

2016

# Identifying Tweets with Implicit Entity Mentions

Adarsh Koruthu Alex  
*Wright State University*

Follow this and additional works at: [https://corescholar.libraries.wright.edu/etd\\_all](https://corescholar.libraries.wright.edu/etd_all)



Part of the [Computer Sciences Commons](#)

---

## Repository Citation

Alex, Adarsh Koruthu, "Identifying Tweets with Implicit Entity Mentions" (2016). *Browse all Theses and Dissertations*. 1577.  
[https://corescholar.libraries.wright.edu/etd\\_all/1577](https://corescholar.libraries.wright.edu/etd_all/1577)

This Thesis is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact [corescholar@www.libraries.wright.edu](mailto:corescholar@www.libraries.wright.edu), [library-corescholar@wright.edu](mailto:library-corescholar@wright.edu).

# Identifying Tweets with Implicit Entity Mentions

A thesis submitted in partial fulfilment  
of the requirements for the degree of  
Master of Science

By

ADARSH ALEX  
B.E, University of Mumbai, 2013

2016  
Wright State University

WRIGHT STATE UNIVERSITY  
GRADUATE SCHOOL

AUG 28, 2016

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Adarsh Alex ENTITLED Identifying Tweets with Implicit Entity Mentions BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science.

---

Amit P. Sheth  
Thesis Director

---

Mateen Rizki  
Chair, Department of Computer Science  
and Engineering

Committee on  
Final Examination

---

Amit P. Sheth, Ph. D

---

Krishnaprasad Thirunarayan, Ph. D

---

Tanvi Banerjee, Ph. D

---

Robert E. Fyffe  
Vice President for Research and Dean of  
the Graduate School

## ABSTRACT

ALEX, ADARSH. M.S., Department of Computer Science and Engineering, Wright State University, 2016. *Identifying Tweets with Implicit Entity Mentions*

Social networking sites like Twitter and Facebook have become a significant source of user-generated content in the past decade. Mining of this user-generated content has proved beneficial for a broad range of applications like Event Extraction, Document Retrieval, and Sentiment Analysis. Identifying entities is one of the major tasks that fuel important information for above tasks. Identification of entities is typically performed in two steps; Named Entity Recognition (NER) and Entity Linking. State of the art NER solutions focus on recognizing the entities that are mentioned explicitly in social media posts. However, entities are frequently mentioned implicitly in them. For example, the tweet ‘*Didn’t know that its the same actress in Fault in our stars and Divergent.*’ contains explicit references to movies *Fault in our stars* and *Divergent* while it implicitly refers to actress *Shailene Woodley*. Spotting and classifying tweets with such implicit entity mentions (i.e. recognize that above tweet has implicit entity of type ACTRESS) is the initial step towards identifying the implicit mention of *Shailene Woodley* in this tweet.

In this thesis, we propose a two step semantic driven approach to address the spotting and typing of implicit entity mentions in text. Specifically, we answer two research questions in this thesis:

1. How to find tweets that have implicit entity mentions of a given type?
2. What features help to distinguish tweets with implicit entity mentions from tweets with explicit entity mentions and tweets with no entity mentions at all?

We answer the first question by developing a technique to find semantic cues that indicate the presence of implicit entity mentions in tweets. The second research question is answered by exploiting the syntactic features of the tweets, along with semantic features extracted from crowd-sourced knowledge bases like Wikipedia and DBpedia, to determine whether a tweet has an implicit entity mention or not. We evaluate our approach by creating a gold standard dataset for two domains namely movies and books.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Related Work</b>	<b>7</b>
2.1	Named Entity Recognition on Organized Text . . . . .	8
2.2	Named Entity Recognition in Unorganized Text . . . . .	8
2.3	Role of Background Knowledge in Text Analysis . . . . .	9
2.3.1	Wikipedia as Background Knowledge . . . . .	10
<b>3</b>	<b>Background</b>	<b>12</b>
3.1	Wikipedia . . . . .	12
3.2	DBpedia . . . . .	13
3.3	Semantic Similarity . . . . .	14
3.4	Word2Vec . . . . .	14
3.5	Random Forest . . . . .	15
<b>4</b>	<b>Approach</b>	<b>16</b>
4.1	Identifying tweets with potential implicit entity mentions . . . . .	17
4.1.1	Formal Semantic Cues . . . . .	18
4.1.1.1	Head Nouns . . . . .	19
4.1.2	Twitter Specific Semantic Cues . . . . .	21

4.2	Classifying Tweets as Implicit, Explicit and Null . . . . .	23
4.2.1	Domain Relevant Entities . . . . .	23
4.2.1.1	Domain Relevant Entities Using DBpedia . . . . .	24
4.2.1.2	Commonness . . . . .	25
4.2.1.3	Domain Relevant Entities Using Wikipedia . . . . .	26
4.2.2	Window Based Bigrams . . . . .	28
4.2.3	Explicit Entity Mentions . . . . .	29
4.2.4	Part-Of-Speech Tags . . . . .	29
4.3	Classifying Tweets into Predefined Type . . . . .	30
<b>5</b>	<b>Evaluation</b>	<b>31</b>
5.1	Semantic Cue Evaluation . . . . .	31
5.2	Classifying Tweets as Implicit, Explicit and Null . . . . .	33
5.2.1	Dataset . . . . .	33
5.2.2	Evaluation Metrics . . . . .	35
5.2.3	Results of Classification . . . . .	35
5.2.3.1	Error Analysis . . . . .	37
5.3	Discussion . . . . .	37
5.3.1	Impact of Relationships on the Classification Step . . . . .	38
5.3.2	Impact of Knowledge on the Classification Step . . . . .	39
<b>6</b>	<b>Conclusion and Future Work</b>	<b>42</b>

# List of Figures

3.1	Sample Wikipedia Page . . . . .	13
4.1	Approach Overview . . . . .	17
4.2	Categories of Furious 7 . . . . .	19
4.3	Parse Tree . . . . .	20
4.4	Semantic Cues for MOVIE . . . . .	22
4.5	Semantic Cues for BOOK . . . . .	23
4.6	DBpedia Subgraph . . . . .	24
4.7	Drawbacks of DBpedia . . . . .	27
4.8	Wikipedia Hyperlink Graph . . . . .	27
5.1	Semantic Cue Evaluation for Movies . . . . .	32
5.2	Semantic Cue Evaluation for Books . . . . .	32
5.3	Semantic Cue Below Similarity of 0.5 for Movies . . . . .	33
5.4	Semantic Cue Below Similarity of 0.5 for Books . . . . .	34
5.5	Impact of Relationships on Classification - Movie . . . . .	38
5.6	Impact of Relationships on Classification - Book . . . . .	39
5.7	Impact of Wikipedia Knowledge on Books . . . . .	40
5.8	Impact of Wikipedia Knowledge on Movies . . . . .	40
5.9	Impact of DBpedia Knowledge on Books . . . . .	41

5.10 Impact of DBpedia Knowledge on Movies . . . . .	41
--	----



# List of Tables

1.1	Example Tweets Filtered by Semantic Cues of the Entity Type MOVIE . . . . .	5
4.1	Tweets with Semantic Cues of Entity Type MOVIE . . . . .	18
5.1	Dataset 1 Statistics . . . . .	34
5.2	Dataset 2 Statistics . . . . .	34
5.3	Classification Results for Books on First Dataset . . . . .	36
5.4	Classification Results for Movies on First Dataset . . . . .	36
5.5	Classification Results for Movies on Second Dataset . . . . .	36
5.6	Classification Results for Books on Second Dataset . . . . .	36

## ACKNOWLEDGEMENTS

My journey through graduate school has been an extraordinary and fulfilling experience. I would like to take this opportunity to thank everyone who has helped me along the way.

First and foremost, I want to express my sincere gratitude towards my advisor Dr. Amit P. Sheth for his continuous guidance. I am thankful to him for providing me with this amazing opportunity to pursue my research interests. His dedication and enthusiasm continues to inspire me even today. I would like to thank Dr. T.K.Prasad for all his insightful comments. I would also like to thank Dr. Tanvi Banerjee for her patience and feedback on this work.

I would like to thank Sujan Perera for guiding me through every stage of my research. It would have been impossible to complete my thesis without his guidance. I would also like to thank the entire Kno.e.sis team.

This acknowledgement would be incomplete without thanking my family. I am thankful to my parents Alexander and Nancy for all the sacrifices they have made and their continued faith in me. Last but not the least, I would love to thank Kamni for her continuous support and motivation.

# 1

## Introduction

The advent of the World Wide Web has been one of the biggest breakthroughs in technological history. Since its inception in the early 1990s, the web has experienced an exponential growth with respect to the user base and the content. Over the past decade, the web itself has been revolutionized with the introduction of social media sites, blogs and other platforms which allow human interactions on the web. The rapid growth of these platforms especially social media sites like Twitter and Facebook has given users a common space in which they can communicate and express their opinions.

Social media has had a tremendous impact in our day-to-day life. Twitter, a microblogging website is one of the social networking giants. The latest Twitter statistics show that there are 500 million tweets per day with an active user base of 320 million users per month. These numbers are increasing at an exponential rate as each day passes by. The discussion topics of the tweets range from what people have done during the course of the day, opinions about new movie, climate change on earth, to presidential elections in USA. Twitter has been extensively used for a variety of applications like event extraction [Ritter et al. 2012], opinion analysis [Pang and Lee 2008] and earthquake detection [Sakaki et al. 2010].

Identifying entities mentioned in tweets is a critical component for all applications mentioned above. Identification of entities is typically performed in two steps. The first step is termed named entity recognition which spots and classifies rigid designators in text to predefined types (e.g. PERSON, LOCATION, OR-

GANISATION) [Nadeau and Sekine 2007]. The second task aims to assign unique identities to the spotted entities by the named entity recognition task w.r.t a knowledge base, and it is termed as entity linking [Rao et al. 2013].

The literature on identifying entities in tweets has focused on explicitly mentioned entities. However, it is observed that more often than not entities are mentioned implicitly in tweets. Consider the tweet *'Didn't know that its the same actress in Fault in our stars and Divergent'*. This tweet mentions the movies *Fault in our Stars* and *Divergent* explicitly, while it implicitly refers to actress *Shailene Woodley*. This thesis is focused on recognizing the tweets with implicit entity mentions of a given entity type. i.e. it aims to recognize that the above tweet has a mention of an implicit entity of type ACTRESS. In other words, this thesis proposes techniques to perform mirror tasks performed by named entity recognition for explicit entities to implicit entities in tweets.

Recognizing tweets with implicit entities of a given entity type is an important task and has great value in applications like event extraction, document retrieval, and opinion mining.

1. **Event Extraction** :Tweets contain up-to-date information and inclusive stream of the current events.

Extraction of such events in real time is invaluable and can lead to key insights. Entity identification is the preliminary step in recognizing the events in tweets [Zhou et al. 2011]. Consider the following tweet which talks about the Presidential campaign of 2016: *'Republican Presidential Candidate says he wants to end birthright citizenship, was born in Canada.'* An automatic event recognition system should be able to understand that the above tweet has a mention of *Ted Cruz* to recognize the event of the announcement. The first step of understanding the mention of *Ted Cruz* in the above tweet is to identify that this tweet has an implicit mention of entity of type POLITICIAN.

2. **Document Retrieval** : Twitter contains a rich source of information about a broad set of topics ranging from healthcare to political activities. People share everything from what they had for lunch to new findings in the fields of science, technology and medicine [Java et al. 2007]. This shared information can cater to the needs of the users seeking information. Soni et al. [Soni 2015] developed a near real time

document retrieval framework for health related documents; and has proved that Twitter can serve as a good source of reliable information. Searching on Twitter enables users to obtain instantaneous updates on issues of their interest (eg., health, politics etc.) in real time, and from multiple perspectives. Building on similar lines, Twitter has been used as a source to retrieve information relevant to a given query [Efron 2011] in the recent past. However, identifying entities are central to retrieving documents relevant to the search query. The following tweet implicitly refers to Respiratory Distress Syndrome (RDS) which is very common among infants: *'(Reuters) - A new drug to treat a breathing disorder that is the leading cause of death of premature babies... <http://bit.ly/18Z64eV>'*. However the key challenge in this scenario is to identify the presence of an entity of type DISEASE in this tweet.

