Generating category-specific entity embeddings for populating Knowledge Graphs

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science in ***Computer Science and Engineering*** *by Research*

by

Gokul Thota
2019111009
gokul.vamsi@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
March 2024

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "**Generating category-specific entity embeddings for populating Knowledge Graphs**" by **Gokul Thota**, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Advisor: Prof. Vasudeva Varma

To *my family*

Acknowledgments

The number of life "chapters" I have learned here at IIIT-H is way more than the number of chapters I have discussed in this thesis. The scared kid who joined this college under a massive amount of self-doubt about "research" did turn out fine after all! This would not have been possible without the many people who were along with me on this journey, and I'm incredibly grateful to each one of them.

First and foremost, I would like to express my immense gratitude to Prof. Vasudeva Varma, who has always tried to support me and motivated me to challenge myself, right from the first time I met him online to discuss joining the iREL. His invaluable insights regarding my work, and kind words of guidance, have always inspired me to push myself. He has always tried to steer me in the right direction and provided really helpful suggestions whenever I was stuck. Thank you very much, sir, for the countless things I have learned from you. I'm very grateful for the opportunity provided by you to work at iREL, where I've got to meet so many amazing people.

I'm very thankful to all the lab members who played an important role in my research journey here. Firstly, I'd like to thank Bhavyajeet, who was someone I could always count on to discuss any of my doubts or difficulties, regarding work, plans for the future, etc. I've always ended up learning a lot and feeling positive after our discussions. Thank you Himanshu sir, who was the first person I discussed joining iREL, and who has provided me with support and guidance whenever I've asked for it. Thank you Anubhav sir, whom I've known since I was performing in Freshers as a singer. Thank you Sagar, I've admired your patience and helpfulness. Sorry for pestering you with many doubts during the NLP course. Thank you Tushar sir, whom I was frequently in touch with during my work in IndicWiki, your guidance has helped me a lot. Thanks to my peers Nirmal, Shivansh, and Aditya, who are constantly pulling each other's leg regarding their current research situation. I would also like to thank everyone I've met here - Dhaval, Ankita, Harshit, Manav, Pavan, Tathagata, Tanmay, and others, who have all made my journey memorable. Last but not least, a big shout-out to the amazing lab parties we had. It was fun to see the reactions on my friends' faces when I said I was going out for a lab party "again".

Apart from my lab, I've met an amazing bunch of characters in this college, each with their uniqueness. Firstly, I'd like to thank my close friend T.H. Arjun, who is not with us today.

He's one of the most incredible people I've met in my entire life, with an unimaginable level of talent, dedication, and kindness. He has impacted me in such a positive manner, that he deserves significant credit for anything I achieve. It was fun hanging out with you and Arvindh, discussing pointless things but mostly memes based on our situations. I would also like to thank my friends' gang - Nikit, Akarsh, Kawshik, Gayatri, Tanvi, Ahana, and Sriram, each with their own talents and quirks. I could never have imagined that such a disparate bunch of individuals would end up being such close friends. Thank you for all the fun times, which have made my journey much more unique and special. I would also like to thank my other mini-gang - Sri Ram, Harsha, Trinadh, and Rakesh. I cherish our outings, and discussions while having Biryani in the mess. Thanks to everyone else I've known here - Megha, Aryan, Akhilesh, Viswanadh, Devata, and others, who have made my time at IIIT memorable.

Finally, I would like to thank my family. Thank you Mom and Dad for your immense love and support, and your massive confidence in me. Thank you bro for always keeping it light and fun, with humour only we both would get. The positive energy and love from you all have always kept me going even in difficult times.

I would like to thank all the people I could not name here but who were just as important to this journey.

Here's to hoping for a happy future!

Abstract

Knowledge graphs (KG), which are structures representing information corresponding to entities/topics and their inter-connections, have been playing a crucial role in leveraging information on the web for several downstream tasks such as text-generation, classification, etc. Hence, it becomes vital to construct and maintain such knowledge graphs. There are some previous efforts in populating such KGs and generating relevant entity/node embeddings for this task. However, these methods typically do not focus on analyzing entity-specific content exclusively, but rely on transformational techniques on a fixed collection of documents with certain entities. We define an approach to populate such KGs by utilizing entity-specific content on the web, for generating category-specific entity embeddings. We empirically prove our approach's effectiveness, by utilizing it for a downstream task of Notability detection, associated with one of the most popular and important Knowledge Graphs - the Wikipedia platform.

Wikipedia is a highly essential platform because of its informative, dynamic, and easily accessible nature. The rate of new content being uploaded to it is very high, which makes it essential to moderate this uploaded content. To ensure that only important and relevant content is uploaded to Wikipedia, its editors define a specific set of "Notability" guidelines. These guidelines indicate whether a particular title warrants its own Wikipedia article. So far notability is enforced by humans, which makes scalability an issue, and there has been no significant work on automating notability detection across diverse categories. We work on this problem of creating an automated system to detect the notability of different types of articles/pages, for a vast set of categories.

It is not a trivial task to define a fixed procedure for determining the Notability of Wikipedia pages, as there are different types of pages in Wikipedia, in the way they correlate with the various categories in which they exist. For a given Wikipedia category, articles/pages that are simple category instances co-exist with pages that are associated with the category in a non-trivial manner. It is essential to distinguish Wikipedia pages based on this fundamental difference, to gauge the notability of the page accordingly, as the parameters to look for performing this Notability test vary in each case.

We divide this problem into two components, based on the nature of an article's title. We define two types of article titles - Simple titles and Complex titles. Simple titles correspond to simple category instances/named entities for a given category. For example, "Virat Kohli" is a

simple title of the category "Cricket Players". Complex titles correspond to article titles that have complex dependencies with their category. For example, an article titled "Wake Island" might be present in the "Birds" category, because of its association with the category, but not because it represents an instance of a bird. This distinction helps us define the categories to analyze for generating category-specific embeddings.

Articles with simple titles are further divided into two classes, the "Abstract" class and the "Generic" class, based on whether they represent abstract concepts (such as Temperature / Pressure) or not, respectively, as the process for notability detection is to be followed differently in each case. We construct a dataset with notable and non-notable samples, for 9 categories belonging to the Generic class and 5 categories belonging to the Abstract class. On the other hand, for articles with complex titles, another dataset is constructed for the 9 categories of the Generic class, as defining complex titles for conceptual entities in the Abstract class is non-trivial. We further design a generalizable mechanism to differentiate between simple and complex titles.

Initially, we specifically worked on designing a notability detection system for articles with simple titles. This system is based on web-based entity features and their text-based salience encodings. We further incorporate neural networks and BERT encodings (transformer encoder) to perform binary classification. For validating our system's performance in this task, we utilize accuracy metrics, correlation analysis, ablation study, and prediction confidence on popular Wikipedia pages. Our system outperforms machine learning-based classifier approaches and existing handcrafted entity salience detection algorithms.

Further, we define a system to detect notability specifically for articles with complex titles. This system is primarily defined on the basis of web-based features and the salience of a title in its web-based documents' text. We train a Graph neural network (GNN) that generates attention-enhanced encodings for classification, with syntactic and semantic document graphs as inputs. We evaluated this system similar to the above system for simple titles and observed that it outperforms existing ML-based, naive transformer-based classifiers and handcrafted entity salience methods.

Overall, we define two multipronged systems, which perform the task of generating category-specific embeddings, for performing notability detection of different types of article titles - Simple and Complex, that exist on the Wikipedia KG. We construct corresponding datasets for both types of article titles and evaluate our systems with respect to these datasets, respectively. These systems provide an efficient and scalable alternative to manual decision-making about the importance of a particular topic, irrespective of its category or nature. Based on the empirical proof of the system's effectiveness, it can be concluded that the approach utilized in defining the systems can be extended to any KG-structure, to generate category-specific embeddings.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

1.1.1 Populating a Knowledge Graph (KG)

Knowledge graphs (KG) are defined as structures utilizing a graph topology to represent information corresponding to a vast number of entities/topics and their inter-connections. They put data into context and enable data integration, analytics and sharing. These KGs contain highly crucial information about entities/topics on the web, and are being increasingly leveraged for several downstream tasks such as entity-identification and linking, text-generation, classification, etc. These wide range of potential applications of such KGs has made it extremely important to effectively construct and maintain such knowledge graphs, by updating their information accurately and in a timely manner.

Despite some previous work in automatically populating such KGs and generating relevant entity/node embeddings for this task, there are various shortcomings to the approaches proposed in terms of their effectiveness in capturing key signals related to entities/nodes. Such works typically leverage the existing Knowledge graph-structure, random walks, linear transformational techniques, basic similarity metrics, on a fixed collection of documents, which contain a pre-determined set of entities. We attempt to define a more generalizable approach to populating such KGs, by relying on entity-specific content on the web, for generating category-specific entity embeddings that capture required connections in an effective manner.

There exist several such Knowledge Graphs which are being incorporated into systems for various use-cases. However, one of the most prominent, popular, relevant, and easily accessible knowledge graph is the Wikipedia platform. We empirically prove the effectiveness of our approach, by utilizing it for a downstream task of Notability detection, associated with the Wikipedia platform.

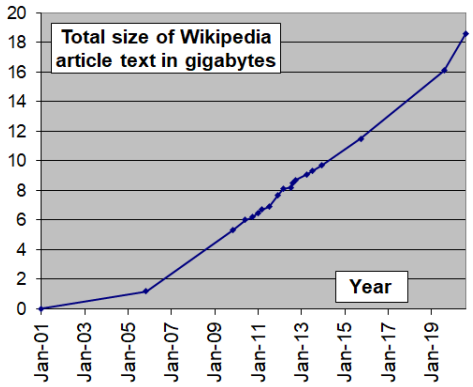

Figure 1.1: Size of articles' text in English Wikipedia, measured in gigabytes (compressed)

1.1.2 Exploring Wikipedia: a massive Knowledge Graph

Wikipedia is a well-known multilingual free encyclopedia, maintained by an open collaboration of volunteers through an editing system. It is a go-to information source for millions of people across the world. Wikipedia represents a massive knowledge graph (KG) structure, with a large collection of article topics as the nodes, and their category-based connections as the edges of the graph. Exploring techniques to effectively populate this KG would assist in several such KG-related downstream tasks.

Hence, we focus on designing systems to analyze and capture the connections between various entities and their categories in this Wikipedia KG, which acts as a fundamental step in populating the KG with more content related to emerging entities. This leads us to the problem of the high content-upload rate to Wikipedia, which should be thoroughly analyzed in order to understand what makes entity-category connections essential according to the guidelines of Wikipedia.

The rate of content being uploaded to Wikipedia is very high, with an average of 545 new articles being created every day[1] (as of November 2023). It can be observed from figure 1.1 that the size of article text in Wikipedia (compressed) has increased more or less linearly to ∼18.5GB, at a rate that increases every few years (as can be seen from the jumps in 2006 and 2019). The article count in English Wikipedia has steadily increased to ∼6.5M million articles over 20 years (figure 1.2). With such a large rate of content creation, a specific degree of moderation must be applied to ensure that only important information is added to Wikipedia. To this end, a set of guidelines called the "Notability" guidelines are defined by the editors of Wikipedia, to regulate the in-flow of content.

[1] `https://en.wikipedia.org/wiki/Wikipedia:Statistics`

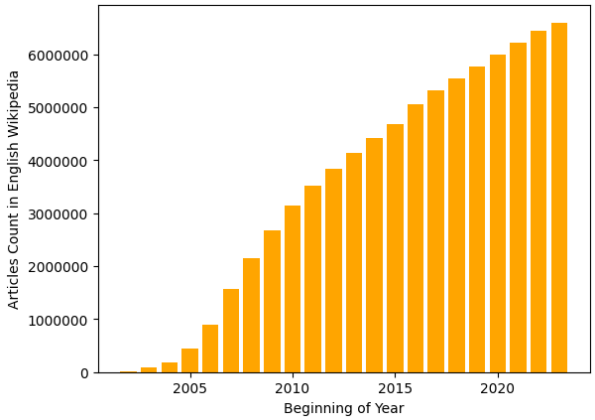

Figure 1.2: Articles Count in English Wikipedia from 2002 to 2023

1.1.3 Wikipedia Notability Test

Notability is a test used by Wikipedia editors to decide whether a given topic warrants its article in Wikipedia. The fundamental aspect used in determining the notability of a topic is the significant coverage of information about the topic in reliable, independent, and verifiable sources (refer figure 1.3). It further depends on various factors like subject-specific guidelines, neutral point of view, etc as described in its official Wikipedia page[2].

Notability detection can be viewed as a binary classification task, where any given topic would have its label, indicating if it is notable or not. To save the effort of content creation for non-notable entities, it is important to perform the Notability check beforehand. This would ensure the efficient functioning of the platform, as storage of unnecessary information is avoided. Performing the Notability test on a large set of topics would require high manual effort and hence there is a need to automate the Notability detection.

[2] `https://en.wikipedia.org/wiki/Wikipedia:Notability`

On Wikipedia, **notability** is a test used by editors to decide whether a given topic warrants its own article.

Information on Wikipedia must be verifiable; if no reliable, independent sources can be found on a topic, then it should not have a separate article. Wikipedia's concept of notability applies this basic standard to avoid indiscriminate inclusion of topics. Article and list topics must be notable, or "worthy of notice". Determining notability does not *necessarily* depend on things such as fame, importance, or popularity—although those may enhance the acceptability of a subject that meets the guidelines explained below.

Figure 1.3: Definition of Notability, Wikipedia

1.1.4 Issues in Automating Notability Detection

The key issue in performing the test in an automated manner is the variations in the applicability of Notability, for different categories. Entities corresponding to a set of categories such as Films or Cricketers have significant information concentrated in specific web domains. On the other hand, entities such as Biological concepts, are relatively more abstract with respect to their content's availability on the web. Additionally, the definition of notability consists of guidelines about reliability, coverage on the web, permanence, verifiability, and subject-specificity. These conditions complicate the decision-making process behind an entity's notability.

Yashaswi et al. [51] proposed a solution utilizing reliable web domain and entity salience features, which are the two most important criteria in the definition of Notability (reliability of content and coverage on the web). This approach was implemented only for the category of "Indian Film Actors", and consisted of several components which were category-specific. Thus, it could not be applied to abstract concepts such as "temperature" and "pressure", which creates an issue concerning generalizability.

