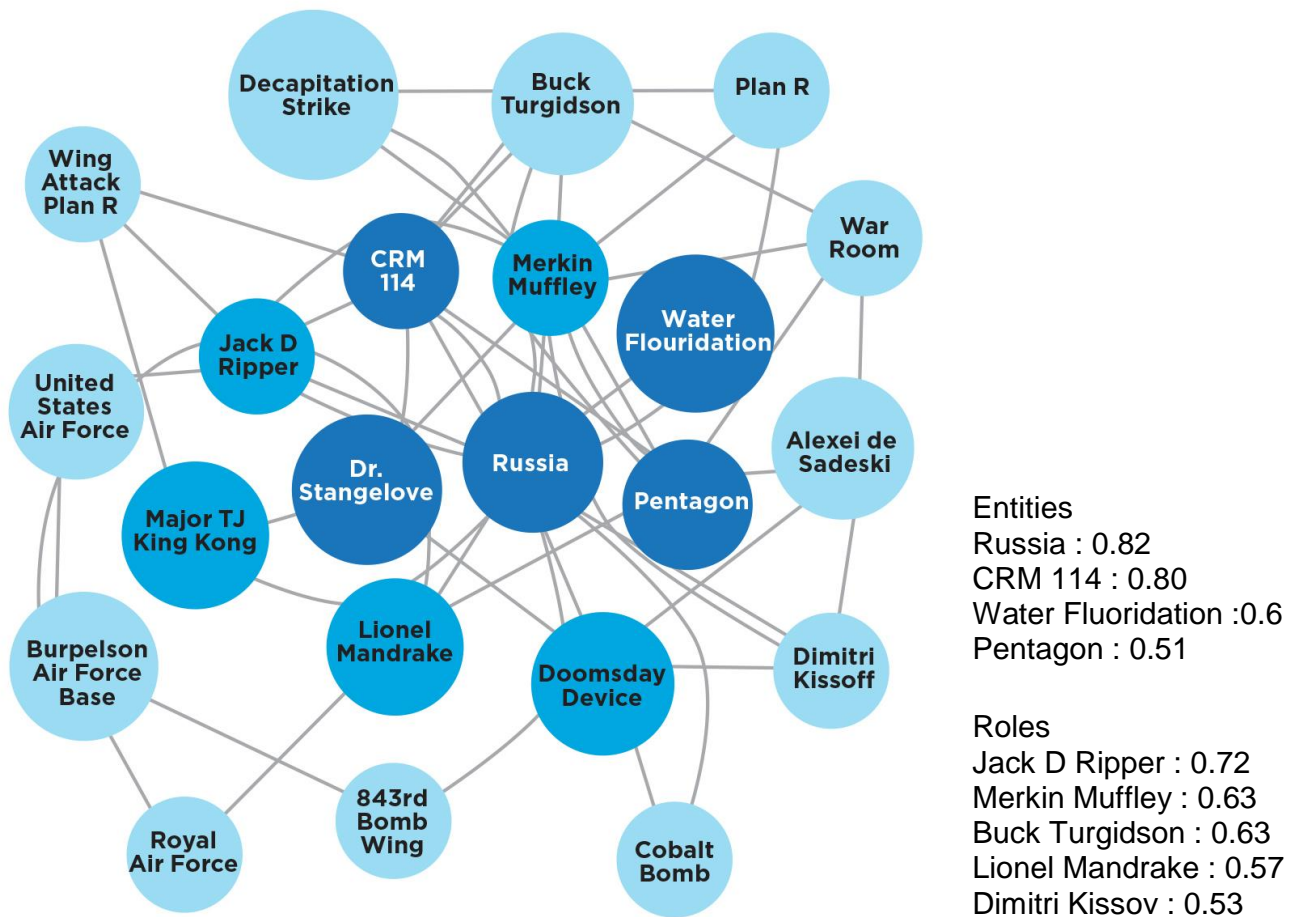
**Examples:**

Following are a few examples of entities and roles extracted by our process. The low-score nodes have been removed for easy representation.