3. **Sentiment Analysis :** Sentiment analysis is vital to understand public opinion on social issues. On February 26, 2015 marijuana was legalized in Washington DC, leading to an enormous amount of tweets regarding this topic. Policy makers were interested in analyzing the sentiment across the US cities on marijuana legalization and tweets were a valuable data stream to get these insights. For example, one of the tweets was as follows: *'Only in America, its legal to smoke weed in the capital, but not in the rest of the country.'* On reading the tweet, we can arrive at the conclusion that the user has a negative sentiment towards the marijuana legalization policy in *Washington DC*. In order to reach that conclusion, one must identify the presence of the implicit entity type CITY in the tweet mentioned above.

The above mentioned examples are few of the many use cases that show the value of identifying implicit entity mentions in tweets. It is clear that any downstream application will be handicapped without identifying the implicitly mentioned entities in tweets. These use cases served as the motivation to solve the problem of identifying tweets with implicit entity mentions of a given entity type.

The nature and the complexity of this problem is different from that of Named Entity Recognition (NER). NER solutions have generally taken two sets of approaches. The first set of approaches exploits the lexical and orthogonal features of the tweets to train sequence labeling algorithm like conditional random fields to spot the entity mentions in tweets and classify them to given types [Liu et al. 2011]. Such an approach uses features

like capitalization, punctuations, part-of-speech tag of the words and spots the entities referred by noun phrases in the tweets and classifies them to predefined types. The second set of approaches builds a vocabulary that contains the names of the entities and their alternative forms (e.g. synonyms, alias, abbreviations). This is used to check for the presence of word/phrases of the vocabulary in the given tweet to spot the entity mentions [Ritter et al. 2011]. The vocabulary is typically built with Wikipedia page titles, anchor texts that appear in Wikipedia pages, and entity labels available in knowledge bases like Freebase.

Nevertheless, none of above two approaches are applicable to spot the implicit entity mentions. As described, they assume the presence of the entity name in the tweet in the form of a noun phrase which can be identified by leveraging syntactic features or dictionary features. However, the main distinguishing feature of the implicit entity mentions from that of explicit entity mentions are as follows:

- Implicit entity mentions do not have a name or an alternative name of the entity and,
- Implicit entity mentions are not always noun phrases nor are they contiguous phrases in the tweet.

These characteristics of implicit entity mentions warrants a new solution to solve the problem of spotting the implicit entity mentions of a given entity type.

**We define the problem as, given the tweet  $w$  and a set of interested entity types  $T$  (e.g. MOVIE, BOOK, ACTOR), recognize whether  $w$  has implicit mention of entity of type  $t \in T$ .**

This thesis addresses two research questions:

1. How to find tweets that have implicit entity mentions of a given type?
2. What features help to distinguish tweets with implicit entity mentions from tweets with explicit entity mentions and tweets with no entity mentions at all?

Our approach exploits the fact that a tweet with an implicit entity typically contains semantic cues for identifying the type of implicit entity mention. As seen in the above example tweets, the terms ‘actress’, ‘candidate’, ‘disorder’ and ‘capital’ indicates the type of the entity being mentioned implicitly. However, the challenge here is to compile a list of terms that indicate the presence of given entity type. This is a

No	Tweet
1	This flick marked the directorial debut of James Wong
2	heres when the movie starts getting sad augustus tells hazel his cancer came back
3	We gonna have movie night today! wanna join?
4	I finally saw The Imitation Game last night It was a rather entertaining flick
5	An amazing flick goal by Jamie Vardy.

Table 1.1: Example Tweets Filtered by Semantic Cues of the Entity Type MOVIE

difficult task due to the fact that Twitter is a very informal medium to communicate and Twitter users use a very diverse vocabulary. For example, Twitter users use formal terms like ‘movie’, ‘film’ and twitter specific terms like ‘flick’ and ‘directorial’ to indicate entity type MOVIE. In order to address this challenge, we used crowd-sourced knowledge of Wikipedia in assigning categories to entities. The labels assigned to the entities in Wikipedia as their categories (a.k.a Wikipedia Category) provide rich source of information to derive set of formal terms that indicate entity type. We expand this set by capturing a set of twitter specific semantic cues by training a distributional semantic model over tweets. These two steps help in capturing both formal and social media specific semantic cues which aid in identifying plausible tweets with implicit entity mentions. We hypothesise that the tweets containing these semantic cues might have an implicit entity mention.

However, the resulting tweets that are selected by the semantic cue filtering technique can be noisy. For example, all the five tweets mentioned above contain semantic cues of the entity type MOVIE. However, only the first and the second tweet contain implicit references to a MOVIE entity; the third tweet does not contain implicit or explicit reference to a MOVIE, the fourth tweet contains an explicit mention to a MOVIE entity while the fifth tweet contains an explicit reference to a SOCCER PLAYER. Hence, the next task is to classify the tweets to three categories; explicit, implicit, and null. The explicit and implicit classes indicate that the tweet has an explicit or implicit mention of an entity of a given type. The null class indicates that the tweet does not have a mention of an entity of a given type. We use a combination of knowledge driven and syntactic

features of the tweet to perform these classification tasks. The knowledge driven features are derived from Wikipedia and DBpedia knowledge bases and represent the knowledge about a domain (e.g. MOVIE). Such knowledge with syntactic features like part of speech tags, n-grams, and tweet length help to classify tweets into explicit, implicit, and null categories.

The rest of thesis is organized as follows. In Chapter 2, we survey related work in the field of Named Entity Recognition and then follow it up by studying the role of background knowledge in text analysis. In Chapter 3, we introduce all the algorithms, tools and resources used in this research. In Chapter 4, we explain the approach taken to solve the research problem. In Chapter 5, we evaluate our approach on a manually created gold standard dataset. Finally, in Chapter 6, we present the future work.



## 2

# Related Work

Named Entity Recognition is the research area which is closest to this work. Named Entity Recognition spots rigid designators in text and classifies them into predefined categories such as MOVIE, PERSON, BOOK [Nadeau and Sekine 2007]. However, Named Entity Recognition systems work on the assumption that the entity mentions are always contiguous noun phrases. The task of identifying implicit entities in tweets does not rely on either of these assumptions. By definition, implicit entities do not have entity names and they are not guaranteed to be manifested as contiguous noun phrases. Consider the tweet ‘*when Augustus tells Hazel his cancer is back is the point where i lose my shit for the rest of the movie*’; this tweet has an implicit mention of the MOVIE entity *The Fault in our Stars*. However, the phrases that help us identify this implicit entity mention are ‘*Augustus/NP tells/VP Hazel/NP*’ and ‘*movie/NP*’ which are not manifested as continuous text segments neither are they noun phrases.

Although the problem addressed in this thesis is significantly different from that of NER, the techniques used and the general area of work are quite similar. Hence we do an extensive survey of Named Entity Recognition. NER has received significant attention from researchers. State of the art NER systems have been developed for both organized text (News articles) and unorganized text (Tweets). In the rest of this chapter we study Named Entity Recognition for organized text, unorganized text and then we survey the impact of knowledge on text mining tasks.

## 2.1 Named Entity Recognition on Organized Text

McCallum et al. [McCallum and Li 2003] presented a work which used feature induction with Conditional Random Fields (CRF). The evaluations are done on the CoNLL-2003 [Tjong Kim Sang and De Meulder 2003] English shared task dataset. The dataset contains text files for English and German languages. They used language level features which were extracted using regular expressions. Along with the language level features they also used general purpose lexicons that they extracted from the Web. They achieved a F1 score of 84.04% on the English dataset using 6423 features that were extracted using the above techniques.

Asahara et al. [Asahara and Matsumoto 2003] uses Support Vector Machines (SVM) which is a supervised learning algorithm, to perform named entity recognition on Japanese texts. The dataset used for evaluation consists of 1174 Japanese newspaper articles. The evaluation is done using five-fold cross validation. The work achieves an F1 score of 87.21% on the test dataset. They use syntactic features such as POS tags, characters, character tags and Named Entity tags to train and test the SVM model.

The two methods mentioned previously are supervised machine learning techniques and hence require huge amounts of training data to achieve good performance. Unsupervised techniques have also been studied for identifying named entities in organized text. Unsupervised techniques do not require any training data to perform the task and hence there is less overhead of performing manual annotations.

Shinyama et al. [Shinyama and Sekine 2004] identifies named entities using the observation that named entities occur synchronously across documents while the same cannot be said about noun phrases or proper nouns. This technique does not produce great recall as compared to the other state of the art techniques. However, this technique identifies about 90% rare named entities.

## 2.2 Named Entity Recognition in Unorganized Text

All the above mentioned techniques perform well on organized text like News articles. However, they cannot be applied to unorganized text like tweets. The short and terse nature of tweet make it extremely difficult to perform text mining applications on them. However, named entity recognition on Twitter has received

substantial traction in the recent past.

Ritter et al. [Ritter et al. 2011], retrained the whole Natural Language Processing (NLP) pipeline for tweets. He uses LabelledLDA along with Freebase dictionaries to use the knowledge about entities to perform the task of Named Entity Recognition. He uses POS tags, chunking, capitalization and dictionary features to efficiently identify named entities in tweets. The test dataset was manually created as there is no standard gold standard dataset available to evaluate the performance of the system. The system produces an F1 score of 66% which is very good in the context of tweets.

Liu et al. [Liu et al. 2011], uses a combination of K Nearest Neighbors and Conditional Random Fields in a semi-supervised setting to perform the task. This method uses bag of words features along with orthogonal, lexical and gazetteers related features. The proposed approach performs reasonably well and produces an F1 score of 78.24% on a manually created dataset of tweets.

### **2.3 Role of Background Knowledge in Text Analysis**

As explained in the previous two sections text analysis systems can be broadly classified into two categories: Supervised and Unsupervised. Supervised techniques require huge amount of training data that has to be manually annotated by humans. On the contrary unsupervised techniques do not rely on human annotators to perform the task.

Background knowledge, is the prior knowledge that humans use to make sense of information. For example, a human can infer that this tweet '*Sandra Bullocks space movie is amazing*' contains a reference to the movie 'Gravity'. The reason being that 'Sandra Bullock' is an actress and she has acted in only one space movie. Background knowledge can be expressed in a variety of different ways. From the computer science perspective background knowledge has been traditionally expressed as dictionaries, thesaurus and knowledge graphs or ontologies. A knowledge graph or an ontology is a collection of classes and relationships. The classes can have a huge set of concepts associated with it and two concepts or two classes can have one or more relationships between them. These ontologies can be domain specific or domain independent. SNOWMED,

UMLS are two of the most popular domain specific ontologies.

WordNet<sup>1</sup> is an example of a lexical knowledge-base. It consists of nouns, verbs, adjectives and adverbs of English language. These words are grouped into sets of synonyms called synsets. Furthermore, each of the synsets are linked to others through conceptual relations such as hypernymy and hyponymy. Richardson et al. [Richardson et al. 1994] used Wordnet to measure the conceptual similarity between words. Wang et al. [Wang and Domeniconi 2009] used WordNet synset to exploit relationship between terms that do not co-occur frequently. They showed that text clustering algorithms perform better on documents enriched with background knowledge compared to documents represented as bag-of-words.

Background knowledge has been extensively used for a wide variety of applications. Perera et al. [Perera et al. 2014] uses background knowledge to identify causal relationships between entities in a medical ontology. Gruhl et al. [Gruhl et al. 2009] leverages a music ontology for linking entities in texts relevant to the music domain.