Further, notability detection is even more complex in cases where the topic is not an instance of the category but has a non-trivial dependency on the category, such as the article of an island being present in the Wikipedia category "Birds". There has been no significant work on Notability detection for titles with complex category dependencies.

It is necessary to design generalizable procedures for automating notability detection, irrespective of the categorization and nature of a particular topic, by addressing all the issues discussed above.

1.2 Dataset Construction

Different types of articles exist in Wikipedia, with respect to their associations with their pertinent Wikipedia category. In order to handle all such cases in a robust manner, it is necessary to take into account the nature of the content of a page, and its dependency with its category. Hence, we distinguish Wikipedia pages based on the strength and nature of dependency with their corresponding Wikipedia category, for deciding on their Notability.

We construct the datasets from scratch, due to a lack of proper pre-existing datasets targeting this specific task. Based on the nature of the dependency of article titles with their Wikipedia category, we create two types of datasets - Simple titles and Complex titles. Simple titles are defined as the named entities which are simple instances of a given category. In contrast, complex titles correspond to article titles having complex and non-trivial dependencies with their particular category. This distinction is explained in detail in section 3.4.

Based on the primary issue discussed in section 1.1.4 of different types of topics, Wikipedia articles with simple titles are further divided into two classes, the "Abstract" class and the "Generic" class. This distinction is based on whether a particular topic represents an abstract

concept (such as Temperature/Pressure). If so, it is considered to belong to the "Abstract" class, otherwise, it belongs to the "Generic" class (distinction explained further in section 3.3). The processing for notability detection is different in each case.

We construct a dataset with notable and non-notable samples, for 9 categories belonging to the Generic class, comprising a total of ∼30K samples, and 5 categories belonging to the Abstract class, comprising a total of ∼5K samples. Roughly, an overall distribution of 55-45 is followed with respect to notable and non-notable samples respectively.

On the other hand, another dataset is constructed for the 9 categories of the Generic class for Wikipedia articles having titles with complex categorical dependencies. For a total of ∼9K samples, a similar overall distribution of 55-45 is followed with respect to notable and non-notable samples respectively. We ignore the Abstract class in this case because defining complex titles for conceptual entities in the Abstract class is non-trivial, as a particular concept could belong to multiple categories in an unambiguous manner rather than as a complex dependency (reason explained further in section 3.4).

We also designed a generalizable mechanism to differentiate between simple and complex titles in general, for any given category. This is an unsupervised approach designed based on the nature of Wikipedia categories for existing Wikipedia articles in the given category, a pre-defined set of category-related keywords, and the categorizations of a particular topic of the category, as identified from the web.

1.3 Notability of Wikipedia article titles with Simple Named entities

We design a system to handle Notability detection of Wikipedia pages with simple titles, by addressing the key challenges discussed above. This system relies on constructing entity embeddings with respect to its corresponding category, which is further used for classification. The dataset of Simple titles, with Generic and Abstract classes, is utilized for validating the effectiveness of this system. The approach designed by us addresses the challenge of category-specific attributes, through the inclusion of a generic set of features that are common across categories, making the entity embeddings generalizable and effective. Further, our solution handles the case of abstract entities, by utilizing reliable content from the web, rather than fixed web domains.

Since it primarily deals with named entities of categories, it relies on several web-based entity salience features such as query logs analysis, relevant documents about the entity on the web, presence in the Wikipedia ecosystem, social media, and online news. Besides heuristic count-based measures extracted from the web, we also utilize Bidirectional Encoder Representations from Transformers (BERT) to incorporate attention-enhanced text-salience encodings. Thus,

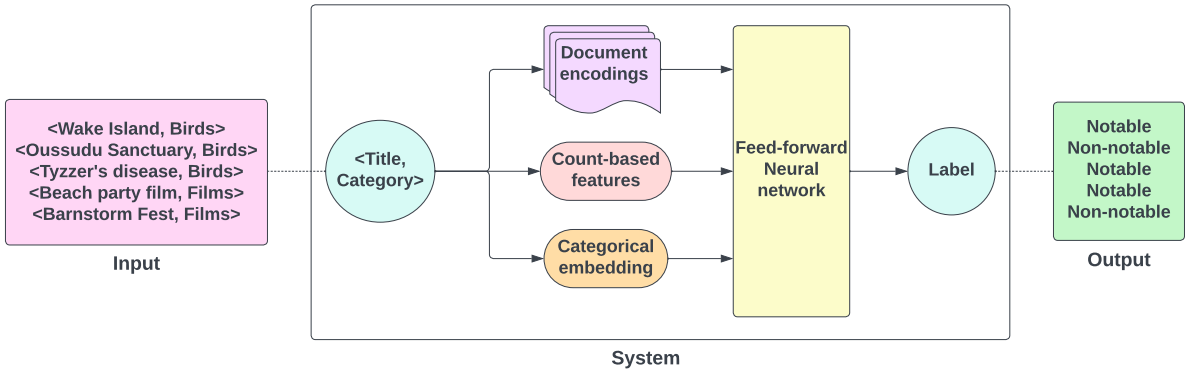

Figure 1.4: Overview of complete pipeline with examples from the dataset (Complex titles)

all these numerical features and encodings are combined to form the unified entity embedding, generated in accordance with its category, to perform binary classification, indicating if a particular entity is notable or not. The high-level pipeline can be understood from figure 1.4 defined for Wikipedia pages with titles having complex categorical dependencies, as it is similar in the case of simple Article titles, where a particular named entity along with its category are passed to the system, which processes it primarily based on statistical features, document encodings, categorical embeddings, and utilizes neural networks to generate the final embedding to decide on Notability.

For validating our system's performance in this task, we utilize accuracy metrics, correlation analysis, ablation study, and prediction confidence on popular Wikipedia pages. Our system outperforms machine learning-based classifier approaches and existing handcrafted entity salience detection algorithms.

1.4 Notability of Wikipedia article titles with complex categorical dependencies

We design another system to specifically handle complex titles, i.e., titles/topics having complex categorical dependencies with their particular category. The dataset of complex titles is used to validate system effectiveness. Similar to the above system, this system has a set of category-agnostic features, which makes it generalizable, and applicable to several other categories. It is ensured that the embeddings generated for each title are generalizable and capture the crucial dependencies between the topic in the title and the category under question.

Web-based features defined above are also a part of this system. BERT is utilized to obtain the initial level of salience encodings of relevant documents for a topic. However, to capture complex categorical dependencies, we further add another graph-based architecture on top of

these encodings, which constructs syntactic and semantic graphs from document content and passes them through Graph Attention (GAT) layers to obtain attention-enhanced encodings. We utilize specific keyword encodings obtained from this process to obtain the final unified embedding, that encapsulates title-category dependencies effectively, to perform binary classification for a title. This pipeline can be summarized from figure 1.4, which contains key high-level components of the system, working on examples from the dataset.

We similarly evaluate this system as done for Wikipedia pages with simple titles. Based on accuracy and prediction confidence, our system outperforms machine learning-based, naive transformer-based classifiers and handcrafted entity salience methods.

1.5 Thesis Key Contributions

With this work, we highlight the need for automatically populating Knowledge graphs (KGs) by reliably encoding the connections between entities and categories which are a part of the KG. In this context, we work on the problem of notability detection of articles for Wikipedia KG and its automation, to reduce the high manual effort in content creation and ensure the efficient functioning of the Wikipedia platform. The key contributions made by our work are listed below.

1. We make key distinctions about different article titles in Wikipedia and propose a generalizable mechanism to distinguish between these titles' types (Simple and Complex).

2. We construct two datasets from scratch, for Simple and Complex article titles respectively, where the labels are binary, indicating notable/non-notable titles.

3. We design two multipronged systems which perform the task of generating category-specific embeddings, for performing notability detection of different types of article titles - Simple and Complex, that exist on the Wikipedia KG, exhaustively. This is performed by emphasizing necessary aspects of system design in each case while relying on web-based information and text-based salience as central components.

1.6 Thesis Outline

Overall, this thesis is divided into 6 chapters, of which a small description of each chapter is given below:

- In **chapter 1** (this chapter), we highlight the need for populating KGs by encoding entity-category connections. In this context, we motivate the need to perform the related downstream task of automating Notability detection for the efficient functioning of Wikipedia. We also introduce the key contributions of our work here.

- **Chapter 2** discusses some of the related work that has been done in the past for Knowledge-graph population, Notability detection and similar directions that have been explored.
- In **chapter 3**, we discuss the need for a relevant dataset and propose our dataset based on the problem definition. We also propose a mechanism here to differentiate between sub-parts of the problem, which is a necessary first step.
- In **chapter 4**, we design a novel web-based system to generate attention-enhanced entity embeddings capturing its connection with its category, to detect the Notability of Wikipedia pages with simple titles in an automated manner.
- In **chapter 5**, we design another web-based Graph-attention-centric system utilizing a new type of category-specific embeddings to detect the Notability of titles having complex categorical dependencies, for Wikipedia, in an automated manner.
- **Chapter 6** concludes this thesis by discussing the overall contributions and impact of the work. We also discuss possible future work in this direction of generating effective category-specific embeddings, to assist with tasks such as improving notability detection.

Chapter 2

Related work

2.1 General work on Entity embeddings and Knowledge graphs

There has been some previous research in the field of generating and utilizing specific node embeddings in the existing Knowledge-graph structures for performing downstream tasks. Palumbo et al. [48] worked on creating a user-item recommender system by extracting property-specific sub-graph embeddings, by using a modified version of the node2vec architecture [21], and performing edge-labelling via similarity metrics. Saeed et al. [58] worked on creating a randomized graph-walk based approach and computing specificity metric to capture semantic relationships between nodes. Shubham et al. [9] performed fine-tuning of BERT [12] to obtain query-specific entity encodings, by leveraging high-level passages in Wikipedia. Yang et al. [73] worked on representing entities and relations via linear and bilinear transformations of vectors and matrices, for performing rule extraction from KG. Wen et al. [76] worked on modelling cross-over interactions between entities and relations in a KG using non-linear transformations and Hadamard product, and additionally performed the task of graph traversal for embedding-based explanation search.

Previous works also exist in the population of Knowledge-bases (KBs) and Knowledge-graphs (KGs) by utilizing content from the web for extracting knowledge. Hailun et al. [38] worked on constructing a static inter-dependence graph based on entities, categories and their inter, intra semantic dependencies. Evidence propagation is performed iteratively on the connectivity matrix in order to associate entities and categories. Dèlia et al. [18] created a system for Knowledge graph population from aggregated news articles, using type markers and imposing data-integrity constraints. BERT is used for entity detection and as the relation-extraction model to formulate initial set of RDF triples.

Despite significant research in these fields, there are certain aspects which are not adequately addressed by these works, both theoretically and practically. The knowledge-graph structures are assumed to be well-established and containing all the entities and categories necessary, which is not the case with newly emerging entities that are to be added to the graph, whose

context is not sufficiently captured in the KG initially. Simple transformation techniques and standard embedding, aggregation methods are typically used for node embeddings, which do not effectively capture the essence of the entity depicted by a node. Further, existing KG-population approaches look at specific set of task-based documents of articles for downstream tasks, which pre-determine the collection of entities. Hence, there's no generalizable entity-specific document-based analysis, where the content is curated specific to each relevant entity and the category-dependency required to be modelled.

In order to address the above-discussed shortcomings, we define generalizable techniques to generate entity embeddings capturing categorical dependencies effectively by focusing on web-components in a specialized manner. These embeddings are utilized in the context of Wikipedia, for the task of Notability determination, to establish the extent of connections between entities and their categories, and marking notable relations. This empirical proof assists in generalizing the defined approach by incorporating these embeddings for other KG-related tasks. We further deep-dive into the field of Notability and research in related directions, to better gauge the depth of the applications.

2.2 Notability-definition related work

There have been some research works and studies on the definition of Notability for Wikipedia. Margolin et al. [13] worked on designing a platform "Wiki-worthiness," where participants can collaboratively assess the notability of potential Wikipedia topics, relying on collective manual effort. Jodi et al. [62] emphasized deletion discussions and their outcomes on Wikipedia, and identified patterns in decision-making based on deletion nomination, indicating variations in community behavior. The study conducted by Shyong et al. [35] examines whether the platform is experiencing an increase in the diversity of articles that receive attention over time. The work by Mackenzie et al. [36] and Franziska et al. [41] stress the impact and role of gender and race in notability determination. Dario et al. [66] advocate for a more liberal content-inclusion approach that involves collaborative decision-making among Wikipedia editors, to mitigate the biases inherent in relying solely on notability criteria.

Despite the above works being significant in the field of Notability and Wikipedia, they do not propose approaches regarding how to solve the scalability issue posed by Notability detection. To the best of our knowledge, there hasn't been much work in the direction of automating this process.

Yashaswi et al. [51] targeted the exact problem of Notability Determination, for Indian Film Actors, utilizing reliable web domain and entity salience features, the two most important criteria in defining Notability (reliability of content and coverage on the web). Initially, the top reliable web domains for a category are identified, such as "imdb.com" for the domain of Actors. These web domains are identified based on their frequency of occurrence in the top

Table 4: Dimensional Vector Model with a single domain RF_k and its entity salience scores as features

Vector	RF_k	$E_p(RF_k)$	$E_f(RF_k)$	....	Label
Entity-1	1	8	9	...	Yes
Entity-2	1	0	0	...	No
Entity-3	0	0	1	...	Yes
.	.	.	.	...	.
.	.	.	.	...	.
Entity-N	0	0	0	...	No

Figure 2.1: Feature vector representation for binary classification in [51]

few retrieved results on performing a search with a query containing entity name and category name, for each entity. Each web domain's frequency is thus noted accordingly, and additionally, its occurrence frequency in Wikipedia articles' external links section is also noted. Both these scores are combined by a weighted average scheme, and the web domains with the highest set of scores are chosen. The presence of an entity in these web domains is also recorded as a boolean feature. Secondly, handcrafted entity salience metrics such as entity mentions count, occurrence in the first sentence, occurrence count in the first 3 sentences, etc., are defined to capture coverage about the entity in its profile page, in these web domains. These two sets of features are combined to perform binary classification if an entity is notable or not, (as in figure 2.1) using SVM architecture.