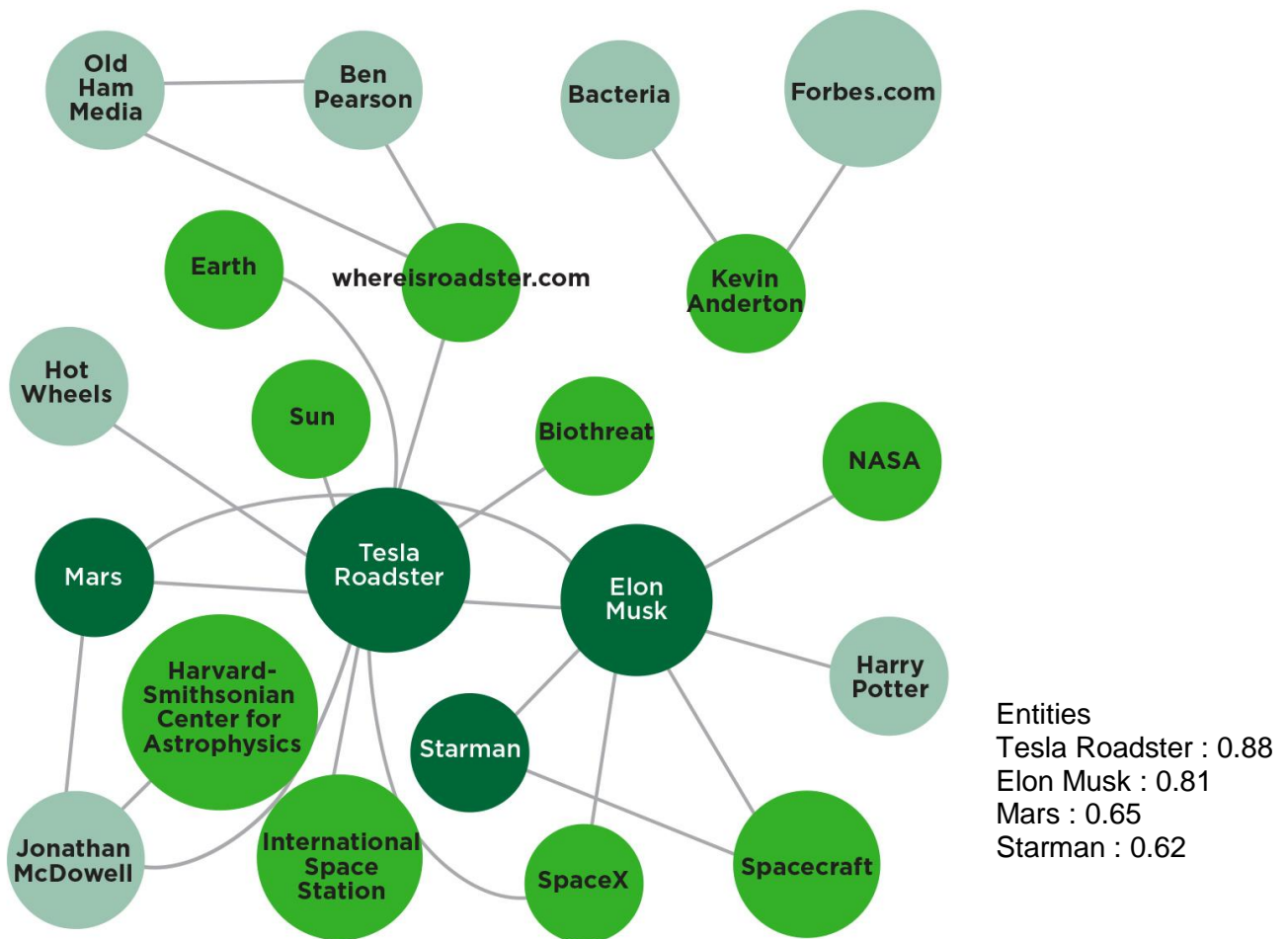


**Figure 3 'Pulp Fiction'**





**Figure 4' Dr Strangelove: Or How I Learned to Stop Worrying and Love the Bomb'**



**Figure 5 ‘Sending Tesla Roadster to Mars’ (8)**

In the movie *Pulp Fiction* (Figure 3), we have the entity “Briefcase” with a high score (as it is the McGuffin driving the plot), which would be difficult to surface with statistical models like TF-IDF. The TF-IDF score of a generic term like “Briefcase” would be very low, and the statistical model fails to grasp the semantic relevance of the phrase in the context of the movie.. In the movie *Dr. Strangelove* (Figure 4), we successfully identify important entities like “Russia,” “CRM-114” and “Water Fluoridation,” all of which would not have been extracted by traditional models. It is also observed that the roles integral to the plot of the movie receive higher scores.