### 2.3.1 Wikipedia as Background Knowledge

Wikipedia which is a collaborative encyclopedia has been empirically used as a knowledge base for a wide variety of text mining applications. Each Wikipedia article represents a single concept. Gabrilovich et al. [Gabrilovich and Markovitch 2006] used the content extracted from Wikipedia pages to enrich document representation for the task of text categorization. Each Wikipedia article is represented as a vector of words that appear in the article. Then, machine learning techniques are used to map text from documents to the aforementioned vector representation of Wikipedia concepts. On a test dataset consisting of documents from Reuters and OSHUMED, they showed that knowledge extracted from Wikipedia is very useful in categorizing short documents. Mukherjee et al. [Mukherjee and Bhattacharyya 2012] proposed an unsupervised approach to perform sentiment analysis of movie reviews. They used the domain-specific information such as crew, plot and character information from the infobox of Wikipedia page review. Their system did not need any labelled data for training and achieved comparable results to the semi-supervised and unsupervised state-of-

---

<sup>1</sup><http://wordnet.princeton.edu/>

the-art systems.

Concepts in Wikipedia are organized in a category structure where each concept belongs to one category. Hu et al. [Hu et al. 2009] used the category structure of Wikipedia for the purpose of document clustering. Each word in the document is weighted using tf-idf and associated Wikipedia concepts and categories are retrieved. Thus a given document is represented as a vector of weighted terms occurring in the document, a vector of relevant Wikipedia concepts and a vector of categories of the concepts. Finally, partitional clustering is used to compute the similarity between the vectors of two documents. Their tests on three datasets showed that category information is useful in document clustering. The category structure of Wikipedia has been utilized by Genc et al. [Genc et al. 2011] to classify tweets. Their approach first maps each tweet to the most relevant Wikipedia concept and further leverages the category structure to find the semantic distance between the mapped concepts for classification. Kapanipathi et al. [Kapanipathi et al. 2014] used an adaptation of spreading activation theory on the category structure to determine the hierarchical interests of users based on their tweets.

Each Wikipedia article contains links to other Wikipedia articles. These links are termed as inter wiki links. Milne et al. [Witten and Milne 2008] used the link structure of Wikipedia to compute semantic relatedness between two terms using the hyperlinks found in their respective Wikipedia articles. First, they use anchor text to determine the link the Wikipedia page that maps to a given term. Then they measure the similarity of the Wikipedia articles using Normalized Google Distance between the vector of links found in the two Wikipedia articles. Krishnamurthy et al. [Krishnamurthy 2015] uses the hyperlink structure of Wikipedia to identify entities that are local to a particular location. These local entities are then used along with Twitter user profiles to identify the user location of a particular Twitter user. On the dataset published by Cheng et al. [Cheng et al. 2010] they achieved 55% accuracy with an average error of 429 miles.

# 3

## Background

This chapter provides a brief overview of the resources and algorithms that are used for conducting the research described in this thesis.

### 3.1 Wikipedia

Wikipedia was founded by Jimmy Wales and Larry Sanger in 2001 which is based on Wiki. A Wiki is a website that provides collaborative modification of its content from the web browser. Wikipedia which is a collaborative encyclopedia is the most popular Wiki based site. It has being a prominent source of information and knowledge for humans. Wikipedia is available in 292 different languages and has 18 billion page views and 500 million unique users each month. There are more than 5 million articles on the English version of it and covers broad range of topics.

Each Wikipedia article talks about a single entity or an event [Hu et al. 2009] and are extremely comprehensive in nature. Each Wikipedia page contains a list of categories that explain the entity [Chernov et al. 2006].

The related entities to the entity that is being discussed in the current article are hyperlinked to the current Wikipedia page. For example, as shown in figure 1 the Wikipedia page of *Guardians of Galaxy* contains a link to the Wikipedia page of *Vin Diesel* among other links. These links are referred to as internal links. “An

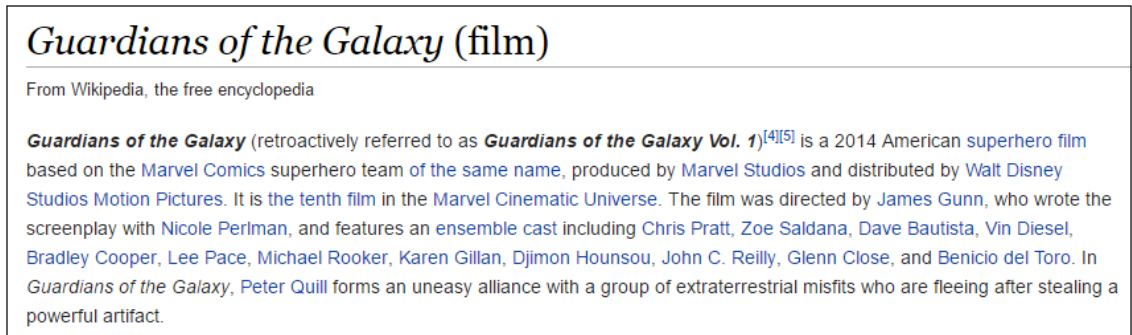


Figure 3.1: Sample Wikipedia Page

internal link is a type of hyperlink on a webpage to another page or resource, such as an image or document, on the same website or domain” according to Wikipedia. The goal of these links are to help the user better understand the particular page.

The hyperlink structure that is created as a consequence of the internal links in Wikipedia can be used as a source of knowledge as it reflects the relationships between entities. Each Wikipedia page is divided into multiple sections. The lead section of a Wikipedia article serves as an introduction to the article and a summary of its most important contents<sup>1</sup>. Consequently, the lead section contains minimal redundant information and all the relevant information regarding the entity.

## 3.2 DBpedia

DBpedia [Auer et al. 2007] extracts structured information from Wikipedia and combines the information into a large multilingual knowledge base. It contains textual descriptions (title and abstracts) of about 38 million concepts in 125 different languages. The English version of DBpedia contains 4.58 million things, out of which 4.22 million are classified in a consistent ontology. Altogether the DBpedia 2014 release consists of 3 billion relationships among entities which is represented as RDF triples. Out of the 3 billion RDF triples 580 million were extracted from the English edition of Wikipedia and the remaining 2.46 billion were extracted from the other language editions. The DBpedia ontology consists of 320 classes and 1650 properties.

<sup>1</sup>[https://en.wikipedia.org/wiki/Wikipedia:Manual\\_of\\_Style/Lead\\_section](https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Lead_section)

### 3.3 Semantic Similarity

Semantic similarity is a metric defined over a set of documents or terms, where the idea of distance between them is based on the likeness of their meaning or semantic content as opposed to similarity which can be estimated regarding their syntactical representation<sup>2</sup>. In literature, semantic similarity between two terms has been calculated using two broad approaches namely Corpus-based and Knowledge-based. Corpus-based measure of word semantic similarity try to identify the degree of similarity between two words using information exclusively derived from a large corpora [Mihalcea et al. 2006]. Latent Semantic Analysis (LSA) [Dumais 2004] is one of the examples of corpus based semantic similarity approaches. Knowledge-based measure of semantic similarity try to identify the degree of similarity between two words by using information drawn from semantic networks or taxonomies [Mihalcea et al. 2006]. Leacock & Chodrow [Leacock and Chodorow 1998], Lesk [Lesk 1986], Wu and Palmer [Wu and Palmer 1994] are some of the most common knowledge-based measures.

### 3.4 Word2Vec

Distributional semantics is a research area that develops and studies theories and methods for quantifying and categorizing semantic similarities between linguistic items based on their distributional properties in large samples of language data. Word2Vec is a distributional semantics tool which provides an efficient implementation of Continuous Bag of Words and Skip-gram architectures for computing vector representations of words. The word2vec tool takes a text corpus as input and produces the word vectors as output. In other words, it takes a huge text file and transforms each word into a k-dimensional vector where the vector is composed of real numbers. On an abstract level the dimensions capture the semantic meaning of each word. If we observe carefully, the similarities between k-dimensional vectors of two words captures the semantic relationships between the words. Mikolov et al. [Mikolov and Dean 2013] showed that simple algebraic operations can be performed on word vectors learned from a text corpus. For example, the algebra opera-

---

<sup>2</sup>[https://en.wikipedia.org/wiki/Semantic\\_similarity](https://en.wikipedia.org/wiki/Semantic_similarity)



tion “King” - “Man” + “Woman” on vectors representing these terms generates a vector closest to the vector representing the term “Queen”.

Word2Vec has two neural network architectures that generate word vectors: Continuous-Bag of Words (CBOW) and Skip-Gram [Mikolov and Dean 2013]. The CBOW model learns a neural network such that given a set of context words surrounding the target word, it tries to predict the target word. The Skip-gram model on the other hand predicts the context words given the target word. In this research, we use the skip-gram model owing to the fact that skip-gram model works well for medium sized datasets [Mikolov and Dean 2013].

### **3.5 Random Forest**

Random forest is an ensemble learning method for classification and regression tasks. Random forest operates by creating a multitude of decision trees at the training time. A new input vector is given to each of the trees that were generated during the training time. Each tree produces a classification result for that vector, and the class with the highest frequency is assigned to the input vector. Random forests correct for decision trees habit of overfitting the data. In this work, we use the implementation on Random Forest provided by Weka [Hall et al. 2009].

# 4

## Approach

This thesis proposes an approach to identify tweets with implicit entity mentions of a given entity type. The problem is defined as:

**Given the tweet  $w$  and a set of interested entity types  $T$  (e.g. MOVIE, BOOK, ACTOR), recognize whether  $w$  has implicit mention of entity of type  $t \in T$ .**

We formally define implicit entities as follows:

*Implicit Entity is an entity mentioned in a tweet where its name is not present nor it is a synonym/alias/abbreviation of an entity name or a co-reference of an explicitly mentioned entity in the tweet [Perera et al. 2016].*

The problem of identifying tweets with implicit entity mentions can be broken down into two sub problems as shown in Figure 4.1. The first sub-problem aims to identify tweets with potential implicit entities of a given entity type. As shown in Figure 4.1 the first step identifies tweets with potential implicit entity mentions of the entity type MOVIE. The second sub-problem analyses the tweets selected by the the solution implemented to solve the first sub-problem to determine the tweets with implicit entity mentions of given type. In Figure 4.1 the first tweet has an implicit entity mention of type MOVIE while the second and the third tweet have no implicit entity mentions. In other words, the first task aims to improve the recall while the second task aims to improve the precision. We solve the first sub-problem by identifying semantic cues that

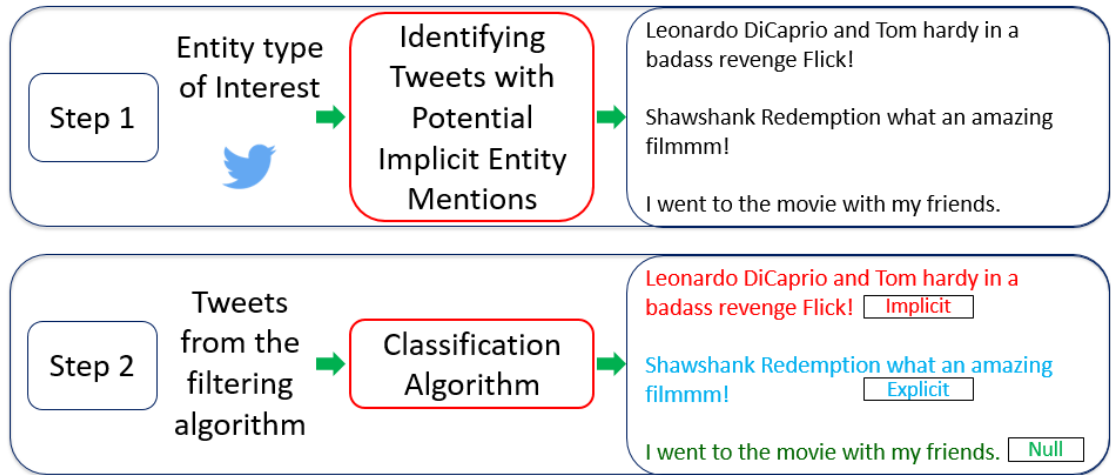


Figure 4.1: Approach Overview

indicate the presence of entities of given type in a tweet. The second sub-problem is formulated as a classification task. It uses the syntactic features extracted from the tweet and background knowledge extracted from crowd-sourced knowledge bases to identify the presence of an implicit entity mentions in tweets.