There has been similar work on detecting emerging entities, which could potentially be used to identify notable entities. Michael et al. [16] worked on this direction of emerging entity detection, which explores the challenges of identifying new entities to be added to Wikipedia based on media monitoring. They further propose an ML-based approach to classify whether an entity is to be added to Knowledge Graphs (KG) or not, based on a dataset constructed from English news articles. Dèlia et al. [17] worked on an online multilingual system for event detection and comprehension from media feeds, by retrieving information from news sites and building a KG, which is extended with a Dynamic Entity Linking (DEL) module to detect emerging entities on unstructured data. The primary drawback with all of these works is the applicability only to a select set of categories, and not considering other web contexts for an entity.

2.3 Text-based understanding for a topic's salience estimation

There have been similar directions as Notability detection, which could potentially be adapted to this problem. As discussed in the definition of Notability and the primary work on Notability detection for Wikipedia (Yashaswi et al. [51]), salience-based approaches are essential

in extracting relevant features corresponding to a particular topic. Hence, we explore similar directions dealing with such work.

2.3.1 Entity Salience and Event Salience based techniques

Identifying the salience of a particular entity or an event in a given document or a set of documents is one of the key components in gaining an understanding of the entity's/event's relevance and coverage. This aspect is an essential part of our final designed system, which has been created based on a review of related approaches while also ensuring generalizability and scalability.

Jesse et al. [14] worked on a novel entity salience task that involves training an ML model to identify and rank entities within a document based on their salience, on a dataset derived from existing knowledge bases and web documents. Xiong et al. [72] worked on identifying salient entities in a document, based on knowledge-enriched entity representations, and Kernel-based modeling. Liu et al. [39] propose feature-based and neural-based models for event salience identification, leveraging features such as event mentions, contextual information, and document structure. Kevin et al. [26] propose incorporating phrase extraction models into the process of entity salience detection, emphasizing the potential benefits of capturing the context in which entities appear. Lu et al. [40] worked on an unsupervised entity and event salience estimation, by constructing dependency-based heterogeneous graphs to capture the interactions of entities and events. Trani et al. [68] discussed the application of lexical features for entity linking and salience detection. Gamon et al. [19] analyzed entity-document correlation with respect to clicks of document URLs for entity-based queries.

2.3.2 Summarization-based techniques

Sentence-level salience also plays a role in identifying which parts of a document are important in providing information about an entity's salience with respect to it, and the category in general. Summarization techniques utilize such encodings to capture key information in a document, that further provides insights about the topic's coverage.

Günes et al. [15] worked on a graph-based algorithm for extractive text summarization, that leverages the PageRank algorithm and takes into account sentence centrality and salience. Rahman et al. [56] worked on query-based text summarization, which finds semantic relatedness score between query and input text document for extracting sentences, by finding the correct sense of each word of a sentence with respect to the context of the sentence. These queries could be tuned in such a way as to gain more information about the desired entity. Rahman et al. [55] also explore the use of linguistic features, graph-based models, and machine learning algorithms in the context of query-based summarization. Ahmed et al. [44] utilize document graphs based on the relationships between sentences, incorporating both content and contextual information,

for ranking sentences to create query-based summaries. Piji et al. [37] utilized Variational Auto-Encoders for learning the latent representations of sentences to summarize.

2.3.3 Semantic modeling for information retrieval

Semantic techniques for information retrieval capture key features while matching queries to documents. Hence, carefully designing a query containing details such as topic name, category, etc. could assist in extracting required features from selected documents, and speak for the salience of the topic with respect to the document and the category. A review of such techniques has helped incorporate such semantic features into our system.

Zhang et al. [75] used semantic similarity measures for table retrieval, to capture the meaning and context of both the query and the tables. Each table in the corpus is represented as a vector by aggregating the word embeddings of its content and matching them with the query's word embeddings to capture relevance. Mayank et al. [59] propose combining both syntactic and semantic similarity measures to form a comprehensive similarity metric, to provide a more nuanced and contextually relevant evaluation of document relevance. Yuanyuan et al. [53] worked on salient contextual aspects to improve semantic matching, and assigned weights to query and document terms based on their salient context. The work by Liang et al. [49] utilized local references and global references of query terms in a document, captured by a deep neural network using term gating mechanism along with a CNN and a 2D-GRU.

2.4 Topic's web-based popularity estimation

According to the definition of Notability, the existence of independent sources about a topic is necessary, and this information about a topic is typically obtained from the web in different forms - either through social media, category-related web domains, news sources, organizations, etc. as this is the primary form of consumption of content for any individual. Signals from the web play a vital role in deciding on notability, and this area has also been widely studied.

Alexandru et al. [67] and Moniz et al. [45] discussed different types of features about an entity's popularity on the web, such as social features, temporal features, and user-interaction-based features, and tested them using ML models. Sofiane et al. [1] used time series forecasting to predict the number of visits an article will receive when posted, relying on the popularity of similar articles and the historical popularity of its main topic. Nadav et al. [20] worked on developing a linear auto-regression model to predict future query counts and additionally plug them into an existing trend detection scheme, to surface search trends.

2.4.1 Popularity signals from Social Media

Several works have exclusively focused on social media platforms for estimating popularity. Priya et al. [54] employed various features for identifying salient named entities in Twitter, including the frequency of entity mentions, the presence of hashtags and user mentions, and the position of entities in the tweet. Hiba et al. [63] discussed the dynamic nature of social media platforms, normalization techniques (such as Z-score normalization), and evaluation metrics to address the challenges in comparing popularity metrics across diverse social media entities. Amir et al. [31] investigated the application of sentiment analysis and social network analysis techniques to assess the political popularity of individuals in the context of social media.

2.4.2 Popularity signals from Online News

Online news and social media have a symbiotic relationship in terms of providing information with respect to a particular topic. Few works have tackled their relation in a conjunctive manner. Kemal et al. [2] used supervised learning-based machine learning techniques in order to predict news popularity in social media sources. Roja et al. [4] worked on predicting the popularity of news articles in social media, by relying on news article features related to textual content, temporal information, and user engagement metrics. Pedro et al. [60] tackled the problem of predicting entity popularity on Twitter based on the news cycle, by applying supervised learning on approaches based on signal-based, textual-based, sentiment-based, and semantic features.

Apart from news in the context of social media, online news sources also play a direct role in providing information about a topic and discussing its salience. Nirupama et al. [3] designed an approach where salient entities in news articles were detected using the aboutness score of an entity, with respect to its description and article's content. Yang et al. [74] proposed a named entity topic model (NETM) to extract the textual factors that can drive popularity growth, and used it to predict the popularity of news articles by computing the accumulation of popularity gains generated by its named entities (NEs) over all the topics.

2.5 Graph-based approaches for text-based dependency extraction

Dependency extraction is essential, especially in cases where notability is to be resolved for titles having complex dependencies with their respective topics. Graph-based techniques have been effective in dealing with such dependency-extraction tasks because of their ability to learn representations in accordance with document structure, based on words or entities. We explore such graph-based techniques, especially utilizing Graph neural networks, to understand their performance on similar tasks.

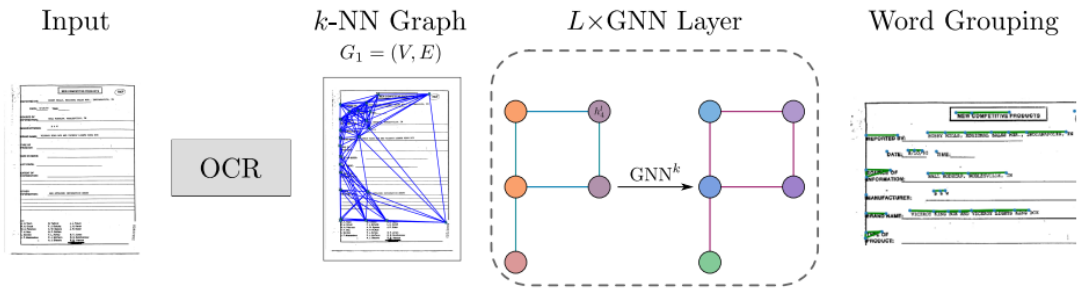

Fig. 1. Overview of the proposed word grouping approach. The text content and location of the words in the input document is encoded in a word level k-NN graph. This is fed into a GNN with L layers. The word grouping is formulated in terms of a binary edge classification problem, that is, 1's indicates that these words belong to the same entity.

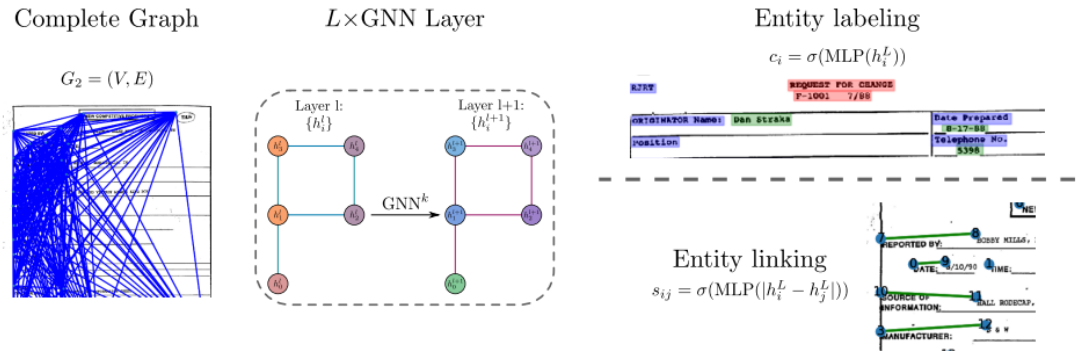

Fig. 2. Given the discovered entities (see figure 1), a complete entity level graph is generated and fed into L GNN layers. Thus, the tasks of entity labeling and entity linking are formulated in terms of node and edge classification respectively. The GNN is trained separately for each task.

Figure 2.2: Architecture overview of [6]

One of the works in this field, which we have used for reference, is the work of Carbonell et al. [6], which deals with Named-entity recognition and relation extraction in semi-structured documents using GNN encodings. In this task, for a given document, its entities are to be defined and classified into pre-defined categories, and meaningful pairwise relationships between entities are to be discovered. The document is initially represented as a graph whose nodes are the words detected in the OCR process, and edges are created based on the k-NN of word representations. Groups of words obtained after edge classification are defined as document entities. Another fully connected graph is constructed with these entities as nodes, and the graph attention mechanism is employed to obtain updated node encodings, and these are used to perform node (entity) classification by passing through an MLP (refer figure 2.2). Link prediction is performed by passing entity embeddings of the pair of entities through another MLP. We utilize this concept of generating graph-based encodings and augmenting a classifier head for classification.

Huang et al. [25] utilized GNNs for text classification, by parameter-sharing across individual document graphs. Wei et al. [29] worked on graph-based text representation, which is capable of capturing term order, term frequency, term co-occurrence, and term context in

documents, to discover unapparent associations between two and more concepts (e.g. individuals) from a large text corpus. Zhou et al. [78], Jae-Young et al. [8] and Sonawane et al. [65] performed comprehensive reviews of various graph-based approaches for relevant downstream tasks, discussing aspects such as message passing mechanisms, spatial and spectral approaches to graph convolution, Graph Attention Networks (GATs), Graph pooling and Graph Pooling and Downsampling, word co-occurrence graphs, syntactic dependency graphs, and semantic graphs, graph-based NER, sentiment analysis etc.

The thorough literature review conducted across different fields of work assisted with designing a robust approach for generating embeddings for notability detection. The works discussed above cannot be directly plugged into the problem of notability because of the constraints posed by the definition of Notability, generalizability, and scalability. We take into account the key concepts discussed by these works to design our systems which are effective in the defined task of automated notability detection of a topic irrespective of its category or nature. Our systems utilize specifically-designed embeddings which can be extended to any similar use-case in such a KG-based setup.

Chapter 3

Dataset: Defining hierarchy of Wikipedia article titles, and extracting data

3.1 Overview

In this chapter, we describe our dataset constructed for the task of generating entity embeddings for Notability determination. We define different types of titles on Wikipedia, based on the correspondence with their category - Simple titles and Complex titles. Simple titles are category instances, while complex titles have non-trivial dependencies with their given category. We further subdivide simple titles based on their nature (concept-based abstract entities or not) and classify them as Generic/Abstract. The Generic class dataset comprises 9 diverse categories, while the Abstract class dataset comprises 5 categories. The complex titles dataset uses 9 categories of the Generic class dataset.

We extract samples for binary classification on notability for each categorization of titles, by following well-defined procedures. We also design an unsupervised approach to differentiate between simple and complex titles in general, in a category-agnostic manner. This approach relies on the categorizations of a particular topic of the category, as identified from the web. The high-level pipeline for a title and its classification as Simple/Complex is provided in figure 3.1, where the first step is to identify the partition of a title and use the respective system to decide on its notability.

3.2 Motivation for defining Simple/Complex Article titles

Wikipedia comprises different types of pages, that have titles having specific associations with their corresponding category. A Wikipedia page could be either a simple category instance or a page loosely coupled to the category. This makes it essential to handle all such pages robustly and effectively while deciding a page's Notability. The nature of the content of an article, and its dependency with its category play a role in the generation of its embedding and

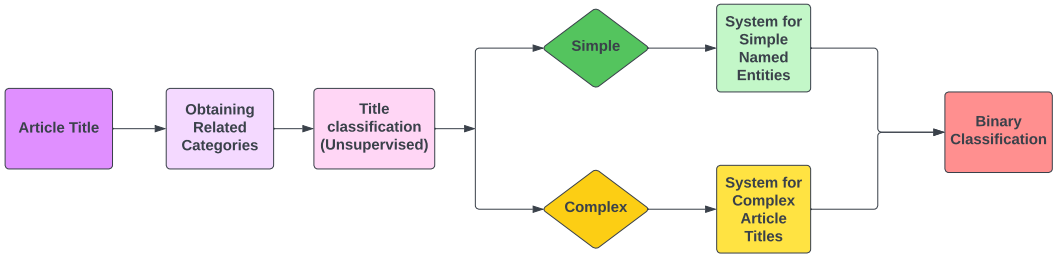

Figure 3.1: Overview of pipeline for title-based binary classification of Notability

classification of the article's topic as notable or non-notable. Hence, it is essential to distinguish Wikipedia pages based on the fundamental factor of the strength and nature of dependency with their corresponding Wikipedia category. This assists in deciding on the notability of the page accurately, by analyzing relevant parameters and capturing entity-category interconnections in the embedding more effectively. Thus, we categorize titles of articles in Wikipedia into two classes at the broader level - Simple titles and Complex titles, which are explained in detail below.