In Figure 5, we ran our model through a news article, “Sending Tesla Roadster to Mars” (8). We successfully extracted entities like “Tesla Roadster,” “Elon Musk,” “Mars” and “Starman” while removing the “noise” – unimportant keywords like “Kevin Anderson,” “bio threat,” “Harry Potter” and “bacteria.”



## APPLICATIONS IN SEARCH AND DISCOVERY

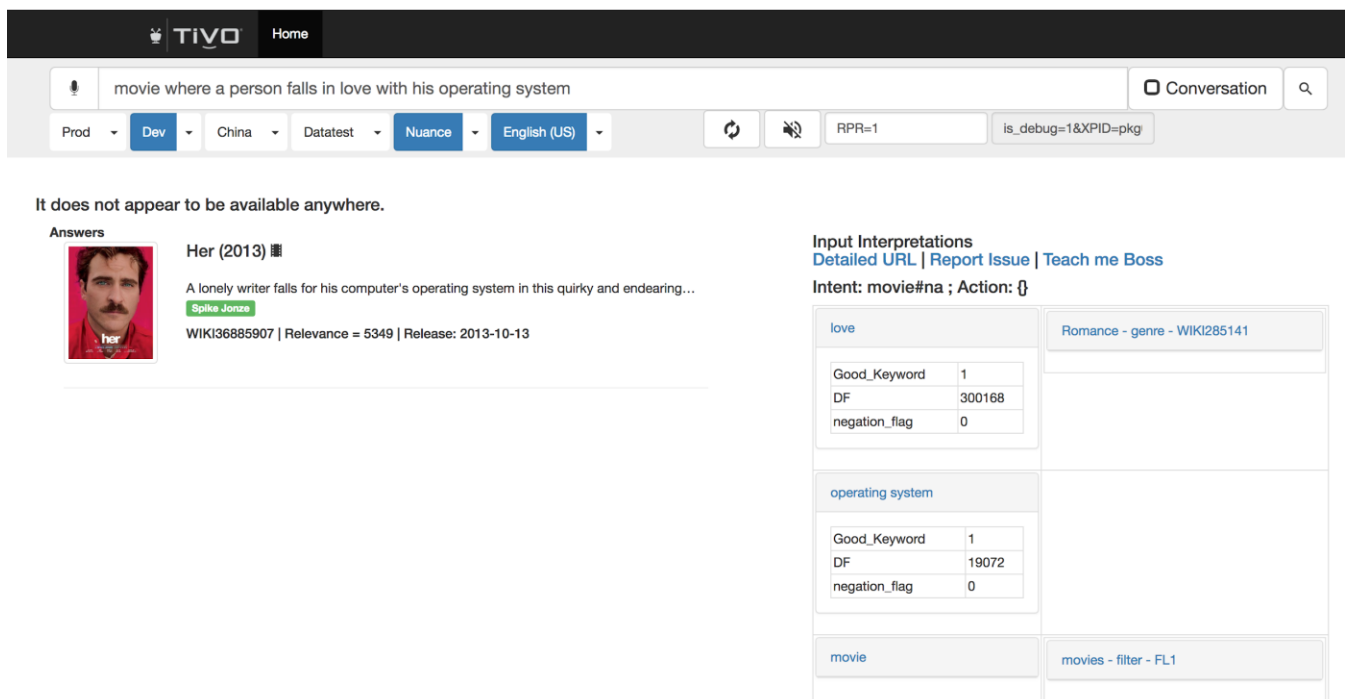
Keywords extracted from models driven by statistical methods like TF-IDF do not really distinguish between contextual elements from irrelevant ones. Keywords derived from semantic graphs take a completely different approach in measuring the relevance by means of informativeness and graph connections to other important topics.

The following examples show how keywords from semantic graphs demonstrate deeper understanding of content and provide rich search experience.

For example, if one were to use voice to find a “movie where a person falls in love with operating system” the semantic graph returns the movie *Her*.

Semantic graph for this movie, which is built based on Wikipedia plot details, understands that “love” and “operating system” are highly relevant and contextual keywords for this movie. The semantic keywords are flagged as “Good\_Keyword” and indexed with higher weight in the search system.

By contrast, a generic term like “love” has a very high term and document frequencies, which traditional TF-IDF based models will not consider as a good weight keyword. The semantic graph approach looks beyond just the stats and measures the relevance of the keyword based on the contextual importance.