## 4.1 Identifying tweets with potential implicit entity mentions

It is observed that when referring to an entity implicitly, people almost always use one or more terms that indicate the entity type (e.g. MOVIE, BOOK). We term these indicators as semantic cues. In other words, semantic cues are terms which indicate the presence of a potential implicit entity in a particular tweet. The first three tweets in the Table 4.1 have mentions which are manifested by the semantic cues ‘thriller’, ‘flick’ and ‘movieee’. In order to identify the tweets with potential implicit mention of entities, we developed a technique that can exhaustively find the semantic cues of a given entity type. There are two types of semantic cues. The first type of semantic cues are rather common in spoken and written language and one would be able to find them out from a standard English dictionary. The terms like ‘thriller’, ‘movie’, and ‘action’ are examples of these type of semantic cues for movies. The second type of semantic cues are not common in standard English writing but rather common in the language used in social media. They are either syntactic

No	Tweet
1	leonardo dicaprio and martin scorsese reteam for serial killer thriller.
2	Relatedly, Im excited about the Jordan Belfort flick
3	I still cry like a baby at the last movieeee of Lord of the Rings
4	guardians of the galaxy is such a great movie
5	Life should be like a Steven Spielberg movie
6	Congratulations real madrid what a thriller of a final

Table 4.1: Tweets with Semantic Cues of Entity Type MOVIE

variations of the terms in above category or new terms. The terms like 'movieeee' and 'flick' are examples of these type of semantic cues for movies. It is critical to capture both these types of keywords to improve the recall of the algorithm (i.e. to capture as much as tweets with implicit entity mentions in a given corpus). We propose an approach to identify the first type of keywords from Wikipedia categories of the entities that belong to the entity type of interest and second type of keywords by analysing a tweet corpus.

#### 4.1.1 Formal Semantic Cues

To identify the formal semantic cues, we start by obtaining all the entities of the interested entity type (eg., MOVIE, BOOK) from Wikipedia. We follow this up by extracting Wikipedia categories of these entities. Once we have all the categories, we need to identify and extract the semantic cues from these categories. All the terms in the categories are not semantic cues. For example, 'films' is a semantic cue for entity type movie in the category 'American films' but not the term 'American'. It is observed that the head nouns of the categories act as semantic cues of the entity type. For example the head noun of the category '2015 films' is 'films' which is a good semantic cue for the entity type MOVIE. However, not all the head nouns can act as semantic cues for the entity type. For example, the head noun of the category 'Record progressions' is 'progressions' which is not a semantic cue for the entity type MOVIE. Hence, there is a need to find

semantic cues which are relevant to the entity type. To achieve this, we start by identifying the term which best represents the entity type. We do this by identifying the most frequent head noun. In the case of the MOVIE entity type, most frequent head noun is ‘film’ and for BOOK it is ‘book’. Once the most frequent head noun for the entity type is identified, we use semantic similarity to identify the other semantic cues.

#### 4.1.1.1 Head Nouns

As mentioned in the previous section, each Wikipedia article contains categories which describe the entity. Figure 4.2, shows the categories of the movie *Furious 7*. Most of these categories contain semantic cues of the entity type. For example the category ‘2015 films’ has the term ‘films’ and the category ‘American sequel films’ has the term ‘sequel’ and ‘film’. However, not all the terms in a category can serve as a semantic cue. For example the category ‘2015 films’, the term ‘2015’ is not a semantic cue and ‘American’ is not a semantic cue in the category ‘American sequel films’.

Categories:	<a href="#">2015 films</a>	<a href="#">English-language films</a>	<a href="#">2010s action thriller films</a>	<a href="#">American films</a>	<a href="#">American 3D films</a>	<a href="#">American action thriller films</a>	
	<a href="#">American sequel films</a>	<a href="#">Auto racing films</a>	<a href="#">Chase films</a>	<a href="#">Dolby Atmos films</a>	<a href="#">Drone films</a>	<a href="#">The Fast and the Furious</a>	<a href="#">Film scores by Brian Tyler</a>
	<a href="#">Films about amnesia</a>	<a href="#">Films about revenge</a>	<a href="#">Films about terrorism</a>	<a href="#">Films directed by James Wan</a>	<a href="#">Films set in Azerbaijan</a>	<a href="#">Films set in the Dominican Republic</a>	
	<a href="#">Films set in London</a>	<a href="#">Films set in Los Angeles, California</a>	<a href="#">Films set in Tokyo</a>	<a href="#">Films set in the United Arab Emirates</a>	<a href="#">Films shot in Abu Dhabi</a>		
	<a href="#">Films shot in Atlanta, Georgia</a>	<a href="#">Films shot in Colorado</a>	<a href="#">Films shot in Los Angeles, California</a>	<a href="#">Films shot in Tokyo</a>	<a href="#">Films shot in the United Arab Emirates</a>		
	<a href="#">Heist films</a>	<a href="#">IMAX films</a>	<a href="#">One Race Films films</a>	<a href="#">Original Film films</a>	<a href="#">Universal Pictures films</a>	<a href="#">Record progressions</a>	<a href="#">Road movies</a>
	<a href="#">Screenplays by Chris Morgan</a>						

Figure 4.2: Categories of Furious 7

Therefore, we propose identifying terms from categories which might qualify as semantic cues. It is observed that these semantic cues are the head nouns of the categories. We use Collins head noun detection technique [Collins 2003] to identify the head nouns of each of the categories. The Stanford CoreNLP API<sup>1</sup> offers an efficient implementation of this head noun detecting technique. Each noun phrase contains a head noun. A category can contain nested noun phrases and hence contain multiple head nouns. For example, consider the category ‘Film scores by Brian Taylor’ is a noun phrase with 4 nested noun phrases. The parse tree of this category is shown in Figure 4.3. Due to this fact, we use the following heuristics to identify the

<sup>1</sup><http://nlp.stanford.edu/software/corenlp.shtml>

head noun for each category:

- Identify the head noun for each of the noun phrases in the category name.
- The head noun of the longest noun phrase sequence is considered to be the head noun of the category.

With the help of these heuristics we can extract the head nouns of all the categories of an entity type. In this case, we can identify that the head noun of the category is *Film*.

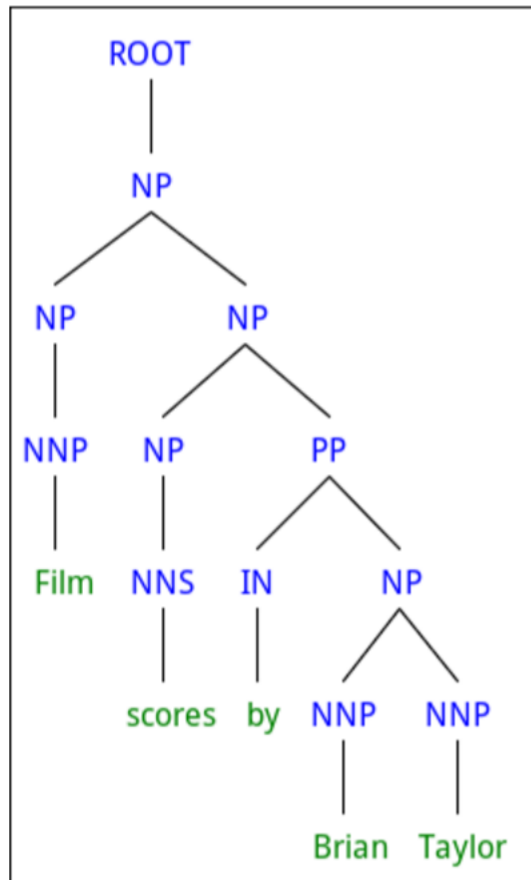


Figure 4.3: Parse Tree

However, the head nouns of all the categories are not necessarily semantic cues of the entity type. For example, the head noun of the category ‘Record progressions’ is ‘progressions’ and as we can clearly see that it is not a semantic cue of the entity type MOVIE. Hence it is important to prune the set of head nouns

identified using the above heuristics to eliminate noisy terms. To tackle this problem, we identify the term that is most relevant to the entity type (e.g. ‘film’ for MOVIE and ‘book’ for BOOK). We refer to this term as the base term. We then identify the head nouns which are semantically similar to the base term. The terms that show a similarity to the base term are considered as semantic cues.

As mentioned in the Section 3.4, semantic similarity is a metric which calculates the similarity between two words or documents on the basis of the likeness of their meaning rather than their syntax. The Skip-gram model is the state-of-the-art neural network model that generates vector representations of words which can be used to measure the semantic similarity.

We use word2vec to train a skip-gram model over the Wikipedia dump of June 2014. When training the skip gram model we set the negative sampling to 10 words which performs well for medium size corpus [Mikolov and Dean 2013]. The context window is set to 5 words, which means that it considers 5 words to the left and right of the current word [Hu et al. 2013]. The minimum word count is set to 30. The decision of using Wikipedia dump to train the model is intuitive owing to the fact that the head nouns are extracted from Wikipedia categories. Once the model is trained, we calculate the similarity of all the head nouns with the base term. The head nouns with a similarity of greater than 0.5 is selected as the semantic cues of the entity type of interest.

#### **4.1.2 Twitter Specific Semantic Cues**

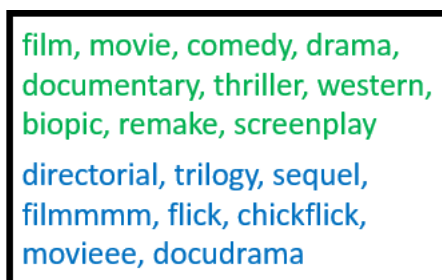
The semantic cues extracted from Wikipedia categories alone will not suffice owing to the language used on Twitter. The usage of Internet slangs, acronyms and syntactic variations are very common in Twitter. For example, the second tweet in Table 4.1 uses the semantic cue ‘flick’ while the third tweet uses the term ‘movieee’ as the semantic cue. The first term is a frequently used internet slang while the second term is just a syntactic variant of the term ‘movie’. Both these terms are good candidates to be semantic cues for the entity type MOVIE. However, the skip-gram model created using the Wikipedia dump will not capture these semantics due to the fact that they will not be used in the context of movies in Wikipedia as they are just internet slangs.

To address this issue, there is a need to identify the semantic cues with respect to the language used in Twitter. We tackle this problem in a similar way, by creating word representations for the terms in tweets. We used vectors created by Godin et al. [Godin et al. 2015] which were created using tweets collected over a period of 300 days (01/03/2013 - 28/02/2014). The vectors were generated using the skip-gram model with negative sampling, a window size of 3 and a vector size of 400 [Godin et al. 2014].

The identification of semantic cues is performed by comparing the word vectors of the words appearing in tweets with explicit mention of the entities of interest. The words that are similar to the base term are selected as semantic cues.

To filter the tweets with explicit mention of the entities of interest, we hand-picked several movies based on their popularity estimated using hits to the corresponding page on Wikipedia. The tweets were collected for a week and cleaned to remove stopwords, emoticons and punctuations and then tokenized to obtain the unigrams. The intuition behind streaming the tweets using explicit entity names is that tweets will be related to the entity type of interest. The unigrams that are extracted from these tweets are potential semantic cues and the semantic similarity of these terms are calculated with respect to the base term identified in the previous step using the skip-gram model trained over the tweets. The unigrams with a semantic similarity score of greater than 0.5 are selected as Twitter-specific semantic cues.