3.3 Extracting samples of Article titles with Simple Named entities

We consider a Wikipedia article to have a simple title, if the title refers to a simple named entity or a category instance, for a defined category. For instance, for the category of "Indian Film Actors", article titles "Shah Rukh Khan" and "Ranbir Kapoor" are categorical instances. They pertain to trivial and direct examples of a category, which typically comprises many such examples. We further partition such "Simple" titles into two classes, based on their nature - Generic class and Abstract class. This distinction is necessary to ensure that named entities are classified as notable/non-notable, to eliminate the bias from the nature of their category, which in turn affects the quantity and quality of information about them on the web. Both of the classes can be extended to any new category with minimal training data, due to their generalizable nature. These classes and their differences are described in the following manner.

3.3.1 Generic Class

The Generic class consists of categories with easily identifiable reliable web domains, such as Films, Actors, Cricketers, Cities, etc (listed in table 3.3.1). For instance, it can be identified that "imdb.com" is a reliable web domain pertaining to the category of Indian Film Actors, and "espncricinfo.com" is a reliable web domain pertaining to the Cricketers category. Such web

domains contain a large number of simple category instances/named entities that belong to the corresponding category, as "imdb.com" contains many profile entries of Indian Film Actors, and likewise "espncricinfo.com" contains many Cricketer profile entries. We construct a dataset with such categories where fixed web domains could potentially be identified for each category. Further, reliable and concrete information about the corresponding categorical instances can be found in these web domains.

For the Generic class, a dataset consisting of about 30K samples is collected, comprising entities belonging to 9 categories, namely Film Actors, Cricket Players, Tourist attractions, Medicinal plants, Universities, Cities, Birds, Football players, and Films. We have ensured that there is diversity among the characteristics of the categories in our dataset. This is done to validate the extendability of our system to any category, that specifically follows the definition of the Generic class.

This dataset's distribution can be found in table 3.3.1, where positive samples correspond to notable entities, and negative samples correspond to non-notable entities.

Dataset Distribution Statistics for Generic Class

Category	$\#Pos$	$\#Neg$	$\#Tot$
Actors	2885	2876	5761
Cricketers	2597	2403	5000
Tourist attractions	2500	2500	5000
Medicinal plants	2500	2034	4534
Universities	1584	1584	3168
Cities	1590	795	2385
Birds	1100	517	1617
Football people	1006	503	1509
Films	938	469	1407
Total	**16,700**	**13,681**	**30,381**

3.3.2 Abstract Class

On the other hand, entities corresponding to the Abstract class have content in documents spread out across the web, rather than fixed web domains. Entities belonging to these categories are said to be at an abstract level with respect to their content's availability on the web. For instance, for the titles "Temperature" and "Pressure", the information corresponding to them is scattered across the web, and cannot be entirely found in specific pre-defined web domains.

Categories corresponding to this class include Biological concepts, Ragas[1] in the Carnatic music system[2], etc. It is not practical to identify specific reliable web domains for categories such as Biological or Chemistry concepts, whose corresponding titles cannot be completely defined or confined to a specific set of web domains. Hence, in such cases, a more reliable mechanism would be to rely on topic-specific documents on the web, as the coverage of such abstract topics cannot be gauged by a pre-selected web-domain-based approach.

For the Abstract class, a dataset consisting of nearly 5K samples is constructed, comprising entities belonging to 5 categories, namely Biology, Chemistry, Psychology concepts, Carnatic Ragas, and Computer Software. Note that the labeling of categories such as "Computer Software" is ambiguous, as it does have a few relevant web domains. However, the "abstractness" here is defined as not having a fixed form, not pertaining to a person/organization/physical product, and most importantly, too many to account for in a small number of web domains (similar to concepts, theories) as content is widely spread across the web. This dataset's distribution can be found in table 3.3.2.

Dataset Distribution Statistics for Abstract Class

Category	$\#Pos$	$\#Neg$	$\#Tot$
Softwares	1384	692	2076
Psychology	732	366	1098
Chemistry	537	269	806
Biology	328	164	492
Ragas	137	283	420
Total	**3118**	**1774**	**4892**

3.3.3 Data collection and annotation

After the classification of categories into Generic/Abstract classes, we worked on defining mechanisms to collect necessary data for each class, and annotate them as notable/non-notable. The ground truth labels for both datasets are defined based on the presence or absence of the entity's Wikipedia articles. If an entity is observed to have a corresponding Wikipedia article, it is considered notable and is assigned the label 1, otherwise, it is considered to be non-notable and is assigned the label 0, as annotated in the work of Yashaswi et al. [51].

We used specific methods to identify notable and non-notable entities, by ensuring that they follow the above-defined rule for annotation. Notable entities were identified by crawling

[1] `https://en.wikipedia.org/wiki/Raga`

[2] `https://en.wikipedia.org/wiki/Carnatic_music`

through the Wikipedia category and sub-categories of the required domain, such as "Indian Film Actors', "Cities', and so on. For obtaining non-notable entities, a more well-defined approach is necessary, to ensure that a Wikipedia article does not exist for that corresponding entity. This approach is briefly described below.

- For a given category, A list of entities belonging to the category is curated, by manually identifying reliable web domains for the categories and extracting titles listed in these domains.
- Based on the list of entities, a query is formulated for each entity, including its title and category name. A search is performed on Wikipedia for this query, and the top 3 retrieved results are recorded.
- Entities having the least overlap with their top 3 results' titles, and having Jaccard similarity co-efficient less than a threshold of 0.24, are considered to be non-notable and form the negative samples in the dataset. This is based on the intuition that if a relevant Wikipedia article existed, it should have appeared in the recorded top 3 results. An example of such a search being performed for a notable and non-notable entity can be visualized in figure 3.2.

In this procedure, the parameters used such as the number of search results to be analyzed (3) and the Jaccard threshold defined (0.24) were decided based on empirical experimentation. It was ensured that both datasets consisted of enough samples for both positive and negative labels, to the extent possible. In both cases, the percentage-wise train-test-validation split of the data was 70-15-15. We make this dataset publicly available[3].

3.4 Extracting samples of Article titles with complex categorical dependencies

We have seen that "Simple" titles comprise instances of a given category, such as an article about the bird "Bulbul" in the "Birds" category, and an article about the film "Inception" in the "Films" category, etc. On the other hand, Complex titles are defined as those article titles having complex and non-trivial dependencies with respect to a category. For example, consider the article "Florida Film Festival" in the "Films" category, and the article "Wake Island" in "Birds" category. These titles are not category instances but are associated with the category, warranting their articles in such categories. According to Wikipedia's structure of articles, such titles should exist in related categories as illustrated for these two articles. These type of

[3] https://www.dropbox.com/sh/3ybp2ha494bikgq/AABVg-vSOTEOMoEcVvexq9YKa?dl=0

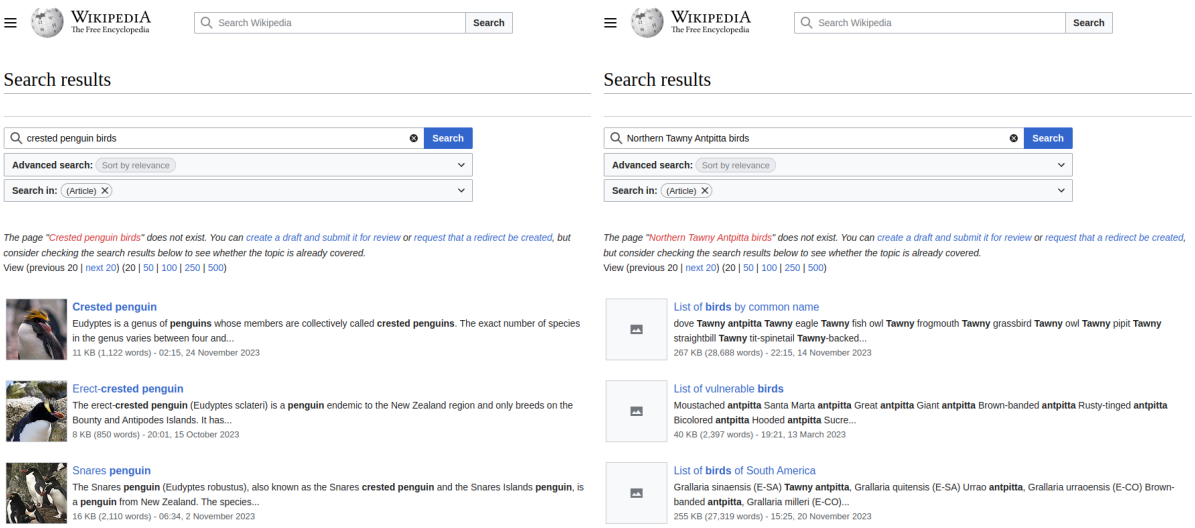

Figure 3.2: For the search of a notable entity (left), the Jaccard similarity of the most relevant result in the top 3 would be high, while it would be typically less than the threshold for a non-notable entity (right)

entity-category dependencies also play a crucial role in the population of Knowledge Graphs in general.

We construct a dataset of complex titles from scratch, with nearly 9K titles, corresponding to 9 diverse categories (table 3.3) discussed in the Generic class above. We do not consider the categorization of Abstract concepts while defining the dataset, as the boundary of definition is not as clear as in the case of a generic category such as Films, Cricketers, etc. For example, an article for "Temperature" as a physical quantity is different from that of a hyperparameter used in neural networks to control randomness, despite having the same title. In this case, the title has multiple definitions and usage contexts, but there is no clear definition of complex dependency.

As data annotation is done in the case of Wikipedia pages with simple titles, a complex title is labeled as notable (positive sample) if it has a Wikipedia article, and labeled as non-notable (negative sample) if it does not have a Wikipedia article. As opposed to the case of simple Article titles, which were category instances that were available in relevant web domains of the category, we require a procedure to identify complex titles that fit the definition regarding their association with the category. We curate a set of complex titles for each of the categories in the dataset, by following the approach described below.

3.4.1 Data collection and annotation

Complex titles for a given category consist of positive and negative samples, which are obtained via different mechanisms. To extract positive sample titles, we crawled through each Wikipedia category. In these titles, complex titles are identified by manually understanding patterns of titles that are not simple category instances. These patterns are analyzed based on the Wikipedia categories section of the articles. For example, in the "Birds" category, terms related to islands in this section indicated a complex title. Further, for each such title, we also store an additional parameter called categorizations. The Wikipedia categories mentioned on the positive sample's corresponding page are defined as the title's categorizations.

We define another procedure to define negative samples, that fit the rule of annotation. Firstly, for a category, the most frequently occurring categorizations in its positive samples are noted. Then, reliable web domains are manually identified for these identified categorizations, and titles mentioned in these web domains that are verified to not have a Wikipedia article are considered negative samples. For instance, in the "Birds" category, the categorizations "Islands" and "Wildlife Sanctuaries" are found to be frequently occurring (Table 3.3). Reliable web domains containing data about "Islands" or "Wildlife Sanctuaries" are manually identified, and titles in these web domains that do not have Wikipedia pages are considered negative samples. We use the approach discussed in section 3.3.3 to decide if titles in these web domains have corresponding Wikipedia pages or not. Note that for a negative sample title, structured information about it on the defined web domains is obtained by web-scraping, and considered as its categorizations parameter. In figure 3.3, we can observe the difference in the way title categorizations are obtained for positive samples and negative samples, for two complex titles in the dataset corresponding to the Cricketers category.

The percentage-wise train-validation-test split of the data was 75-10-15. Table 3.3 lists sample counts, and the categorizations used to obtain negative samples. Non-trivial complex dependencies can be clearly visualized for each category. For example, "Music albums" was

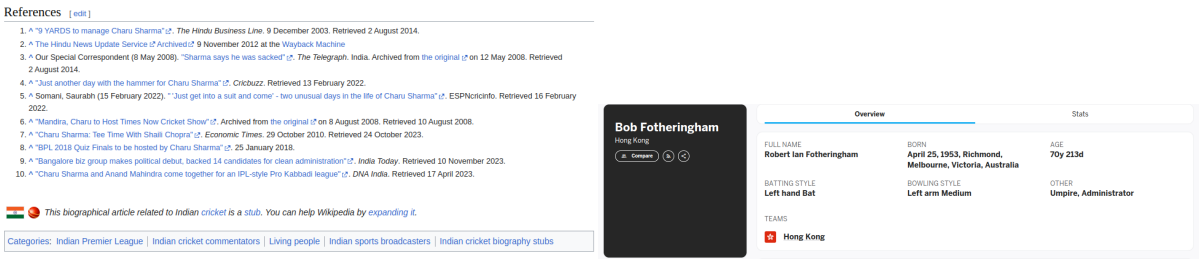

Figure 3.3: For a notable complex title (left), the categories mentioned in its Wikipedia page are defined as its categorizations. For non-notable titles (right), the structured data in reliable web domains is scraped to define similar categorizations.

Table 3.3: Complex Titles' dataset statistics

Category	**Negative samples categorizations**	*#Pos*	*#Neg*
Birds	Islands, Wildlife Sanctuaries	308	277
Cities	Cricketers, Politicians and Kings, Sports venues	400	155
Cricketers	Cricket umpires, Cricket administrators and referees, Sports films, sports Video games	282	359
Films	Film Festivals	700	700
Football people	Sports commentators, Businesspeople and Entrepreneurs, Classical Music composers, Football clubs	700	653
Indian film actors	Films, Lyricists, Music Directors, Singers	346	448
Medicinal plants	Films related to drugs, Music albums	700	700
Tourist attractions	Festivals of the world, Television shows	500	305
Universities, Colleges	Attorneys	700	678

observed to be a valid categorization in the category of "Medicinal plants", describing drug use in music, bands, etc. Titles pertaining to cricket commentators and umpires are found to have complex dependencies with the Wikipedia category of "Cricketers", and titles of politicians associated with a city are found in the Wikipedia category of "Cities". These examples indicate the nature of the organization of content in Wikipedia and establish the necessity to handle such complex categorical dependencies. Note that the categorizations used are based on frequency and data availability in the required format. We make the dataset publicly available[4].

To extend the datasets of simple and complex titles, by adding titles that satisfy the above-defined partition of article titles, it is necessary to define a generalizable procedure for understanding if a particular title of a category is a simple/complex title. We discuss this method of title classification below.