The screenshot shows a TIVO search interface. The search bar contains the query "movie where a person falls in love with his operating system". Below the search bar, there are several dropdown menus for "Prod", "Dev", "China", "Datatest", "Nuance", and "English (US)". To the right of the search bar, there are buttons for "Conversation" and a search icon. Below the search bar, there are several buttons for "Prod", "Dev", "China", "Datatest", "Nuance", and "English (US)". To the right of the search bar, there are buttons for "Conversation" and a search icon. Below the search bar, there are several buttons for "Prod", "Dev", "China", "Datatest", "Nuance", and "English (US)". To the right of the search bar, there are buttons for "Conversation" and a search icon.

It does not appear to be available anywhere.

Answers

**Her (2013)**

A lonely writer falls for his computer's operating system in this quirky and endearing...

**Spike Jonze**

WIKI36885907 | Relevance = 5349 | Release: 2013-10-13

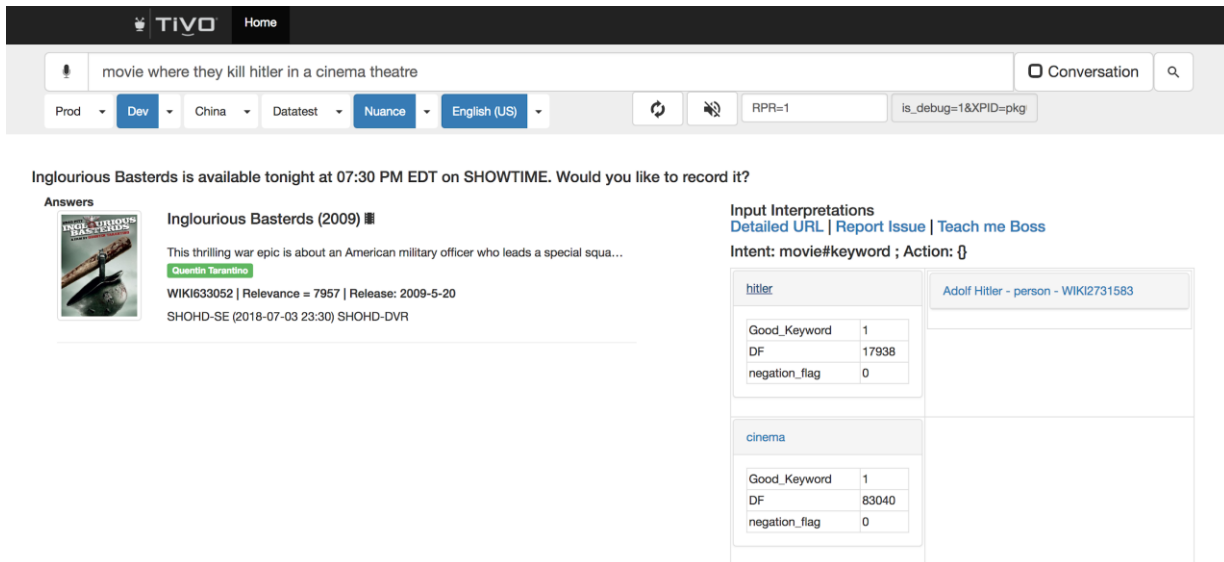
Input Interpretations  
[Detailed URL](#) | [Report Issue](#) | [Teach me Boss](#)

Intent: movie#na ; Action: {}

love	Romance - genre - WIKI285141						
<table border="1"> <tr><td>Good_Keyword</td><td>1</td></tr> <tr><td>DF</td><td>300168</td></tr> <tr><td>negation_flag</td><td>0</td></tr> </table>	Good_Keyword	1	DF	300168	negation_flag	0	
Good_Keyword	1						
DF	300168						
negation_flag	0						
operating system							
<table border="1"> <tr><td>Good_Keyword</td><td>1</td></tr> <tr><td>DF</td><td>19072</td></tr> <tr><td>negation_flag</td><td>0</td></tr> </table>	Good_Keyword	1	DF	19072	negation_flag	0	
Good_Keyword	1						
DF	19072						
negation_flag	0						
movie	movies - filter - FL1						

Figure 6 ‘Searching for the Movie *Her*’

In another example, one could ask to find a “movie where they kill Hitler in a cinema theatre”. We are able to extract “Hitler” and “cinema” as important keywords and return *Inglorious Bastards*.



The screenshot shows a TIVO search interface. The search bar contains the query "movie where they kill hitler in a cinema theatre". Below the search bar, there are navigation tabs for "Prod", "Dev", "China", "Datatest", "Nuance", and "English (US)". The search results section displays "Inglorious Basterds (2009)" with a small image of the movie cover. To the right of the search results, there is a section titled "Input Interpretations" which shows the extracted keywords and their associated data.