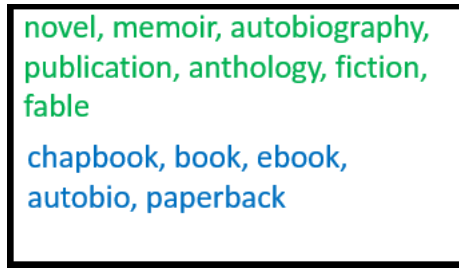
Figure 4.4 shows the list of semantic cues for MOVIE and Figure 4.5 shows the list of BOOK semantic cues. The semantic cues in blue are the formal semantic cues and the the green semantic cues are informal.



film, movie, comedy, drama,  
documentary, thriller, western,  
biopic, remake, screenplay  
directorial, trilogy, sequel,  
filmmmm, flick, chickflick,  
movieee, docudrama

Figure 4.4: Semantic Cues for MOVIE





novel, memoir, autobiography,  
publication, anthology, fiction,  
fable

chapbook, book, ebook,  
autobio, paperback

Figure 4.5: Semantic Cues for BOOK

## 4.2 Classifying Tweets as Implicit, Explicit and Null

The tweets that contain the semantic cues extracted in the previous step do not always contain implicit entity mentions. In the table Table 4.1, all the tweets contain semantic cues for the entity type MOVIE. However, only the first three tweets contain implicit entity mentions of type MOVIE while the fourth tweet has a reference to an explicit entity of the type MOVIE and the last two tweets have no entity mentions of the entity type MOVIE. As a consequence the next task is to identify tweets with implicit entity mentions. We define this as a classification problem. The first three tweets should be classified into the Implicit class as they contain implicit entity mentions of the entity type MOVIE. The fourth tweet is classified into the Explicit class as it contains explicit entity mentions of the entity type MOVIE. The last two tweets can be classified into the null class as they contain no entity mention of the entity type MOVIE. To solve this classification problem, we use the syntactic features that are derived from the tweets and the knowledge extracted from crowd sourced knowledge bases like Wikipedia and DBpedia. We use the following five features to perform this classification task:

### 4.2.1 Domain Relevant Entities

Humans make use of knowledge about the entity when referring to an implicit entity. This knowledge can comprise of entities which are of high importance to the entity type. The presence of highly relevant entities to the movie domain in tweets is a strong indicator towards the presence of an implicit mention of movie.

For example, in the first tweet, there are explicit references to *Leonardo DiCaprio* an ACTOR and *Martin Scorsese* a DIRECTOR while the second tweet has an explicit reference to *Jordan Belfort* a PERSON who has been portrayed in a movie and these tweets have implicit mention of movies. Such entities are termed as Domain Relevant Entities. Domain Relevant Entities are entities that can be used to indicate the presence of an implicit entity mention. Intuitively, we can say that *Leonardo DiCaprio* may be considered as a domain relevant entity of the entity type MOVIE on the account that he is an ACTOR, whereas *United States* which is a country, may not be considered as a domain relevant entity. We use Wikipedia and DBpedia, two crowd-sourced knowledge bases to identify the domain relevant entities.

#### 4.2.1.1 Domain Relevant Entities Using DBpedia

As explained in the Section 3.2, DBpedia is a knowledge base automatically extracted from Wikipedia. DBpedia is rich in relationships which are expressed in the form of RDF triples.

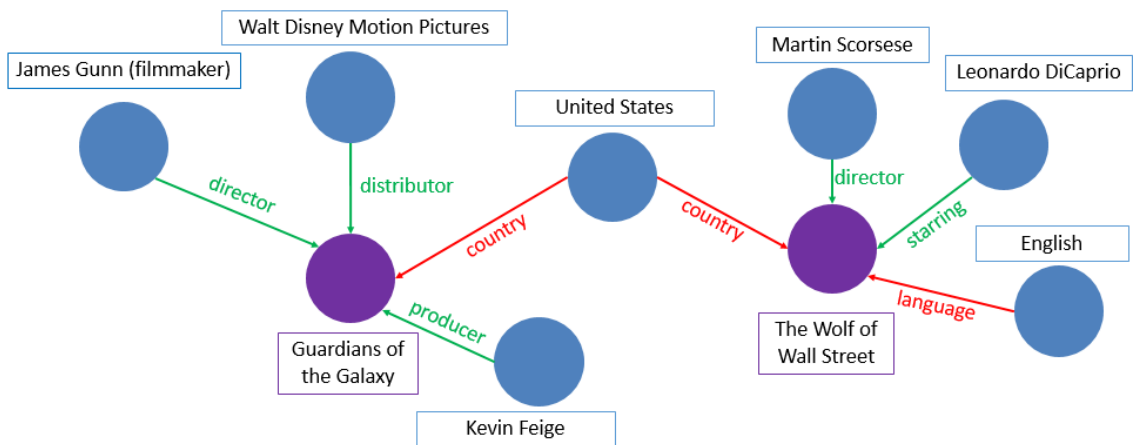


Figure 4.6: DBpedia Subgraph

For a set of entities  $S$  of an interested type  $t$  we extract all triples that have at least one of the entities in  $S$  as a subject or an object [Perera et al. 2016]. However, as shown in Figure 4.6 for a given entity type not all relationships are relevant in order to find domain relevant entities. For example, a movie has relationships ‘director’ and ‘starring’ as well as ‘language’ and ‘country’. The former two relationships are more important

when describing a movie than the latter. From the above example we can say that *United States of America* is not a domain relevant entity for the entity type MOVIE as it is connected using the relationship ‘country’. We capture this intuition by ranking the relationships based on a joint probability [Perera et al. 2016].

$$P(r, t) = \frac{\text{no. of relationships of } r \text{ with instances of type } t}{\text{total number of relationships of } r} \quad (4.1)$$

where  $r$  is the relationship. The relationships of all the entities in  $S$  of type  $t$  that have one of the top  $m$  relationships from the ranked set of relationships for  $t$  contribute to identifying the domain relevant entities. We collect the `rdfs:label` value of the entities connected to at least one entity from  $S$  through the selected relationships as the domain relevant entities. We use the following method to identify domain relevant entities in tweets.

#### 4.2.1.2 Commonness

Entity linking in social media is a challenging task. One of the strong baseline for this task is assign the most common entity that is being referred by the terms. For example, if the term *Chicago* appears in the text, it can refer to *Chicago City*, *Chicago* movie, or even the basketball team. However, it is found that the term *Chicago* is more frequently being used to refer to *Chicago* city. Hence, just linking all occurrences of term *Chicago* to *Chicago* city produce strong results in entity linking task. This idea is captured by the commonness measure. Hence, we deploy a simple entity spotting mechanism by leveraging the anchor texts in Wikipedia. Anchor text is a visible clickable link in a hyperlink<sup>2</sup>.

We start by extracting all the anchor texts from Wikipedia and the corresponding pages that are linked using them. We generate the statistics for all these internal links. For example, the anchor text *Guardians of the Galaxy* occurs 7596 times on Wikipedia; out of which 5000 times it is linked to the movie while 2596 times it is linked to the cartoon.

*‘leonardo dicaprio and martin scorsese reteam for serial killer thriller.’*

Consider the above mentioned tweet. We segment the tweet recursively, and check if any of the  $n$ -grams is an anchor text in Wikipedia. We start by segmenting the tweet into larger chunks (5-grams) and then proceed

<sup>2</sup>[https://en.wikipedia.org/wiki/Anchor\\_text](https://en.wikipedia.org/wiki/Anchor_text)

to smaller chunks till we reach unigrams. We observe that, in tweets that the proper upper bound for segmentation is 5. When we encounter an anchor text  $a$  (eg., leonardo dicaprio, martin scorsese) in a tweet we consider the prior probability that  $e$  is the target of a link with anchor text  $a$  in Wikipedia. In simple terms, we calculate the probability of *leonardo dicaprio* being linked to ‘https://en.wikipedia.org/wiki/Leonardo\_DiCaprio’ and *martin scorsese* being linked to https://en.wikipedia.org/wiki/Martin\_Scorsese. This is known as the commonness [28] and is defined as:

$$\text{commonness}(e, a) = \frac{|L_{a,e}|}{\sum_{a'} |L_{a,e'}|}$$

where  $e$  is the domain relevant entity,  $a$  is the anchor text (spotted text), and  $|L_{a,e}|$  denotes the number of times anchor text  $a$  linked to entity  $e$ . Once a n-gram is spotted as a domain relevant entity, we do not further consider it in the segmentation. For example, once we identify that *leonardo dicaprio* is a domain relevant entity we do not further check if ‘leonardo’ and ‘dicaprio’ are anchor texts.

#### 4.2.1.3 Domain Relevant Entities Using Wikipedia

DBpedia is a well structured and widely used knowledge base. However, DBpedia suffers from a couple of drawbacks as shown in Figure 4.7. The first drawback is the lack of coverage. For example, the entity *Guardians of the Galaxy* does not have a single starring relationship in DBpedia. This problem is prevalent across a large number of entities in DBpedia. The second drawback is that DBpedia does not have all the entities. For example, the entity *Jordan Belfort* which is a domain relevant entity for movies is not present in DBpedia. To address these issues, we use Wikipedia which is a highly comprehensive knowledge base.

Wikipedia is crowd-sourced and unstructured knowledge base. As explained in Section 3.1, the hyperlink structure of Wikipedia provides links to pages that are topically relevant to the article. For example, as shown in the Figure 4.8 there is a link from the page *Guardians of the Galaxy* to *Chris Pratt* because he has starred in that movie. This hyperlink structure of Wikipedia forms a **Wikipedia Hyperlink Graph** where the nodes are Wikipedia pages and the edges represent the links between the pages.

The graph for an entity type  $t$  consists of the set of entities  $S$  of type  $t$  and the set of entities  $O(S)$  that are linked by anchor texts in the lead sections of at least one entity in  $S$ . Formally the graph for an entity type  $t$  is

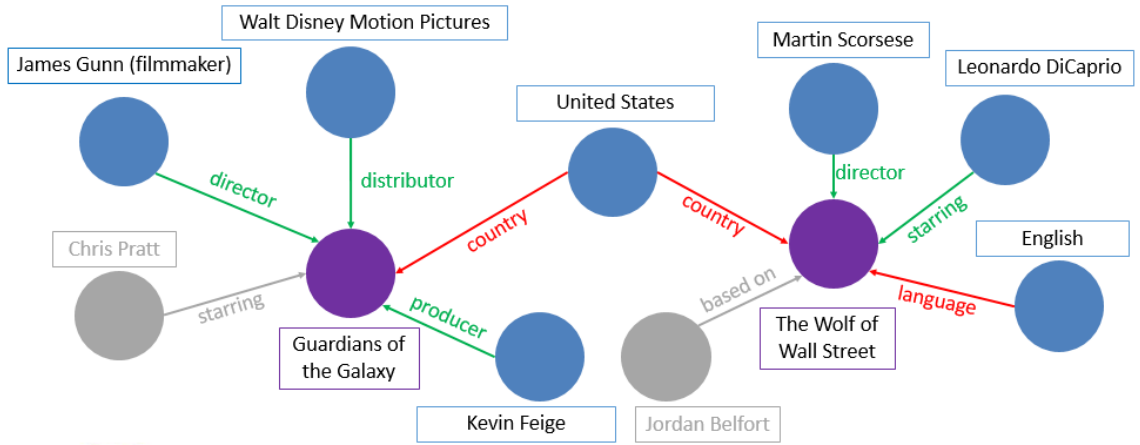


Figure 4.7: Drawbacks of DBpedia

represented as a directed graph  $G_t = (V_t, E_t)$  where the vertices  $V_t \in (S \cup O(S))$  and edges  $E_t \in V_t \times V_t$ . There is an edge from  $v_{t_i}$  to  $v_{t_j}$  if the Wikipedia page of  $v_{t_i}$  has a link to the entity  $v_{t_j}$ . An example of the subgraph is shown in Figure 4.8