3.5 Title Classification as Simple/Complex

It is not a trivial task to automatically detect if a given article title is simple or complex, with respect to its category. We define a generalizable mechanism to differentiate between simple and complex titles in general, by performing title classification in an efficient and category-agnostic manner. This is an unsupervised approach designed based on the nature of Wikipedia categories for existing Wikipedia articles in the given category, a pre-defined set of category-

[4] `https://www.dropbox.com/scl/fi/yw3zecrhaagqtwjjxu66r/data.zip?rlkey=lkui787ld7fvtvjpf6r4lr7pf&dl=0`

related keywords, and the categorizations of a particular topic of the category, as identified from the web. This approach is validated with the help of titles and their categorizations obtained above. This approach could also be used extended for performing entity classification tasks for a Knowledge Graph in a task-specific manner.

3.5.1 Two-level Clustering-based Approach

We design an automated unsupervised approach to classify titles as simple/complex. Firstly, we construct a small new dataset across all categories, to validate our approach. In this dataset, complex titles are considered to be positive samples while simple Article titles are considered to be negative samples. This dataset comprises all the categories defined in the Generic class above. A random sample of positive sample titles from the Generic class dataset (i.e, Simple titles having their corresponding Wikipedia article) are considered as negative samples for this task of title classification. Positive samples in the above constructed complex titles dataset are considered to be positive samples for this task. The positive-negative sample ratio is 1:1.

Each sample in this newly constructed dataset consists of the title and its corresponding categorizations. For both types of titles, the corresponding Wikipedia categories at the end of their article are defined as their categorizations, as in the complex titles dataset. Additionally, a set of category-related keywords should be manually defined for each category, correlating with categorizations of simple category instances. The intuition behind defining these additional sets of category-specific keywords is to differentiate between patterns of category-related dependencies for Wikipedia pages with simple titles and complex titles.

We define a two-level clustering-based approach, which relies on syntactic and semantic features of titles, categorizations, and keywords, for separating the homogeneous set of simple titles from complex titles. This procedure is explained in detail as follows.

- Firstly, we generate a syntactic embedding for each sample in the dataset. This embedding is constructed from an initial TF-IDF [61] vector representation of the corresponding sample's categorizations defined above. The Wikipedia categories for each sample in the dataset are pre-processed and represented as a single TF-IDF vector.

- In the extracted TF-IDF representation from the sample's categorizations, vector components corresponding to the defined category keywords are extracted as the syntactic embedding. In this manner, only the encodings pertaining to pre-defined categorical keywords are retained, and other unnecessary sample-specific details are discarded. This is done to ensure that only discriminatory features are highlighted.

- For generating semantic embedding, we first obtain pre-trained semantic word embeddings of the samples categorizations (Word2vec [43] embeddings from spacy), for each sample in the dataset. After obtaining the initial semantic word embeddings, we compute

the word-word cosine similarity [22] between category-related keywords and words in categorizations, to gauge the semantic similarity in vector representations of the category's metadata and the sample's metadata.

- Mean-pooling of similarity scores is performed at the keyword level, i.e, for each keyword, the mean-similarity score across all categorization terms is computed as its vector component. Thus, the final vector representation, where each component corresponds to a keyword, is considered as the sample's semantic embedding.

- Both the syntactic and semantic embeddings are combined for each sample, and hierarchical clustering [46] is performed to group samples based on their categorizations and similarity with respect to category-related keywords. Euclidean distance metric is utilized for the clustering. We perform an exhaustive search for an optimal similarity threshold, by iterating between various values in the range of 0 to 1. The offset used for searching optimal threshold is dynamically varied, to maximize the Calinski-Harabasz Index [5], which was observed to be the best-performing metric while experimenting with clustering metrics and hyperparameters via a manual grid search.

- After clustering, the syntactic and semantic components of each cluster center are mean-pooled separately. This is done to aggregate the keyword-level encodings to represent sample-keyword similarity, which assists in the smoother functioning of further steps. Thus, a two-dimensional vector is obtained for each cluster center, which contains a syntactic score component and a semantic score component respectively.

- The syntactic score component of every cluster center is stored in a list and then normalized using min-max normalization [50]. These normalized values are used to update the corresponding syntactic score component for each sample. The same is performed for the semantic score component. The two-dimensional cluster center vector representation now contains corresponding normalized values, which have been updated separately for each component. Based on figure 3.4, complex titles (red points) have lower similarity scores with respect to category keywords, in comparison with simple titles (blue points).

- The mean of the syntactic and semantic score of each cluster center is defined as its score. This aggregated score is a measure that defines each cluster's samples based on its cluster center's score.

- A second level of clustering is performed, by utilizing these cluster centers as data points and their corresponding aggregate scores as single-component feature vectors. Like in the first level of clustering, an exhaustive search is performed for an optimal threshold score between 0 and 1, maximizing the Calinski-Harabasz Index. This second level of clustering is essential to obtain a more concise and robust partitioning of samples, with respect to their similarity of category-related keywords.

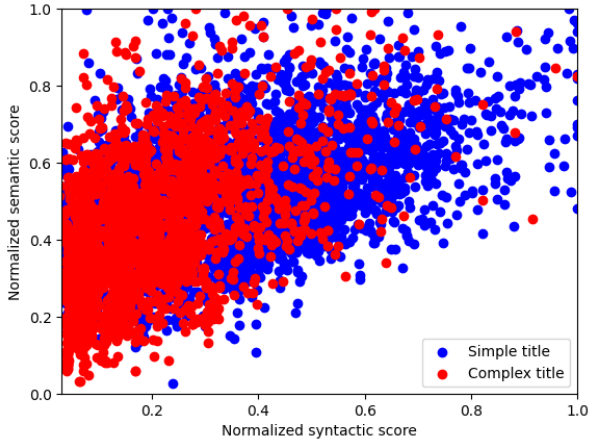

Figure 3.4: Titles' syntactic and semantic similarity scores (normalized)

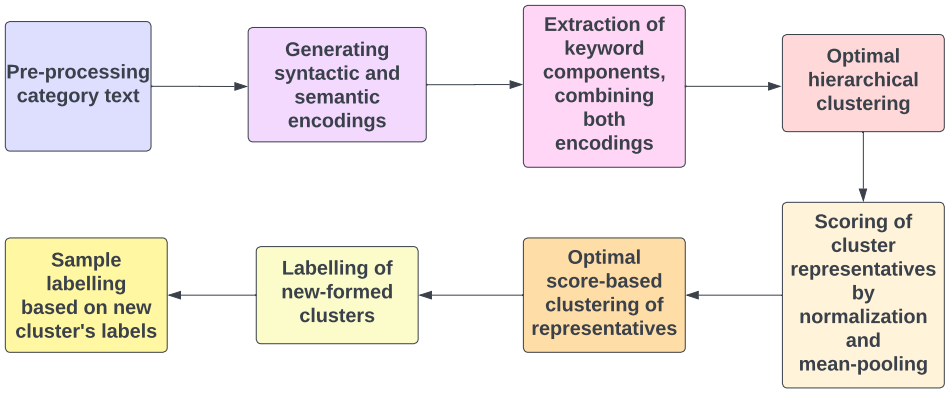

Figure 3.5: Title classification flow

- Cluster centers having this score less than the optimal threshold, are considered to have low similarity with category keywords, indicating that the dependencies are not as trivial as in the case of simple titles, and hence are classified as "Complex" titles. Similarly, cluster centers with high aggregate scores, more than the threshold, are considered to correspond and align to the standard category-instance format for that category and are classified as "Simple" titles. To label all the samples in the initial clustering, every sample of a particular cluster is assigned the same label as its corresponding cluster center's classification label.

A brief process outline of the entire procedure has been provided in figure 3.5. Thus, samples in the dataset are classified as Simple/Complex, based on the correlation between their categorizations and category keywords. This technique is generalizable to any new Wikipedia categories and does not require any manual annotation of titles as simple/complex. Further

details regarding the experiments conducted and results obtained are discussed in section 3.5.2.

3.5.2 Experiments and Results

In the unsupervised approach for title classification as Simple/Complex, experiments were conducted with respect to syntactic and semantic embeddings used. These experiments are enumerated below:

- **Only TF-IDF**: We utilize the complete TF-IDF vector representation as the syntactic embedding for performing the title-wise binary classification. Semantic embeddings are not considered in this case.
- **Only Mean word embedding**: We compute the mean semantic word embedding of title categorization text and category keywords, for each title, and use that as the semantic embedding in the approach. Syntactic embeddings are not considered.
- **TF-IDF, Mean word embedding**: We utilize both syntactic and semantic embeddings as discussed in the above two experiments, and plug into the defined approach.
- **Only Sentence transformer embedding**: As a replacement to the current semantic embedding, the title categorization text is passed through a sentence transformer model [57], and the encoding obtained is plugged in as the semantic embedding. Syntactic embedding is not considered.
- **TF-IDF, Sentence transformer embedding**: We utilize the complete TF-IDF vector representation as syntactic embedding and sentence transformer embedding as the semantic embedding, and plug into the defined approach.
- **Only Semantic keyword embedding**: We follow the idea of our approach which computes keyword-based semantic embeddings. Additionally, we discard syntactic embeddings in the representation.
- **Only TF-IDF keyword embedding**: We follow the idea of our approach which computes keyword-based syntactic embeddings from TF-IDF vector. We further discard semantic embeddings.

We have conducted a manual grid search to identify ideal metrics - error metrics such as Silhouette score [64], Davies Bouldin score [10], etc., clustering techniques, linkage types [27],

Table 3.4: Experiments results for Title Classification

Domain	Num. samps.	TF-IDF	Mean word emb.	TF-IDF, mean word emb.	TF-IDF, sent. trans. emb.	Sent. trans. emb.	Semant. key-word emb.	TF-IDF key-word emb.	Our Approach
Birds	596	0.453	0.441	0.458	0.448	0.419	0.607	0.883	**0.898**
Cities	796	0.592	0.535	0.594	0.606	0.771	0.81	0.758	**0.961**
Cricketers	564	0.592	0.576	0.589	0.589	0.652	0.746	0.787	**0.863**
Films	1298	0.342	0.362	0.347	0.313	0.287	**0.668**	0.594	0.546
Football people	1238	0.473	0.476	0.481	0.487	0.591	0.673	0.7	**0.733**
Indian film actors	652	0.63	0.567	0.629	0.647	0.81	0.597	0.908	**0.92**
Medicinal plants	1348	0.732	0.695	0.731	0.757	**0.929**	0.726	0.66	0.905
Tourist attractions	940	0.582	0.568	0.587	0.59	0.659	0.673	0.564	**0.785**
Universities and colleges	1370	0.599	0.641	0.609	0.609	0.731	0.632	0.9	**0.9**
Overall	8802	0.553	0.544	0.557	0.559	0.652	0.681	0.734	**0.817**

and their distance metrics [34], etc. to achieve optimal parameters. We also conducted experiments with just one level of clustering, and using different representations as above, but it was inferred that the two-level clustering approach with defined parameters achieved better results.

Our proposed mechanism for title classification has achieved a classification accuracy of 81.7% across all categories. It has achieved optimal results (Table 3.4) across all conducted experiments, which establishes how each step of the approach is essential to obtain the desired title label. This generalizable technique can be employed for any Wikipedia category to identify if a particular title is simple/complex. Further, it could be extended to perform similar node-classification tasks in a Knowledge Graph similar to the Wikipedia sub-structure, as entity-category embeddings are neatly captured.

3.6 Summary

We have defined the various types of article titles that exist in Wikipedia and constructed datasets accordingly for each such type and sub-type. We defined the notion of simple and complex titles based on their dependencies with a particular category, and distinguished Wikipedia pages with simple titles further into Generic/Abstract classes, based on their nature. The Generic class dataset consists of nearly 30K samples across 9 categories, and the Abstract class dataset consists of nearly 5K samples across 5 categories. The Complex titles dataset comprises about 9K samples pertaining to 9 categories of the Generic class. We have made these datasets publicly available. We also design an unsupervised, generalizable, automated approach, to classify titles as simple/complex, irrespective of their category. This classification acts as the first step in arriving at a Notability label for a title. The techniques used for designing the

datasets, and performing title classification could assist with similar tasks in the context of other Knowledge graphs by capturing entity-category dependencies.

Chapter 4

Generating category-specific embeddings for Notability detection of Article titles with Simple Named entities

4.1 Overview

In this chapter, we describe the system we designed to construct entity embeddings for performing Notability detection of simple titles. Initially, we discuss the baselines used and then describe the feature extraction procedures, which contain how several relevant web-centric features have been collected for entities of the Generic class and the Abstract class to obtain entity embeddings. These web-based components, from which text-based salience features are extracted, include reliable web sources for entity-related documents, Wikipedia ecosystem, query logs-based analysis, presence in social media, and relevance in online news websites.

We then explain our final classification architecture, which consists of neural networks and BERT encodings (transformer encoder) to obtain entity embeddings for binary classification. For validating our system's performance in this task, we utilize accuracy metrics, correlation analysis, ablation study, and prediction confidence on popular Wikipedia pages. The limitations and bottlenecks of the system are also discussed briefly.

4.2 Baselines

We utilize two models as baselines to compare and validate the effectiveness of our system. These two models are chosen as the appropriate baselines, based on their usage in similar downstream tasks, scalability, and generalizability. Further, these baselines adhere to the definition of Notability.

4.2.1 Handcrafted entity-salience features: SGD

Yashaswi et al. [51] targeted the exact problem of Notability Determination, by focusing on the two key aspects of the definition of Notability - reliability and significant coverage. We consider this to be the first baseline approach to validate our model. This approach was implemented for the category of Indian Film Actors. It consists of two major steps.

- **Reliable web domain identification**. Top reliable web domains for a category are found (based on the web presence, number of entity profile pages it comprises, etc). For a web domain and a given entity, it is recorded whether there is a profile page for the entity in this web domain. For instance, it is verified if an entity like Shah Rukh Khan (actor) has a profile page in a web domain like "imdb.com".

- **Entity salience computation**. Handcrafted features are defined to capture entity salience from a body of text, for each entity and its related documents. These features include the number of entity mentions in the whole text, mentions count in the first three sentences, occurrence in the first sentence, and index of the sentence in which the entity occurs for the first time. The text from the entity's profile page in the above web domains is analyzed with the help of these features.