Keyword	Good_Keyword	DF	negation_flag
hitler	1	17938	0
cinema	1	83040	0

Figure 7 ‘Searching for the Movie *Inglorious Bastards*’

## APPLICATIONS IN RECOMMENDATIONS

Semantic Graph gives the most important nodes from the unstructured text (Wikipedia Movie plot). Some of the nodes will have backlinks(Entities) and others are just keywords without links. The Entities are considered as semantic concepts and similarities of entities is used in recommendations.

For example, in the movie *Margin Call*, “financial crisis”, “mortgage-backed security” are important thematic concepts. The recommendation system leverages these and recommends similar movies like *The Big Short*, and *Too Big to Fail*.



Margin Call

[WIKI27863892](#)

DEV : 20180702T160023

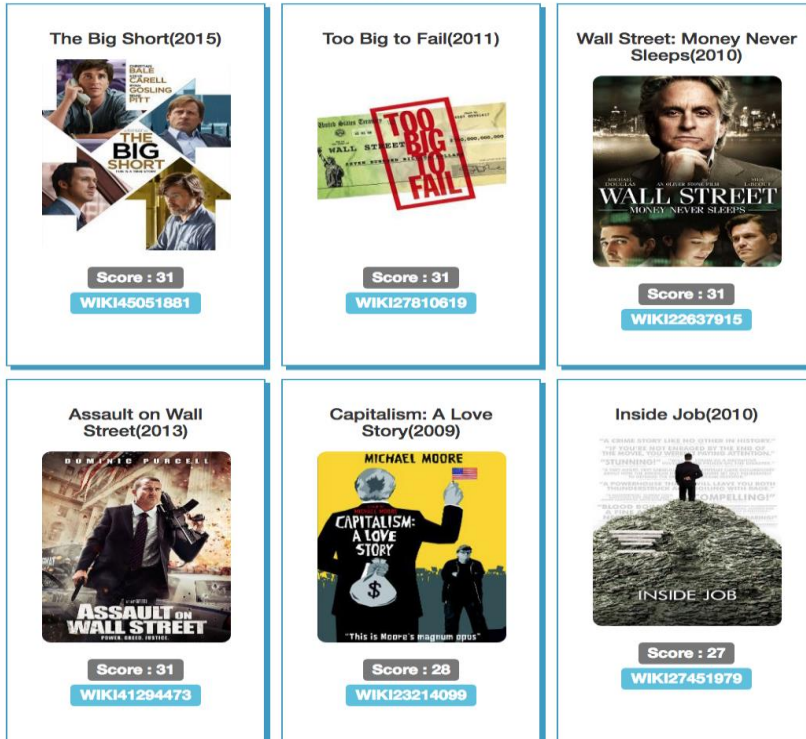


Figure 8 'Recommendations for the movie *Margin Call*'

In *Argo*, "CIA" is an important concept and the recommendation engine pulls relevant movies based on the same concept.



Argo

[WIKI33028800](#)