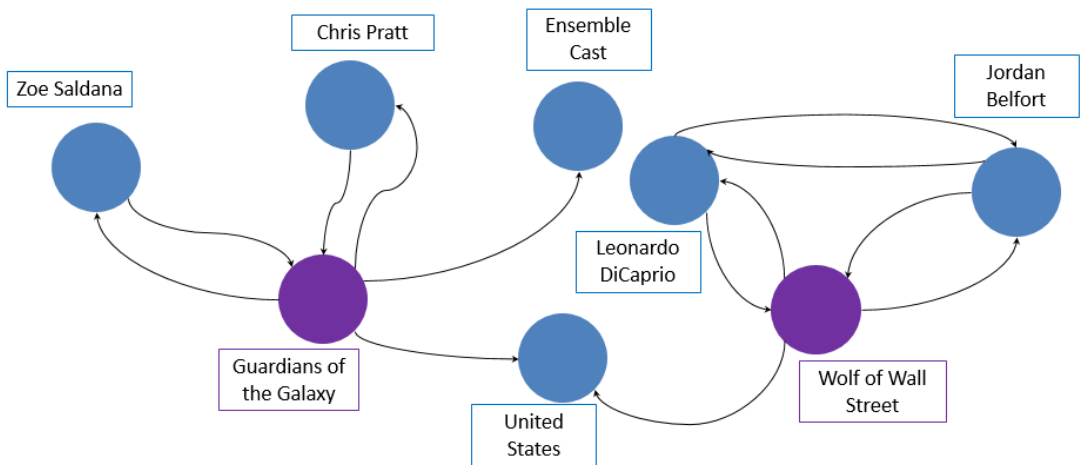


Figure 4.8: Wikipedia Hyperlink Graph

Not all the entities in the Wikipedia hyperlink graph is relevant to the entity type. *Chris Pratt* and *Zoe Saldana* are highly relevant to the entity type MOVIE due to the fact that they are ACTORS. However,

*ensemble cast* is a collection of all the actors and crew members and hence is not relevant to the entity type MOVIE. Similarly, each entity type will have a set of entities that are highly relevant to the particular entity type. On the other hand, every entity type will also have entities that have little or no relevance to the entity type. In other words, these entities do not help in inferring the presence of implicit entity mentions. Our goal is to find the degree of relevance of an entity to a particular entity type.

We use a metric called as *Normalized Outdegree*

$$\text{normalized\_outdegree}(v_i) = \frac{\sigma_{v_i}(S)}{\sigma_{v_i}}$$

where  $v_i \in O(S)$  and  $v_i \notin S$ ,  $\sigma_{v_i}$  represents the total number of outgoing edges from  $v_i$  and  $\sigma_{v_i}(S)$  represents the number of outgoing edges from  $v_i$  to an entity in  $S$ . The intuition to using this metric is that an entity which is highly relevant to an entity type  $t$  will be connected to a larger number of entities of the type  $t$ . For example, *Chris Pratt* and *Zoe Saldana* is highly relevant to the entity type MOVIE and hence has a normalized outdegree score of 1.0 while *Ensemble Cast* and *United States* are not relevant to MOVIE and hence a score of 0.0.

For domain relevant entities from Wikipedia we calculate the feature as  $\text{commonness}(e, a) \text{ outdegree}(e)$ . For an entity  $e$  and an anchor text  $a$ , the commonness score represents the probability of the anchor text  $a$  being linked to the entity  $e$  or in other words, the probability of  $e$  being a domain relevant entity while the outdegree represents the probability of entity  $e$  being relevant to the domain. The feature captures the probability of  $e$  being an entity relevant to the domain of interest. The commonness score is calculated as mention in the previous section.

### 4.2.2 Window Based Bigrams

The third tweet in the table has an implicit mention to a MOVIE entity. However, it does not contain a domain relevant entity. On the contrary it contains an explicit entity of the type MOVIE. Domain relevant entities are not the only way people express implicit entity mentions in tweets. Due to this it is essential to capture the bigrams from either sides of the semantic cues to identify implicit entity mentions. We do this by extracting

the bigrams from either side of the semantic cues that are present in the tweet.

### 4.2.3 Explicit Entity Mentions

All the above mentioned features are very critical in identifying tweets that contain implicit entity mentions. In order to identify the presence of an explicit entity mention in a tweet, we focused on identifying the presence of any explicitly mentioned entity of the interested entity type in the tweet. For example, the fourth tweet is a candidate for an implicit entity mention of the entity type MOVIE because it contains the semantic cue 'movie'. However, it does not contain an implicit entity mention but contains an explicit entity mention of the same type. We use the presence of an explicit entity of as a feature to recognize the tweets with explicit entity mentions. To spot explicit entity mention, we use the commonness metric described in the previous section.

### 4.2.4 Part-Of-Speech Tags

The presence of domain relevant entities does not always indicate the presence of an implicit entity mention. For example, the fifth tweet has the mention of *Steven Spielberg* a director which is relevant to the entity type MOVIE. However, if we observe the tweet carefully we can deduce that the tweet does not have an implicit or an explicit entity mention of the entity type MOVIE. To capture this, we used Part-Of-Speech tags of words before the semantic cues as feature. 'adjectives', 'adverbs', 'determiners' (the) and 'verbs' are positive indicators of an implicit entity mention; however 'determiners' (a, an), 'prepositions' are strong indicators of the null category. If there is an entity before the semantic cue then we use the POS tag of the term before the start of the entity. Like in this example, the POS tag is that of the term 'a' which is 'determiner'. If the semantic cue appears at the start of the tweet we set the POS tag to be 'null'.

### **4.3 Classifying Tweets into Predefined Type**

The tweets which contain the semantic cues for the entity type of interest  $t$  are given to the classification step as a vector of features. The tweets are cleaned before the feature extraction step. We remove the punctuations, emoticons and normalize the numbers to the pseudo-string 'NUMBER' in tweets. The hashtags and username mentions that are written in camelcase are retained after decomposing (@VeronicaRoth Veronica Roth, #markWahlberg mark Wahlberg) and the others are retained by removing '#' and '@' symbols. Once the feature extraction is done, we train a Random Forest algorithm. The task of classification can be visualized as a decision tree. Random Forest is a collection of multiple decision trees which solves the problem of overfitting. We use the open source tool Weka [Hall et al. 2009] to train and test the datasets.



# 5

## Evaluation

In this chapter, we evaluate both the steps involved in identifying tweets with implicit entity mentions.

### 5.1 Semantic Cue Evaluation

In this section we evaluate the precision of the semantic cues in identifying tweets with potential implicit entity mentions. We define precision as the amount of topically relevant tweets that are extracted using a particular semantic cue.

*'Will Poulter joins Kathryn Bigelow's Detroit riots drama'*

*'Math is a drama queen. It cant have that many problems'*

Both the tweets mentioned above have the semantic cue 'drama' which is relevant to the entity type MOVIE. However, the first tweet is topically relevant to MOVIE while the second tweet is not.

To calculate the precision of the semantic cues, we collect random samples of tweets from Twitter using the Twitter Streaming API<sup>1</sup>. We collect 100 tweets for each semantic cue which has a similarity score of greater than 0.5. After collecting all these tweets, we check the number of topically relevant tweets for each semantic cue.

As we can see from the figure Figure 5.2 and Figure 5.1 that the similarity score and the topical relevance

---

<sup>1</sup><https://dev.twitter.com/streaming/overview>

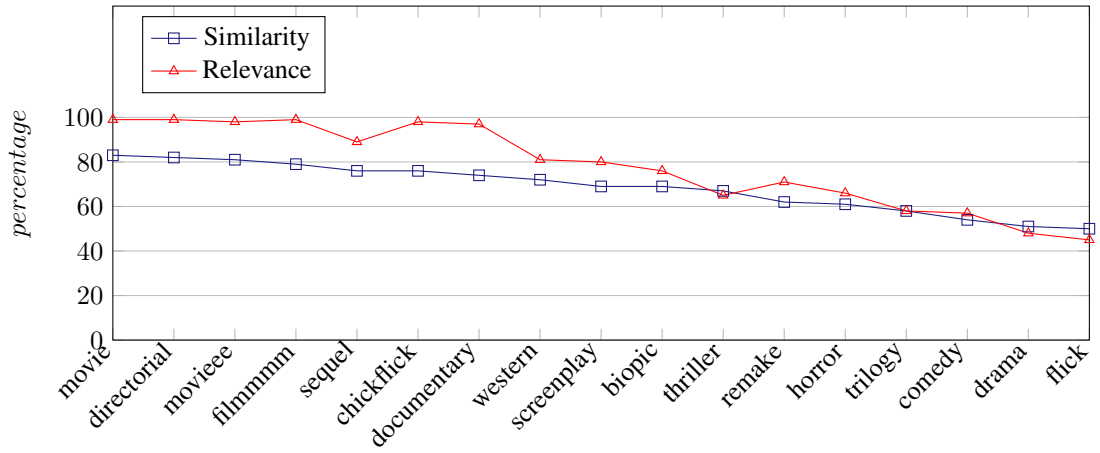


Figure 5.1: Semantic Cue Evaluation for Movies

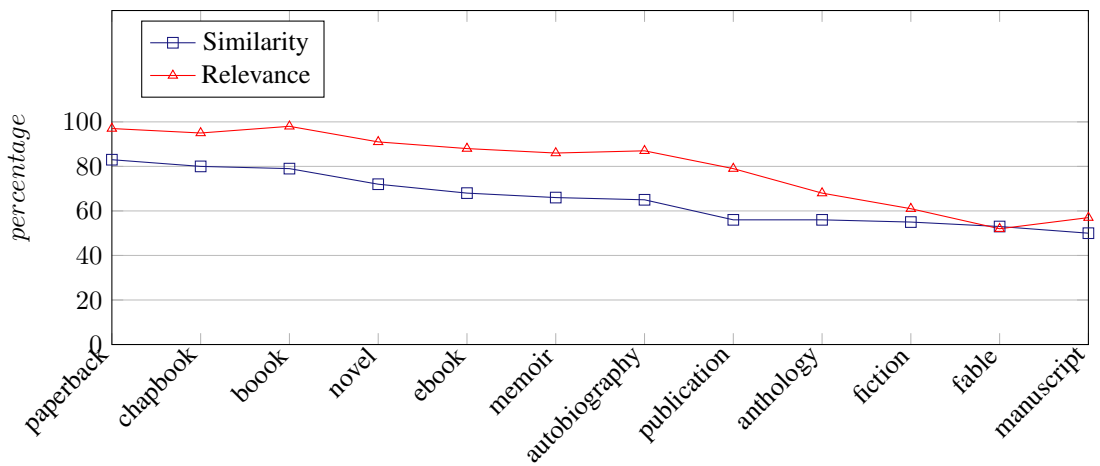


Figure 5.2: Semantic Cue Evaluation for Books

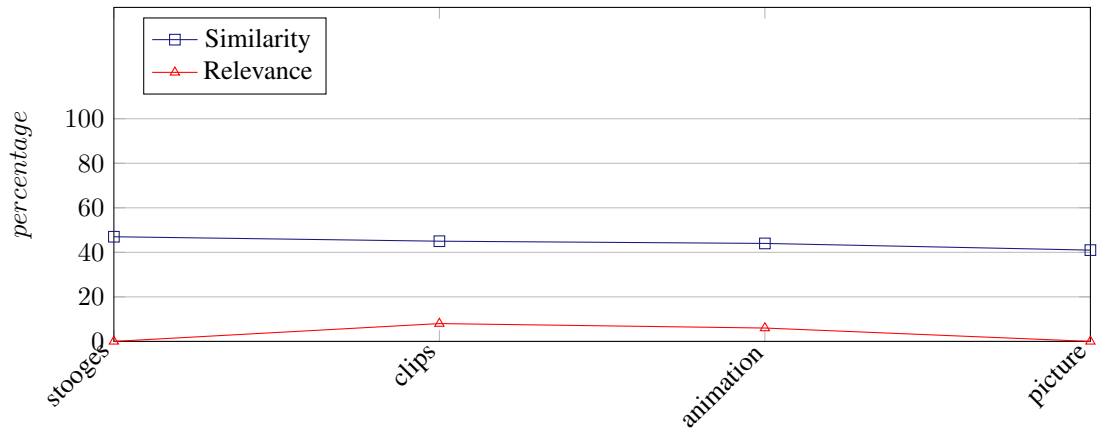


Figure 5.3: Semantic Cue Below Similarity of 0.5 for Movies

of the tweets are directly proportional to each other. As it is evident that the as the similarity score and the topical relevance of the tweets are directly proportional to each other. We do not select the semantic cues which have similarity scores of less than 0.5 as these semantic cues attract a lot of noisy tweets which do not have any topical relevance to the entity type. As shown in Figure 5.3 and Figure 5.4 the semantic cues with a similarity score of less than 0.5 do not contain topically relevant tweets and hence are not good candidates for identifying tweets with potential implicit entity mentions.