Note that the features defined here are also considered "domain-specific" features, as their values vary across web domains. These heuristic-based numeric features are passed through a Stochastic Gradient Descent [32] classifier. This baseline is referred to as "SGD" in further sections.

4.2.2 Word-embedding Semantic-similarity based classification

We implement a second baseline, which focuses on the semantic nature of entity-related content to perform classification, rather than syntactic features as in the previous baseline. It is centered around the idea described in the work of Zhang et al. [75], which deals with semantic similarity-based encodings [69]. Initially, a query is formulated comprising entity name, category name, and category-related keywords. Word-word semantic similarity is computed between query terms and terms from each of the above reliable web-domain documents. These similarity encodings are combined and passed through a feed-forward neural network to perform classification. This baseline is referred to as "Semantic embeddings" in further sections.

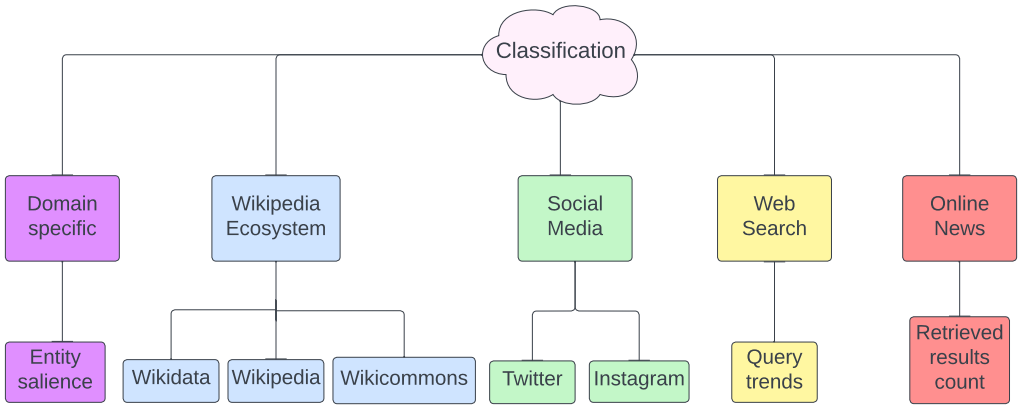

Figure 4.1: Feature hierarchy defined for the Generic class

4.3 Feature extraction from Web-components to capture salience signals

Both of the described baselines focus on specific ways to encode entity-related document content extracted from web domains. However, there exist other relevant web components which have not been explored in the context of entity-specific content analysis. We design web-based features in such a manner that, they capture significant key signals about an entity with respect to the web.

As discussed in the construction of the dataset, simple titles comprise two sub-divisions - Generic Class and Abstract class. We define different types of features extracted for embedding generation in both Generic and Abstract classes. For the Generic class, the defined feature sets and their design choice are given below (also refer to figure 4.1).

- **Domain-specific features** correspond to hand-crafted entity-salience metrics. These features are similar to the baseline features (discussed in section 4.2.1) and are important to capture as they contain profile pages for entities with detailed information. The quality and quantity of the information indicate the popularity range and importance of an entity.

- For the same reason as above, the **Wikipedia ecosystem** is also targeted. We look at structured data (Wikidata), related articles (Wikipedia), and image content (Wikicommons) to obtain a consolidated understanding of an entity with respect to this verifiable encyclopedic platform.

- The popularity of an entity on **social media** provides insights about its interest levels at a point in time. This aspect is captured by examining the follower count on Twitter, and Instagram (although applicable only for people/organizations, etc).

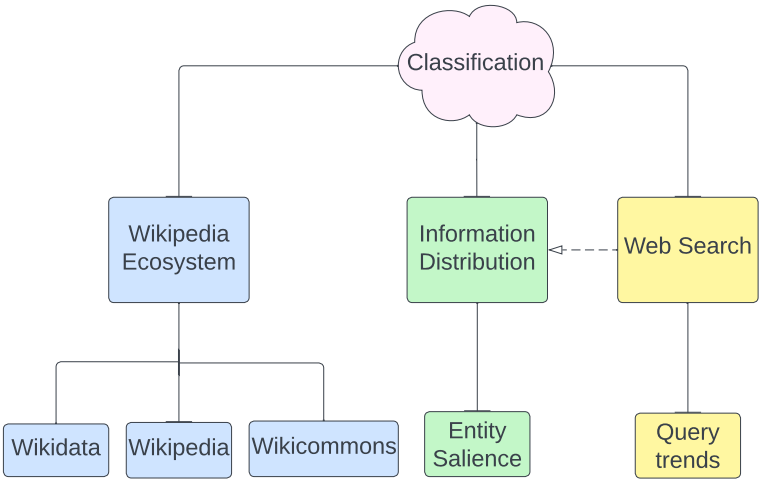

Figure 4.2: Feature hierarchy defined for the Abstract class

- An important signal with respect to an entitys importance is regarding the extent to which its information is accessed/searched for by the general public, at any given period of time. This is captured by **query logs** analysis (Google Trends).
- The available **news content** for an entity on top international news web domains also plays a role in identifying an entitys importance.

For the Abstract class, a similar feature set is defined (refer to figure 4.2). The above reasoning applies to feature sets such as query logs and the Wikipedia ecosystem. Note that by definition of an Abstract class, there are no pre-determined web domains for entity-specific data collection. Alternatively, **"information distribution"** about the entity on the web is analyzed (explained in section 4.3.2). Social media and online news-based features are present in the Generic class, but not in the Abstract class, because of their inapplicability to such concept-based entities in general. For example, the followers count for a biological concept like "Temperature" would not exist.

All of the features that are a part of the final constructed entity embedding are discussed in detail in the subsequent sections. We make the code publicly available[1].

4.3.1 Domain Specific features (Generic Class)

This sub-part of features is similar to the domain-specific features described in the baseline. Additionally, the feature set is augmented with more information about the content of a web page in general. The set of features extracted here is based on an entity's content availability in reliable web domains of that category (such as IMDb for Film actors) and coverage of the

[1] `https://anonymous.4open.science/r/Notability-Detection-System-EB4E`

entity in this content. This set of features provides a concrete understanding of entity-specific content, in comparison with other web-based signals.

We define two procedures, to identify reliable web domains, and to compute the Jaccard threshold[2] [47] for entity-entry matching in each web domain. We mention certain numeric parameters and their values in these procedures, wherever used. Such parameters include the number of retrieved results, the number of web domains, and the sample count. Note that the values of these parameters were tuned based on a manual grid search. It was observed that lower values for these parameters provided insufficient content to analyze, while higher values required more data collection effort and introduced noise.

For defining a procedure to identify reliable web domains, we use the intuition that such reliable domains comprise concrete information about many entities belonging to the category. Blogs and social networking sites are filtered out initially, as the focus here is on concrete, credible, and verifiable textual information about the entity, which is not guaranteed in these cases. The procedure for identifying web domains is given below.

- For every entity, we perform a web search on a search engine (Google here), by formulating queries including the title of the entity, its category, and the keywords "profile" and "text". This is done to prioritize those web domains with significant text in entity profile pages. The top 6 retrieved results are considered, and the corresponding web domains' frequency is increased by 1 (for one entity).

- The external links section in Wikipedia articles of notable entities are examined, and the frequencies of web domains mentioned here are also increased by 1.

- Frequencies of web domains are computed from the above 2 tasks (processing each entity of that category), and a weighted average is computed to score each web domain. Thus, a web domain with higher score in both aspects has more probability of being chosen as the reliable web domain.

- Along with the frequency-based score, we also store the average rank of occurrence of a web domain in the top 6 retrieved results. The rank indicates the domain authority of that web domain[3] while retrieving results for the given query.

- The required number of web domains (4 here) with the highest frequency-based scores, further sorted by domain authority rank, are chosen as the reliable web domains. The web domains are manually vetted, if necessary.

On identification of reliable web domains for a given category, an entity's presence in each such web domain is recorded. This is a boolean attribute, which is 0 if the entity's entry is not

[2]https://en.wikipedia.org/wiki/Jaccard_index

[3]https://en.wikipedia.org/wiki/Domain_authority

found in that particular web domain, and 1 otherwise. To identify if an entity's entry exists in a web domain, we use a similar syntactic matching technique as used in matching if a title has a Wikipedia page. However, the nature of web domains varies across categories. Hence, we design a detailed heuristic-based approach. We first compute a threshold that varies across web domains, called the Jaccard threshold. This threshold has been computed in the manner defined below (for a given web domain).

- A sample of 100 notable entities in the dataset is selected, and a web search is performed with the query consisting of the entity's title and the web domain's name, and keywords "profile" and "text" (as above). This notable entity sample is used to set a threshold that is to be achieved for an entity to have a corresponding page in the web domain, as notable entities most likely have a page.

- The top 3 retrieved results are examined and results not corresponding to the required web domain are discarded.

- In the filtered URLs, the title of the web page is extracted, and all of its skip n-grams [24] are enumerated ("n" is the number of tokens in the entity name), to consider all combinations of tokens for matching. A Jaccard similarity score is computed, by comparing each of these skip n-grams with the entity name. The maximum score observed across all skip n-grams is treated as the matching score between the webpage's title and entity name. Note that skip n-grams are used for the title of the web page to ensure that other undesirable tokens do not affect the entity entry-checking process.

- The above score is computed for all the filtered URLs and the maximum value is recorded.

- After all the maximum values are recorded for all entities in the sample, the average of the least $K\%$ (chosen to be 33 based on observations) Jaccard indices is chosen to be the Jaccard threshold for that web domain in that category. The significant percentage (K) considered is necessary to alleviate any noise in the lower range of values, making the threshold computation robust.

For any entity of a given category, we compare the Jaccard similarity of the entity name's tokens and web page title. If the maximum score of the top retrieved results exceeds the threshold, the corresponding web page is chosen to be the relevant entry for that entity, in the web domain. Note that this could lead to noise in the data causing false matches. However, the robust design of the threshold computation procedure has ensured that such cases were insignificant.

In cases where the search engine is not able to process requests, manual intervention is necessary to verify the identified reliable web domains, find the entry for the entity from the web domain directly, and extract only relevant text from the web domain's webpage format.

Apart from the reliability-based boolean feature, we also consider the coverage of the entity in its corresponding web page in the web domain. These features are extracted after filtering boilerplate text [33] and performing co-reference resolution[4]. Some of the features were directly used from the baseline, while new heuristic-based features such as domain authority, number of relevant images in the profile web page (containing at least one token of entity's title tokens in its label text), and size of content are additionally incorporated. All of the entity salience features are defined in Table 4.3.1. Thus, syntactic entity-related features - corresponding to reliable web domain and entity-salience aspects indicating coverage, emphasize on the two primary criterion mentioned in the definition of notability. These criteria also play an essential role in summarizing entity-specific signals on the web, which could be leveraged for entity embedding generation and related downstream tasks. However, note that web domains are absent for the Abstract class, and hence an alternative approach is to be defined for imitating these domain-specific features. This approach is discussed below.

Category-specific entity salience features for a web domain

Feature	**Description**
present	Check on entity's entry's existence in web domain
pos-score	Average rank of web domain in top 6 search results
sent-count	Number of sentences in web page text
imgs-count	Image count in web page including entity's mention
E_p	Check on named entity's mention in first sentence
E_i	Number of entity's mentions in first 3 sentences
E_f	Index of sentence containing entity's first mention
E_h	Total number of entity's mentions in text

4.3.2 Information Distribution on the Web (Abstract Class)

As discussed in the definition of Abstract class, reliable web domains cannot be identified for categories corresponding to the Abstract class. Hence, we define an alternative generalizable mechanism for analyzing entity-specific content for embedding generation of such abstract entities and thereby detecting their notability. This set of features pertains to "Information Distribution", as it primarily gathers features corresponding to the distribution of content about an entity on the web. This is similar to extracting content from a specified set of reliable web domains and utilizes entity salience features such as E_p, E_f, E_i, and E_h.

[4] `https://spacy.io/universe/project/neuralcoref`

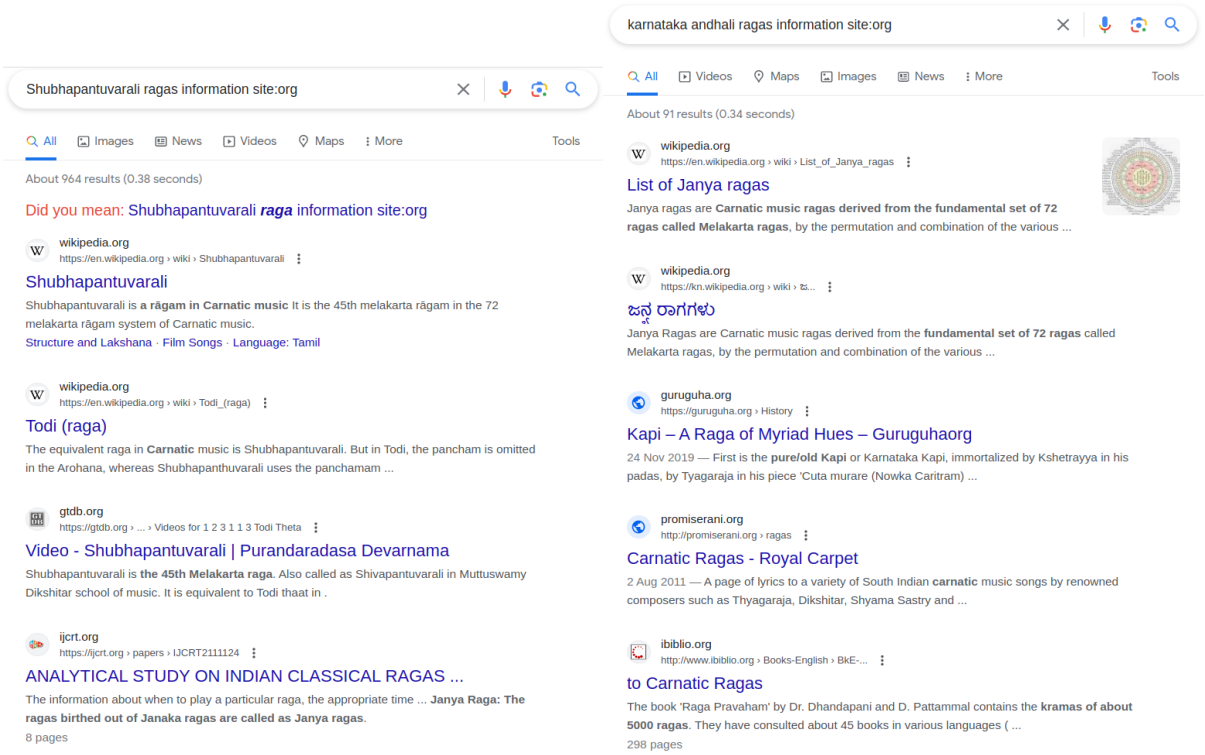

Figure 4.3: For a notable title (left), there is more relevant and entity-centric information corresponding to it on the web in independent sources, in comparison to a non-notable title (right)

The intuition behind the feature set is that if an entity is notable and satisfies corresponding guidelines, its coverage on the web is high, and thus more documents centered around the entity are retrieved by performing a web search. The following steps are followed for extracting this set of features.