DEV : 20180702T160023

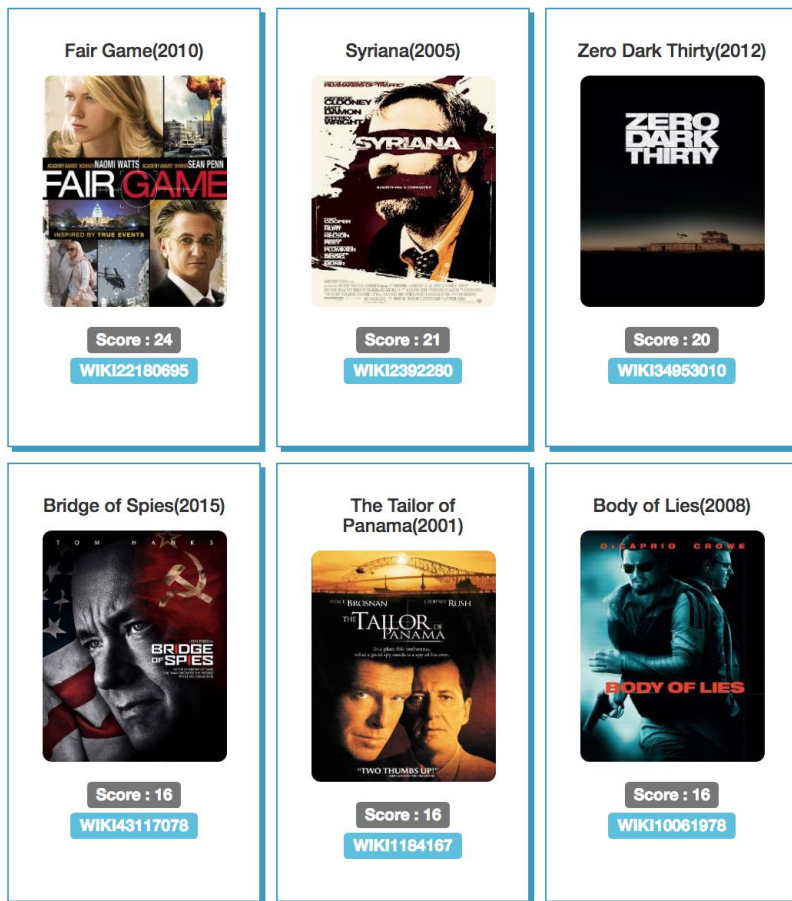


Figure 9 'Recommendations for the movie *Argo*'

## ADDITIONAL APPLICATIONS

As shown in the figures above, semantic graph features can be applied to a variety of content, not just movies and TV shows, but also news articles, short-form content and even one-time events, such as award shows. The information gleaned from these graphs can be applied in improving the discovery of content, and create relevant results and meaningful recommendations for consumers:

**Trending Topic Identification:** Extraction of trending topics from unstructured sources like Google News. From a news article, we highlight the most relevant entities and suppress noisy entities of fleeting mentions. The semantic graph's node-scoring mechanism helps us to evaluate the most relevant entities



**NER (Named Entity Extraction):** Automatic extraction of contextually important entities or keywords from unstructured text (i.e., news article, content description) for content discovery

**Role Importance:** Classification of important and unimportant cast members and roles in a movie based on the node score from the semantic graph. For example, in Figures 3 and 4, the important roles achieved a very high score

## CONCLUSION

Organisations can effectively use semantic graphs in combination with machine learning to gain a deeper understanding of content – quickly identifying relevant entities/keywords based on context and extending entertainment discovery beyond sometimes exhausting “search and find” methods. Viewers are no longer tied to remembering an exact title or character, but can use natural language to find the content they are interested in. This foundation for contextually relevant, voice-powered search results and recommendations satisfies consumers’ desire to quickly find the right content and allows content owners to increase viewership of their long-tail catalogues.

## REFERENCES

- [1] New TiVo Survey Reveals “Show-Dumping” is the Latest Challenge as Consumers Navigate Content Choices, 2016, [www.businesswire.com](http://www.businesswire.com)
- [2] Lorand Dali, Delia Rusu, Blaž Fortuna, Dunja Mladenić and Marko Grobelnik, 2009, Question Answering Based on Semantic Graphs, In Proceedings of the Workshop on Semantic Search (Sem-Search 2009), pp. 24-30
- [3] I. F. Moawad and M. Aref, 2012, Semantic graph reduction approach for abstractive Text Summarization, In Proceedings of International Conference on Computer Engineering and Systems (ICCES 2012), pp. 132-138
- [4] Kenton Lee, Luheng He, Mike Lewis and Luke Zettlemoyer, 2017, End-to-end Neural Coreference Resolution, In Proceedings of Empirical Methods in Natural Language Processing (EMNLP 2017), pp. 188-197
- [5] L. C. Freeman, 1979, Centrality in social networks: Conceptual clarification, Social Networks, vol. 1, pp. 215-239
- [6] L. C. Freeman, 1977, A set of measures of centrality based on betweenness, Sociometry, vol. 40, no. 1, pp. 35-41
- [7] Y.H. Kerner, Z. Gross, and A. Masa. 2005. Automatic extraction and learning of zeyphrases from scientific articles. In Proceedings of Computational Linguistics and Intelligent Text Processing (CICLing 2005), pp. 657–669
- [8] Musk's Tesla going where no car has gone before, [http://qconline.com/news/local/musk-s-tesla-going-where-no-car-has-gone-before/article\\_52e8512c-e9bf-5de4-8d39-50227087d87e.html](http://qconline.com/news/local/musk-s-tesla-going-where-no-car-has-gone-before/article_52e8512c-e9bf-5de4-8d39-50227087d87e.html)