## 5.2 Classifying Tweets as Implicit, Explicit and Null

In this section, we evaluate the performance of model created on the classification step.

### 5.2.1 Dataset

There is no annotated gold standard dataset available for this task since it is a novel task introduced by our previous work [Perera et al. 2016]. Hence, we created a two separate gold standard datasets by collecting tweets for two entity types: BOOK, MOVIE using their semantic cues. The first dataset is created using the base terms and the most relevant term of each entity type to create the dataset ('movie' and 'film' for MOVIE and 'book' and 'novel' for BOOK). The tweets were collected during the time period of August 2014 using

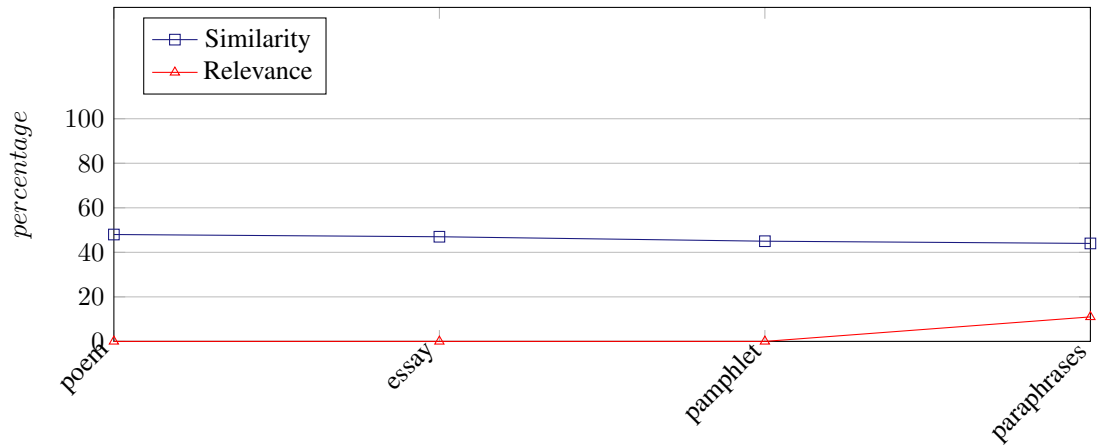


Figure 5.4: Semantic Cue Below Similarity of 0.5 for Books

Entity Type	Explicit	Implicit	Null
Movie	420	179	385
Book	160	199	311

Table 5.1: Dataset 1 Statistics

the Twitter Streaming API. The tweets were annotated as Explicit, Implicit and Null by two annotators. The tweets that have a mutual agreement between the annotators were the only ones that were added to the gold standard dataset. Table 5.1 shows the characteristics of the first gold standard dataset.

The second gold standard dataset is created using all the semantic cues apart from the ones used to create the previous dataset. Table 5.2 shows the characteristics of the second gold standard dataset. The goal of this evaluation is to demonstrate that the performance of the model does not depend on the semantic cues used to create the dataset.

Entity Type	Explicit	Implicit	Null
Movie	123	96	141
Book	135	77	122

Table 5.2: Dataset 2 Statistics

The main ingredient for our classification step is the knowledge extracted from Wikipedia and DBpedia. We use Wikipedia and DBpedia snapshots of July 2014 as this helps us set up a realistic evaluation environment since we have restricted knowledge of what would have been available at the time period of our corresponding evaluation.

### 5.2.2 Evaluation Metrics

We use standard evaluation metrics of Precision, Recall and F-Measure to evaluate the classification step.

Precision is the fraction of retrieved instances that are relevant.

$$Precision = \frac{|\{relevant\_tweets\} \cap \{retrieved\_tweets\}|}{|\{retrieved\_tweets\}|}$$

Recall is the fraction of relevant instances that are retrieved.

$$Recall = \frac{|\{relevant\_tweets\} \cap \{retrieved\_tweets\}|}{|\{relevant\_tweets\}|}$$

F-Measure is the harmonic mean of precision and recall.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

### 5.2.3 Results of Classification

We evaluate the classification step using 5-fold cross validation and report the results in this section. We use all the features mentioned in the previous chapter to perform the classification step. Table 5.3 reports the results of the classification step for the entity type Books while Table 5.4 shows the results for the entity type Movies.

We further evaluate the second dataset using the model created using the first set of tweets. Table 5.5 shows the results of the classification step for the entity type MOVIE and Table 5.6 shows the results of the classification step for the entity type BOOK. Although, the training and the test data-sets are created using tweets with different semantic cues the results prove that this does not affect the model. This is due the fact that we do not use the semantic cues as the features for classification.

	Precision	Recall	F Score
Explicit	0.796	0.756	0.775
Implicit	0.647	0.602	0.624
Null	0.721	0.765	0.683

Table 5.3: Classification Results for Books on First Dataset

	Precision	Recall	F Score
Explicit	0.725	0.648	0.684
Implicit	0.605	0.593	0.605
Null	0.642	0.729	0.683

Table 5.4: Classification Results for Movies on First Dataset

	Precision	Recall	F Score
Explicit	0.702	0.689	0.695
Implicit	0.675	0.634	0.654
Null	0.699	0.729	0.714

Table 5.5: Classification Results for Movies on Second Dataset

	Precision	Recall	F Score
Explicit	0.757	0.768	0.762
Implicit	0.662	0.617	0.639
Null	0.715	0.747	0.731

Table 5.6: Classification Results for Books on Second Dataset

### 5.2.3.1 Error Analysis

The errors in the classification step can be primarily attributed to these two factors

- **Errors due to insufficient knowledge:** Wikipedia and DBpedia are rich sources of knowledge which has been used extensively for a variety of applications. However, in our case this knowledge is not always enough to solve the classification problem. For example, consider the following tweet '*ISRO sends probe to Mars for less money than it takes Hollywood movie to send a woman to space*'. This movie has an implicit entity mention of the MOVIE *Gravity*. However, it does not contain a relevant entity to the entity type MOVIE. To deduce the presence of the implicit entity mention in this particular tweet, we need to discover the relationship between the daily entities and events . In this case we need to identify the relationship between the budget of the spacecraft launched by *Indian Space Research Organisation (ISRO)* and the movie *Gravity*. However, it is impossible to find such knowledge on Wikipedia or DBpedia.
- **Errors due to domain relevant entities of a different entity type:** Sometimes, implicit entity mentions of a particular type are manifested by domain relevant entities of a different entity type. For example, consider the tweet '*What a great book, Jamie Blackley is a babe*'. This tweet contains an implicit reference to the entity *If I Stay* which is the entity type BOOKS. However, this entity is referred to by using the domain relevant entity *Jamie Blackley* which is relevant to the entity type MOVIE and not the entity type BOOK. However, this phenomenon is not frequent as the dataset contains only a few tweets.

## 5.3 Discussion

In this section, we discuss the effect of number of relationships used to select the domain relevant entities from DBpedia, on the classification step. Finally, we study the impact of the knowledge extracted from the two knowledge sources on the classification step.

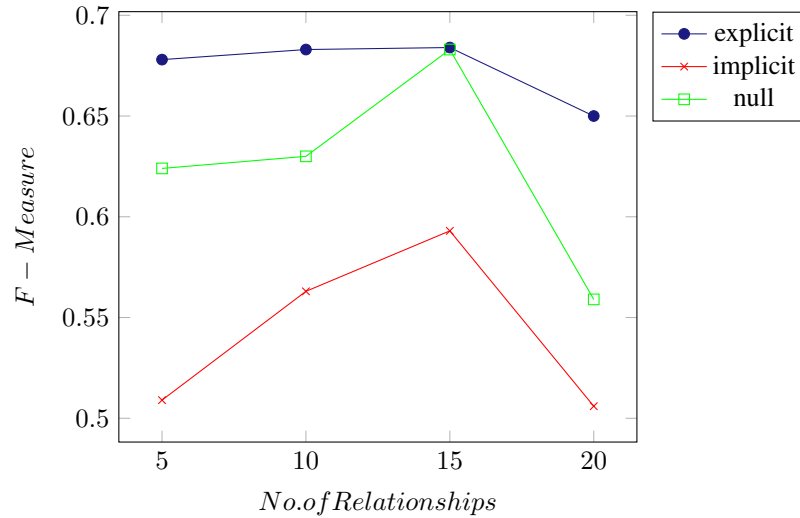


Figure 5.5: Impact of Relationships on Classification - Movie

### 5.3.1 Impact of Relationships on the Classification Step

The number of top- $k$  relationships that are selected for identifying domain relevant entities have a huge impact on the classification step. The number of domain relevant entities is directly proportional to number of top- $k$  relationships we select from DBpedia. Fig Figure 5.5 and Figure 5.6 shows the F score of all the three classes for the entity types MOVIE and ACTOR.

As it is clearly evident that scores are the lowest when  $k$  is equal to 5. The F score keeps on increasing as we increase the number of relationships, but after 15 the scores go down. As we increase the number of relationships from 5 to 15 the number of entities that are relevant to the entity type keep on increasing. However, after 15 the entities that are captured are no longer relevant to the entity type and hence has a negative impact on the classification step.

For example, when the value of  $k$  is 5 *John Green* who is an author and highly relevant to the entity type BOOK is not considered as a domain relevant entity because the relationship author is not present in the top-5 relationships. On the other hand when we select the top-20 relationships the entity *United States of America* is considered as a domain relevant entity because of the relationship country. However, *United*



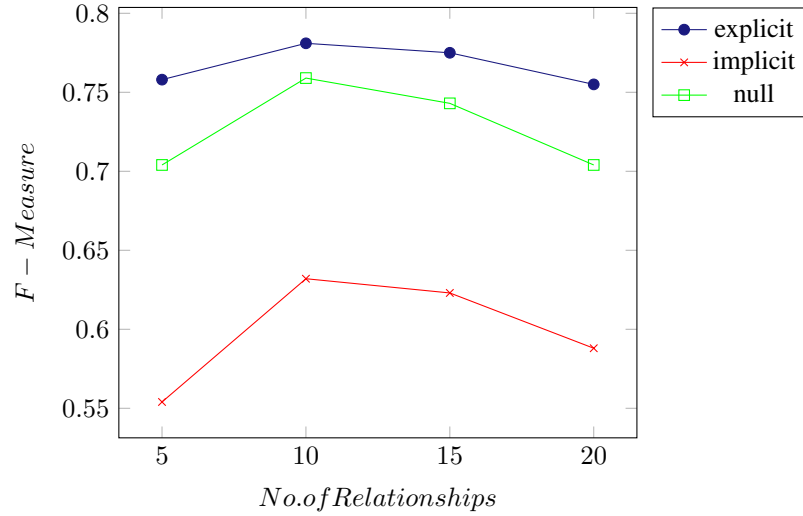


Figure 5.6: Impact of Relationships on Classification - Book

*States of America* is a very generic entity and cannot be considered as a domain relevant entity for the entity type BOOK.

### 5.3.2 Impact of Knowledge on the Classification Step

We use knowledge extracted from two different knowledge sources to perform the classification task. In this section, we study the impact of each of these knowledge sources individually. Figure 5.7 and Figure 5.8 demonstrates the impact of the knowledge extracted from Wikipedia on the classification step for both MOVIE and BOOK. Figure 5.9 and Figure 5.10 shows the impact DBpedia knowledge on classifying the tweets for entity types MOVIE and BOOK. It is evident from the graphs that the by using knowledge extracted from one knowledge base is insufficient to perform the classification.