- A search query is formulated, with entity name, category name, and the keyword "information", and a search is performed on a search engine (Google here). Note that these steps were performed for two types of queries - one of them limiting retrieved results to the site extension ".org" to ensure more reliability in content retrieved (as information displayed is from reliable organizations). The other type of query does not have any such restrictions to reflect generic content corresponding to the entity on the web.
- The retrieved documents are examined and the top 15 results are considered and partitioned into 5 sets in the order of retrieval. It follows that the first 3 results belong to the first set and the last 3 results form the fifth set. The values for parameters such as the number of results, and the number of sets, are tuned such that there is sufficient and relevant content about the entity to analyze.

- For documents corresponding to each set, the document with the most amount of text is picked, to ensure that the document consists of significant analyzable textual information about the entity. On performing this check, the possibility of a short article is eliminated. Filtering is performed to ensure that the documents do not belong to the social media platforms such as Facebook, Twitter, Instagram; or blogs, as the emphasis here is on concrete entity-centric information.

- Entity salience features such as E_p, E_f, E_i, and E_h (discussed above) are extracted for the text obtained above. Note that 5 documents are analyzed for both types of queries, resulting in 10 different text-rich documents, each representing a key partition of retrieved results by the search engine.

An example of the execution of this procedure is illustrated in figure 4.3, where we can observe the search results from reliable organizations (with site extension ".org") for two examples from the "Ragas" category in the dataset of Abstract samples. We observe more relevant and title-centric web documents for the notable title, while mostly irrelevant documents are observed for the non-notable title. Thus, the extent of content about the entity on the web is captured, which is essential for embedding generation. It was observed that the relevant documents - both from reliable organizations and generic web sources, provided a good range of content for entities pertaining to the Abstract class, replicating the effect of web domains as in the Generic class.

Apart from these entity-centric web documents which are also used in the baselines, other key components of the web are also explored, which provide useful insights for extracting entity-related signals and deciding on their notability. These components and their features are discussed below.

4.3.3 Wikipedia Ecosystem

The presence of an entity in the Wikipedia ecosystem, even if it does not have its own page, is assumed to contribute significantly to its notability. For both classes, we examine the entries of entities in three primary components in the Wikipedia ecosystem - Wikidata, Wikicommons, and Wikipedia. We design procedures for extracting features from each component, which assist in identifying if there is sufficient coverage. In these procedures, we further define values for numeric parameters such as the number of retrieved results analyzed. These values are defined and tuned in a similar manner as in section 4.3.1, as the same reasoning applies here.

In Wikidata, a search is performed via a search query (text) comprising the entity's name and category, to identify most relevant retrievals with respect to the entity. The top 7 retrieved results are analyzed. In these retrieved documents, a document is considered relevant if it contains at least one token of the entity's name and at least one of the related keywords of

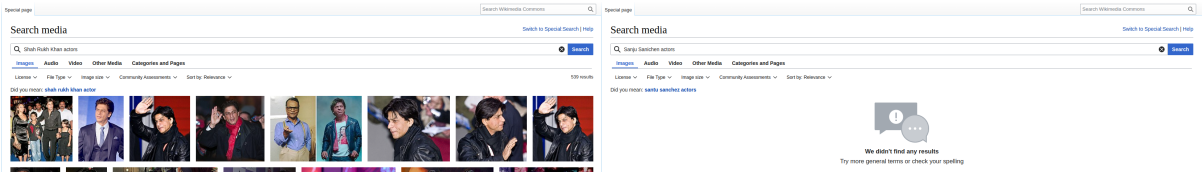

Figure 4.4: For a notable actor (left), there are more relevant images on Wikicommons in comparison to a non-notable title (right)

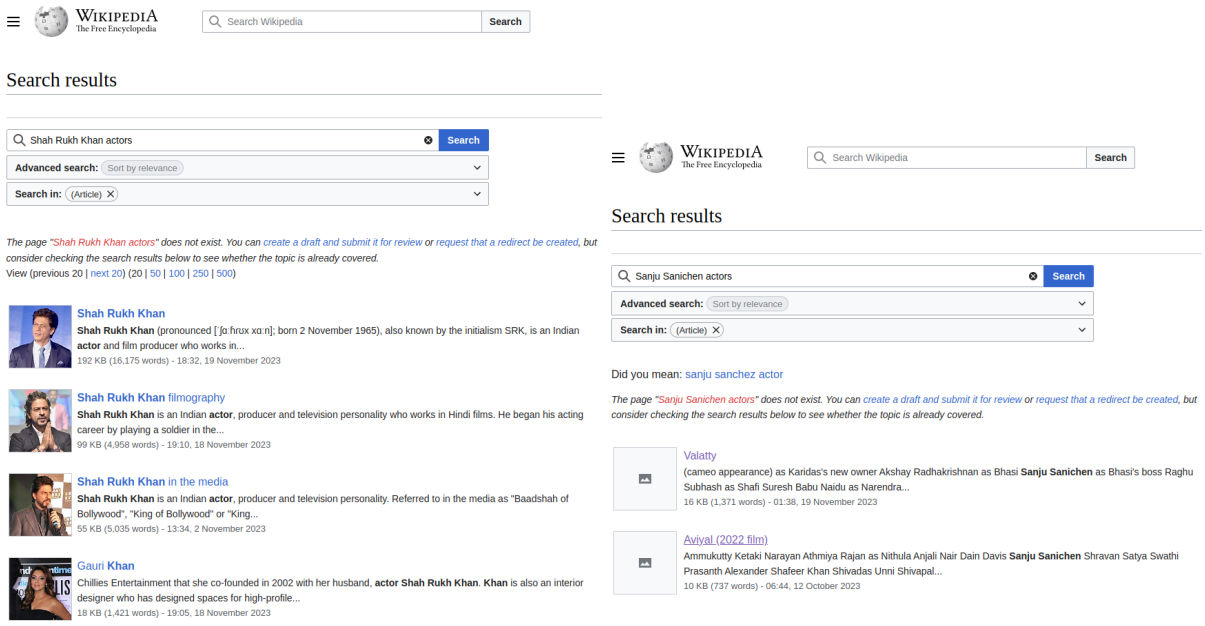

Figure 4.5: For a notable actor (left), there are more relevant Wikipedia articles mentioning their name, in comparison to a non-notable title (right)

the category (such as "Actor", and "Movie" for the "Film actors" category). The count of such documents in these results is recorded as a Wikidata count feature.

Further, the same search is performed in the Wikicommons repository, and the count of images retrieved is stored as a feature. In figure 4.4, it has been illustrated how the count of relevant images varies for two examples of The "Actors" category. For a notable actor like "Shah Rukh Khan", there are many relevant images in comparison to a non-notable actor like "Sanju Sanichen", who does not have any images in Wikicommons.

In the case of Wikipedia-based feature extraction, similar to the Wikidata count-based feature, a thorough search is performed for the entity including its title and category name. The top 10 relevant results are iterated one by one, and the first 3 relevant documents encountered are chosen for analysis, to ensure maximum relevance and sufficient content to gauge entity salience. A document (Wikipedia article) is considered relevant if it contains at least

one category-related keyword in its Categories section. For the 3 chosen documents, relevant entity salience features such as E_f, E_i, and E_h are computed. Other entity salience metrics are discarded because of their inapplicability, as the web domain is fixed and the nature of the content is pre-defined.

In figure 4.5, it has been illustrated how the mentions in relevant Wikipedia articles vary for two actors (same example as in Wikicommons - Shah Rukh Khan and Sanju Sanichen). However, despite the absence of a Wikipedia article for Sanju Sanichen, it can be observed that two Wikipedia articles mention the name. This indicates how the information from the Wikipedia ecosystem plays a vital role in determining a topic's notability, irrespective of whether it has a Wikipedia article or not. This further speaks for the entity-related signals from reliable platforms on the web, which are captured by our system and further processed for embedding generation.

4.3.4 Query Logs

Apart from the documents in specific web domains and reliable platforms, entity-related signals can also be found in the general public's interest in an entity, as reflected in their queries in search engines. This is an important signal with respect to an entity's importance and assists with Notability detection as it indicates the extent to which its information is accessed/searched at any given period of time.

Query log features of a title pertain to frequency analysis of search queries regarding the title, in a search engine (Google here). This feature set is designed based on Wikipedia's notion of notability that if an entity was notable at a period of time, it is always considered notable. Further, the temporal aspects regarding the interest around the entity are captured here.

We incorporate the query-log signal by including a set of aggregated scores from Google Trends[5] [7], describing the historical data about queries corresponding to an entity. For a given query consisting of the entity name and category, we can obtain a general interest score for it over a period of a month. This score was obtained for the months between 2004 and 2022. The aggregates of the scores such as minimum, median, maximum, mean, first, and third quartiles were recorded, to represent the query logs data about the interest over an entity in a concise manner.

The difference in interest level scores for two films from the dataset is illustrated in figure 4.6, where the interest scores are higher for the notable film "The Shining", and there is negligible interest regarding the non-notable film "Operation Merry Christmas:The Elf Con".

[5] `https://pypi.org/project/pytrends/`

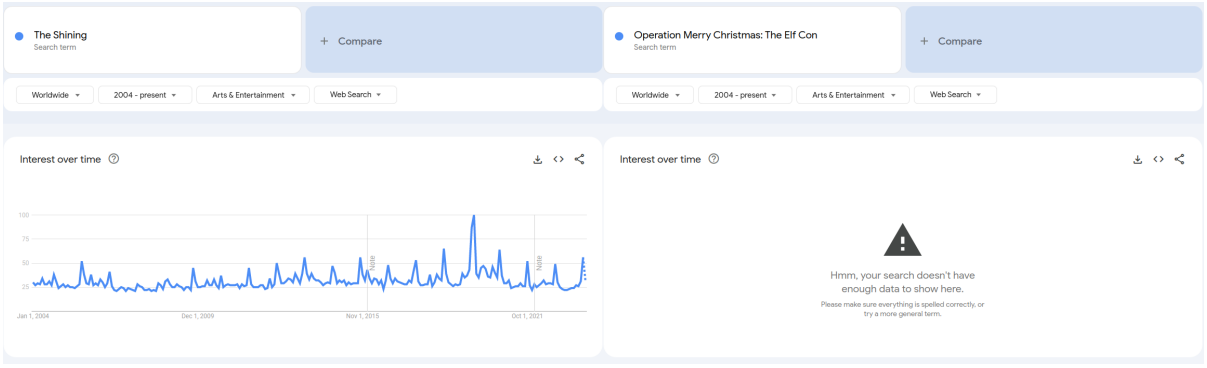

Figure 4.6: For a notable film (left), higher interest levels are recorded for a longer period of time in Google trends, in comparison to a non-notable film (right)

4.3.5 Social Media (Generic Class)

Another key metric indicating the public's interest in an entity similar to query logs is the popularity of an entity in social media platforms. It plays a significant role in indicating the interest for an entity in a period of time, especially for categories corresponding to people/organizations. This signal is captured by recording the followers count of entities on Twitter [63] and Instagram platforms. The Jaccard-index-based approach defined in section 4.3.1 is used for performing entity name-profile matching.

However, it is to be noted that this feature set does not directly correspond to notability guidelines to the extent of previous feature sets. These features are also not applicable in all scenarios, such as in categories of the Abstract setting, or categories such as Medicinal plants in the Generic setting.

4.3.6 Online News (Generic Class)

The available news content for an entity on top international news web domains also plays a role in identifying an entity's importance. Thus, a small set of such web domains is manually curated based on their generalizability to a variety of categories. These web domains are the New York Times (NYT), The Sun, American Broadcasting Company (ABC), Cable News Network (CNN), and Channel News Asia (CNA).

For a given entity, we perform a search on these news web domains, by formulating queries including entity and category name. The counts of retrieved results in each of the web domains are recorded as features. These features are not considered for categories of the Abstract class, due to their inapplicability to such concept-based/abstract entities.

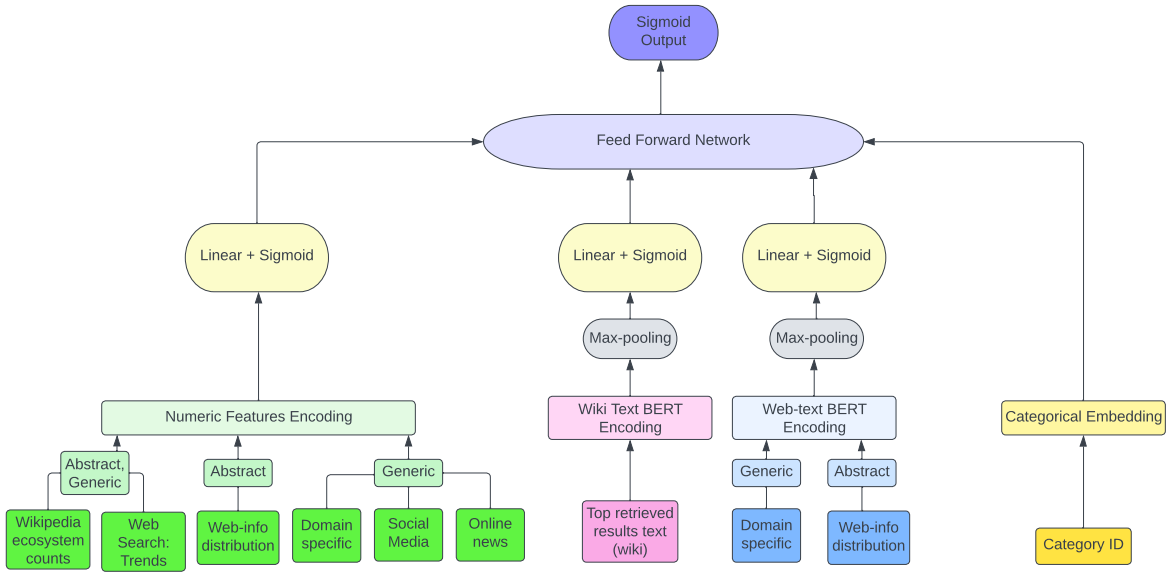

Figure 4.7: System architecture for generating category-based entity embeddings for classification

4.4 Category-specific embeddings: Web-based count features + BERT

Based on the above set of features extracted from different web components, an architecture is designed to obtain final entity embeddings capturing its associations with its category, and further perform the final binary classification task to arrive at the Notability label. The architecture is defined such that it takes the corresponding class's (Generic or Abstract) parameter set as input, thereby utilizing only features relevant to the corresponding class, to perform binary classification. A consolidated understanding of the entire classification architecture is provided in figure 4.7. This architecture's design is explained in detail below.