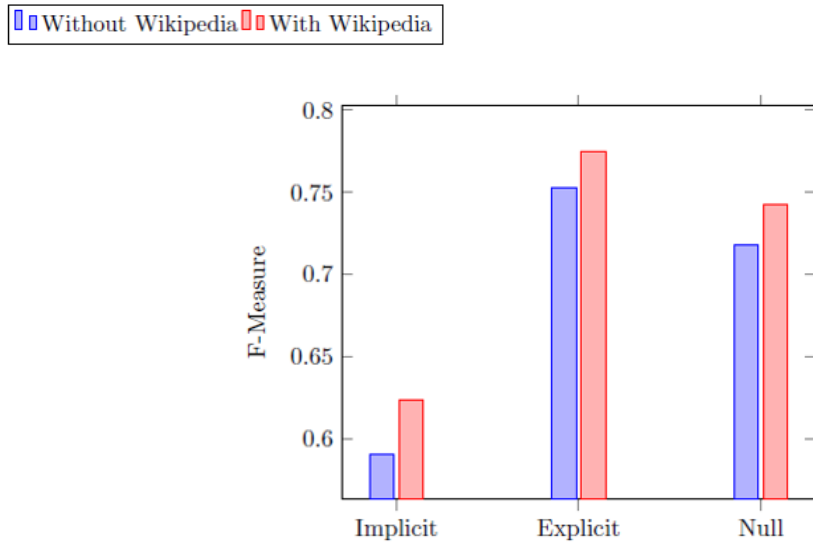


Figure 5.7: Impact of Wikipedia Knowledge on Books

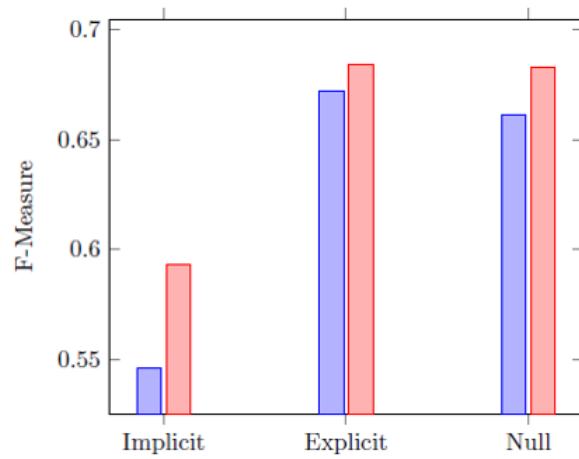


Figure 5.8: Impact of Wikipedia Knowledge on Movies

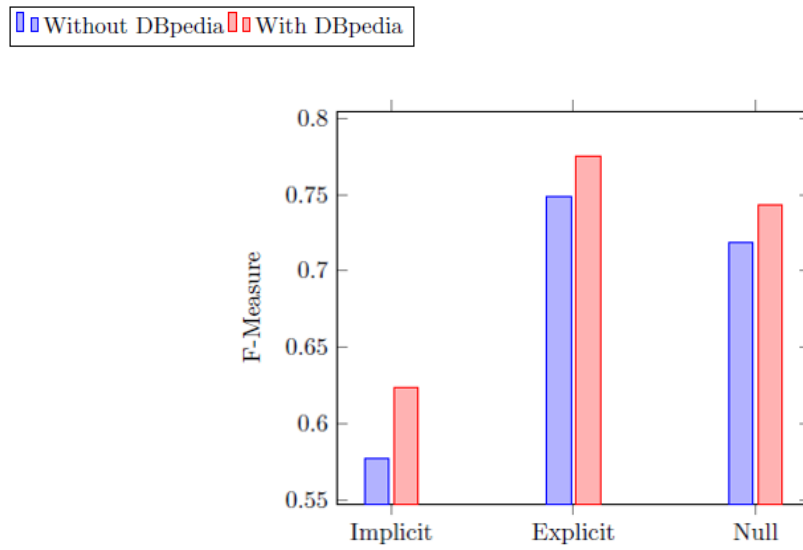


Figure 5.9: Impact of DBpedia Knowledge on Books

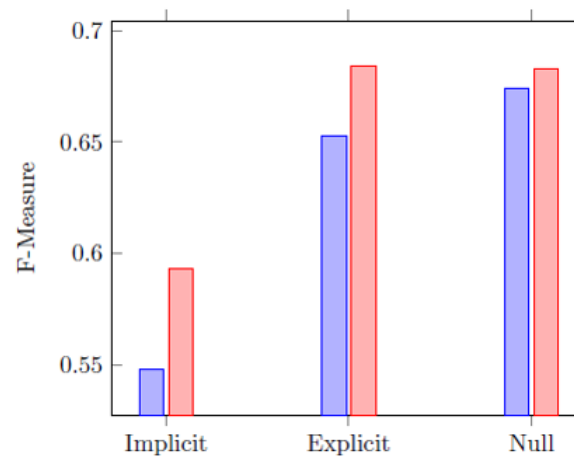


Figure 5.10: Impact of DBpedia Knowledge on Movies

# 6

## Conclusion and Future Work

In this thesis, we presented a novel approach to identify tweets with implicit entity mentions given an entity type. We introduced the concept of semantic cues which capture the occurrences of a given entity type in tweets. We created a gold standard dataset for two entity types namely MOVIE's and BOOK's to foster future research. We also investigated the impact of knowledge harvested from multiple knowledge bases (Wikipedia and DBpedia) on the task of identifying tweets with implicit entity mentions.

In the future, we will investigate approaches to identify text segments which indicate the presence of implicit entity mentions in tweets. We also plan to incorporate knowledge extracted from domain specific knowledge bases (eg., Rotten Tomatoes for MOVIE's, Goodreads for BOOK's) to improve the performance of the algorithm. We would also work towards incorporating more knowledge driven features to further enhance the approach.

# References

- ASAHARA, M. AND MATSUMOTO, Y. 2003. Japanese named entity extraction with redundant morphological analysis. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, 8–15.
- 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.
- CHENG, Z., CAVERLEE, J., AND LEE, K. 2010. You are where you tweet: a content-based approach to geolocating twitter users. In *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, 759–768.
- CHERNOV, S., IOFCIU, T., NEJDL, W., AND ZHOU, X. 2006. Extracting semantics relationships between wikipedia categories. *SemWiki 206*.
- COLLINS, M. 2003. Head-driven statistical models for natural language parsing. *Computational linguistics* 29, 4, 589–637.
- DUMAIS, S. T. 2004. Latent semantic analysis. *Annual review of information science and technology* 38, 1, 188–230.
- EFRON, M. 2011. Information search and retrieval in microblogs. *Journal of the American Society for Information Science and Technology* 62, 6, 996–1008.

- GABRILOVICH, E. AND MARKOVITCH, S. 2006. Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In *AAAI*. Vol. 6. 1301–1306.
- GENC, Y., SAKAMOTO, Y., AND NICKERSON, J. V. 2011. Discovering context: classifying tweets through a semantic transform based on wikipedia. In *International Conference on Foundations of Augmented Cognition*. Springer, 484–492.
- GODIN, F., VANDERSMISSEN, B., DE NEVE, W., AND VAN DE WALLE, R. 2015. Multimedia lab@ acl w-nut ner shared task: Named entity recognition for twitter microposts using distributed word representations. *ACL-IJCNLP 2015*, 146.
- GODIN, F., VANDERSMISSEN, B., JALALVAND, A., DE NEVE, W., AND VAN DE WALLE, R. 2014. Alleviating manual feature engineering for part-of-speech tagging of twitter microposts using distributed word representations. In *Workshop on Modern Machine Learning and Natural Language Processing, NIPS*.
- GRUHL, D., NAGARAJAN, M., PIEPER, J., ROBSON, C., AND SHETH, A. 2009. Context and domain knowledge enhanced entity spotting in informal text. In *International Semantic Web Conference*. Springer, 260–276.
- HALL, M., FRANK, E., HOLMES, G., PFAHRINGER, B., REUTEMANN, P., AND WITTEN, I. H. 2009. The weka data mining software: an update. *ACM SIGKDD explorations newsletter 11*, 1, 10–18.
- HU, X., ZHANG, X., LU, C., PARK, E. K., AND ZHOU, X. 2009. Exploiting wikipedia as external knowledge for document clustering. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 389–396.
- HU, Y., TALAMADUPULA, K., KAMBHAMPATI, S., ET AL. 2013. Dude, srsly?: The surprisingly formal nature of twitter’s language. In *ICWSM*.
- JAVA, A., SONG, X., FININ, T., AND TSENG, B. 2007. Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*. ACM, 56–65.

- KAPANIPATHI, P., JAIN, P., VENKATARAMANI, C., AND SHETH, A. 2014. User interests identification on twitter using a hierarchical knowledge base. In *European Semantic Web Conference*. Springer, 99–113.
- KRISHNAMURTHY, R. 2015. Knowledge enabled location prediction of twitter users. Ph.D. thesis, Wright State University.
- LEACOCK, C. AND CHODOROW, M. 1998. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database* 49, 2, 265–283.
- LESK, M. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*. ACM, 24–26.
- LIU, X., ZHANG, S., WEI, F., AND ZHOU, M. 2011. Recognizing named entities in tweets. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 359–367.
- MCCALLUM, A. AND LI, W. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, 188–191.
- MIHALCEA, R., CORLEY, C., AND STRAPPARAVA, C. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI*. Vol. 6. 775–780.
- MIKOLOV, T. AND DEAN, J. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*.
- MUKHERJEE, S. AND BHATTACHARYYA, P. 2012. Wikisent: Weakly supervised sentiment analysis through extractive summarization with wikipedia. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 774–793.
- NADEAU, D. AND SEKINE, S. 2007. A survey of named entity recognition and classification. *Linguisticae Investigationes* 30, 1, 3–26.

- PANG, B. AND LEE, L. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval* 2, 1-2, 1–135.
- PERERA, S., HENSON, C., THIRUNARAYAN, K., SHETH, A., AND NAIR, S. 2014. Semantics driven approach for knowledge acquisition from emrs. *IEEE journal of biomedical and health informatics* 18, 2, 515–524.
- PERERA, S., MENDES, P. N., ALEX, A., SHETH, A. P., AND THIRUNARAYAN, K. 2016. Implicit entity linking in tweets. In *International Semantic Web Conference*. Springer, 118–132.
- RAO, D., MCNAMEE, P., AND DREDZE, M. 2013. Entity linking: Finding extracted entities in a knowledge base. In *Multi-source, multilingual information extraction and summarization*. Springer, 93–115.
- RICHARDSON, R., SMEATON, A., AND MURPHY, J. 1994. Using wordnet as a knowledge base for measuring semantic similarity between words.
- RITTER, A., CLARK, S., ETZIONI, O., ET AL. 2011. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1524–1534.
- RITTER, A., ETZIONI, O., CLARK, S., ET AL. 2012. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1104–1112.
- SAKAKI, T., OKAZAKI, M., AND MATSUO, Y. 2010. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*. ACM, 851–860.
- SHINYAMA, Y. AND SEKINE, S. 2004. Named entity discovery using comparable news articles. In *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, 848.
- SONI, S. 2015. Domain specific document retrieval framework on near real-time social health data. Ph.D. thesis, Wright State University.



- TJONG KIM SANG, E. F. AND DE MEULDER, F. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, 142–147.
- WANG, P. AND DOMENICONI, C. 2009. Towards a universal text classifier: Transfer learning using encyclopedic knowledge. In *2009 IEEE International Conference on Data Mining Workshops*. IEEE, 435–440.
- WITTEN, I. AND MILNE, D. 2008. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy*, AAAI Press, Chicago, USA. 25–30.
- WU, Z. AND PALMER, M. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 133–138.
- ZHOU, D., CHEN, L., AND HE, Y. 2011. A simple bayesian modelling approach to event extraction from twitter. *Atlantis*, 0.