Initially, all the numerical features extracted above from the web-based components are combined to obtain a numeric feature-based encoding, defined as n_e, for the entity e. The encoding includes a concatenation of the web-based count-type features extracted and acts as a consolidation of that information, summarizing the corresponding web-signals. This numerical encoding is passed through a linear layer, with sigmoid activation, and updated encoding N_e is obtained. This encoding is an indication of numerically quantized data corresponding to the data on the web, and does not take text into account.

$$N_e = \sigma(W_n^t n_e + b_n) \tag{4.1}$$

Note that entity-related textual information has also been extracted for both classes to perform in-depth entity-salience analysis. For the Generic class, there are a total of 7 documents

with text, based on the format of data collection. This includes 4 documents corresponding to text from the entity's profile page in 4 reliable web domains, and 3 most relevant documents corresponding to the entity in Wikipedia. Similarly, the Abstract class consists of 13 documents with text. This includes 10 documents indicating information distribution from the web (5 each for two types of queries formulated - restricted on reliability, and general content), and 3 most relevant documents corresponding to the entity in Wikipedia. The justification for choosing these values for parameters is given above. For all documents, we process the information further, rather than relying only on heuristic salience measures.

The information corresponding to the entity salience of a body of text is captured by obtaining its encoding from a transformer encoder, which is effective in such representation-based tasks. Each of the bodies of text is passed through the BERT model (Bidirectional Encoder Representations from Transformers) [12], which encodes the corresponding text by paying attention to key parts of the content. We used Bert-base-uncased which was fine-tuned in the process of training.

For a given entity e, the i^{th} relevant Wikipedia document's BERT encoding is represented by w_{ei}; the j^{th} reliable web domain's text encoding in the Generic class or analogously the j^{th} partition's document in the Abstract class is represented by t_{ej}. Max-pooling is performed on all w_{ei} and t_{ej} encodings to preserve key information and reduce dimensionality. New salience encodings W_{ei} and T_{ej} are obtained by passing the w_{ei} and t_{ej} encodings through linear layers with sigmoid activation. Further, each of the w_{ei} encodings is unified to obtain a single representation of text collected from Wikipedia, defined as Wk_e. Similarly, the unified text representation of all t_{ej} encodings is defined as T_e. These unified text representations encode key entity-related signals in the documents extracted from corresponding web domains/platforms for each class.

$$w'_{ei} = maxpool(w_{ei}) \tag{4.2}$$
$$t'_{ej} = maxpool(t_{ej}) \tag{4.3}$$
$$W_{ei} = \sigma(W_w^t w'_{ei} + b_w) \tag{4.4}$$
$$T_{ej} = \sigma(W_t^t t'_{ej} + b_t) \tag{4.5}$$
$$Wk_e = \cup_{i \in I} W_{ei} \tag{4.6}$$
$$T_e = \cup_{j \in J} T_{ej} \tag{4.7}$$

The classifier model is further enabled to identify the differences in the nature of features for different categories, based on categorical embeddings [11] [23]. The categorical embedding for a category c, is defined as e'_c, which is obtained by passing the unique integer identifier of the category id_c to an embedding layer, represented by the embeddings matrix Emb_c.

$$e'_c = Emb_c(id_c) \tag{4.8}$$

All of the outputs such as numerical encoding, salience encodings, and categorical embeddings are concatenated to find the final entity embedding E'. This entity embedding E' comprises of key web-based signals, which are useful to establish interconnections between entities and categories in the Wikipedia KG. This unified entity embedding E' for each sample is passed through a Feed Forward Neural network comprising three linear layers. The final output, defined as o, is of a single neuron to which sigmoid activation is applied (to obtain a value ranging between 0 and 1), which indicates the score assigned by the system to the entity regarding its Notability. Further specifications regarding the model definition and training are described below.

$$E' = [N_e, Wk_e, T_e, e'_c] \tag{4.9}$$

$$o = \sigma(FFN(E')) \tag{4.10}$$

4.4.1 Experimental Setup

The percentage-wise train-test-validation split of the data was 70-15-15. For the Generic class, the classifier was trained for a total of 40 epochs, with a batch size of 24 and a learning rate of 0.0001. The word count of the text attributes of the entity has a limit of 300 words. The vocabulary was restricted to 50k words (first 50k unique tokens encountered), and categorical embeddings were assigned a dimension of 150. The Abstract class has a similar architecture and hyperparameters, but it was trained for 25 epochs with a batch size of 6, based on computational constraints, and speed of training. Binary cross-entropy is the loss function used. These hyperparameters were found to obtain optimal results, in comparison with other variations tried.

Experiments were also performed by replacing the BERT model with other potential alternatives for creating encodings, which are suitable for such tasks. This was done to validate the effectiveness of the usage of a transformer-encoder architecture such as BERT to generate entity-salience encodings. The attempted alternatives are listed below:

- **1D Convolutional Neural Networks**: Salient aspects of content is captured on defining appropriate 1D filters to process document text.

- **Match-LSTM type network** [70]: Gauges the similarity between entity-related metadata comprising entity name, category, and textual documents at a sequential level.

BERT-based architecture significantly outperformed these architectures and hence was chosen for the final system to generate entity embeddings, validating its effectiveness for such tasks. The results of the baseline approaches, experiments, and our defined system are compared in detail for an overall analysis. This is explained in section 4.5.

Table 4.2: Metrics for baselines, experiments, ablations and final system

Generic class					Abstract class				
System	**ACC**	**PR**	**REC**	**F1**	**System**	**ACC**	**PR**	**REC**	**F1**
SGD	0.7821	0.7844	0.7821	0.7817	SGD	0.6186	0.6745	0.6667	0.6181
Semantic embeddings	0.8232	0.8274	0.8232	0.8226	Semantic embeddings	0.6585	0.7121	0.5577	0.5088
Match-LSTM	0.8107	0.8402	0.8108	0.8065	Match-LSTM	0.7738	0.7789	0.7302	0.7408
1D-CNN	0.8459	0.8493	0.8459	0.8455	1D-CNN	0.8093	0.7997	0.7906	0.7945
- All count features	0.8041	0.8198	0.8042	0.8017	- All count features	0.6763	0.7695	0.5777	0.5376
- Wikipedia counts	0.8302	0.845	0.8302	0.8283	- Wikipedia counts	0.7517	0.8084	0.6816	0.687
- Categorical embedding	0.8767	0.8768	0.8767	0.8767	- Categorical embedding	0.8448	0.8549	0.8135	0.8263
- Domain features	0.8771	0.8779	0.8771	0.877	- Site:org features	0.847	0.8464	0.8244	0.8327
- Trends counts	0.8797	0.8801	0.8797	0.8796	- Trends counts	0.8514	0.8429	**0.8405**	0.8417
- News features	0.8819	0.8835	0.8819	0.8818	- Generic search features	0.8514	0.8514	0.8291	0.8375
Final System	**0.8848**	**0.8855**	**0.8848**	**0.8848**	Final System	**0.8581**	**0.8596**	0.8356	**0.8446**

4.5 Results and Discussion

Since Notability detection is a binary classification task, standard accuracy metrics such as Precision, Recall, F1 score, and Accuracy are used for comparing the performance of various architectures and experiments. To ensure fairness in validation across experiments, it was ensured that the test set follows a similar distribution as in the entire data collected, approximating the real-world data availability for varying categories in the dataset. It was observed that the performance accuracy increased by nearly 7% for the Generic class and 20% for the Abstract class, in comparison with both the baseline approaches (refer to table 4.2), indicating the additional signals incorporated in our defined system and how our defined approach to extract entity embeddings was effective in the task of performing Notability detection for Wikipedia. Ablation study, correlation analysis, and validation on WikiProject popular pages were conducted to understand and establish the superior performance of our system.

4.5.1 Ablation Study

Ablation experiments were conducted by removing components from the system (only one at a time), to better analyze the impact of individual component contributions and validate their applicability. For both classes, we can observe that all the web-based count-features (numerical encoding N_e as described in section 4.4) and Wikipedia-count-based features played a significant role in the classification process (table 4.2), justifying their presence in the final entity embeddings. Other components such as categorical embeddings, online news features, etc. had relatively weaker contributions, as can be observed from the minor difference in performance on their removal. In order to understand feature contributions at a more granular level, correlation analysis is performed.

Table 4.3: Pearson correlation coefficients for numeric features

Generic class							
Attribute	Score	Attribute	Score	Attribute	Score	Attribute	Score
wikidata-docs-count	0.553	Wikipedia-Ef-0	0.406	Wikipedia-Ef-1	0.336	Wikipedia-Ei-0	0.328
Wikipedia-Eh-0	0.317	Wikipedia-Ef-2	0.283	dom3-present	0.28	Wikipedia-Ei-1	0.253
Wikipedia-Eh-1	0.248	dom3-Ef	0.229	dom3-Ep	0.229	trends-max	0.227
Wikipedia-Ei-2	0.215	Wikipedia-Eh-2	0.215	dom4-Ef	0.198	dom4-sent-count	0.192
dom4-Ep	0.186	trends-mean	0.178	trends-3/4	0.177	dom4-present	0.169
dom4-Ei	0.162	trends-med	0.16	dom3-Ei	0.153	trends-1/4	0.151
trends-min	0.106	dom3-sent-count	0.104	dom3-imgs	0.101	dom4-imgs	0.099
Abstract class							
Attribute	Score	Attribute	Score	Attribute	Score	Attribute	Score
wikidata-docs-count	0.515	org-rel-count	0.282	org-d1-Ef	0.204	gen-rel-count	0.189
org-d4-Ep	0.182	org-d4-Ef	0.182	trends-max	0.274	Wikipedia-Ei-0	0.219
Wikipedia-Ef-0	0.209	Wikipedia-Eh-0	0.208	org-d1-Ep	0.182	gen-d1-Ef	0.173
gen-d3-Ef	0.166	gen-d3-Ep	0.165	org-d1-Ei	0.164	trends-mean	0.161
trends-3/4	0.155	org-d5-Ef	0.148	gen-d1-Ei	0.144	gen-d2-Ef	0.143
org-d3-Ep	0.142	org-d4-Ei	0.141	org-d5-Ei	0.141	org-d3-Ef	0.14
gen-d3-Ei	0.138	gen-d1-Ep	0.136	Wikipedia-Ef-1	0.135	org-d3-Ei	0.134
gen-d2-Ep	0.129	org-d2-Ef	0.129	trends-med	0.128	org-d2-Ep	0.127
gen-d4-Ef	0.12	gen-d2-Ei	0.12	Wikipedia-Ei-1	0.118	trends-1/4	0.114

4.5.2 Correlation Analysis

Correlation analysis is performed to validate the applicability of defined web-based features and understand why the model performs better for both classes. The Pearson correlation coefficient [71] was computed for each of the numerical features, with respect to the notability label of entities. These correlation scores obtained for both classes are enumerated in table 4.3. Note that only attributes with relatively higher scores (at least 0.1) are displayed, indicating that these attributes had a significant role to play in the classification process and hence represented more essential parts in the entity embedding.

It can be observed that the Wikipedia ecosystem-based features have a superior correlation score for both classes, especially the relevant document count of Wikidata. The handcrafted entity-salience metrics of relevant Wikipedia documents and reliable web domains / selected "information distribution" documents also achieve relatively higher correlation scores. The high-correlation scores of document-based entity-salience analysis imply that these salience-based features were effective during classification, and are essential components in the entity embedding. It further establishes the superior performance of our model, as it analyzes these relevant document text encodings using the attention mechanism.

Among other web-component-based features, Query logs have a relatively lower, albeit significant correlation with respect to the above-document-based entity salience features. For both classes, we observe "trends-max"[6] to have a higher correlation, indicating that an entity is still

[6]the maximum popularity score for an entity over a month

notable if it was notable at a point in time (which is mentioned in the definition of Notability for Wikipedia). These features capture the temporal aspects related to the entity in its embedding. The other Google Trends-related score aggregates: minimum, mean, median, and quartiles (1/4, 3/4); have significantly lesser correlation than "trends-max", indicating how the dynamic nature of an entity's popularity affects its notability labeling.

However, certain web-component features did not seem to improve the effectiveness in the performance of the system, as they exhibited lower correlation coefficients. Features corresponding to online news websites and social media platforms, for the Generic class, were observed to have a very weak positive correlation with respect to the notability label. This effect is expected because both of these feature sets do not generalize as well as above above-discussed feature sets. However, these count-based features function effectively for certain categories such as Film Actors, and Cricketers, and hence cannot be completely ignored while constructing entity embeddings. Wikicommons image count was also observed to have a low correlation for both classes, indicating the noise introduced due to retrieved results.

From the above observations, we can infer that the presence of an entity in the Wikipedia ecosystem (especially Wikidata), as well as entity-centric information in reliable web domains / analogous selected documents, corresponds the most to Wikipedia's notion of notability. Thus, our model has utilized the corresponding textual content and processed these salience encodings further to obtain the final embedding, which is a better approach than relying on handcrafted measures about entity salience.

4.5.3 Validation on WikiProject Popular Pages

Apart from the feature-level and component-level analysis, it is essential to gain an understanding of the system's performance on manually rated pages on Wikipedia, to note the difference in the confidence of Notability label prediction, and the effectiveness of our entity embeddings in differentiating among entities based on their related content. A deeper analysis is performed by testing the model's performance on popular Wikipedia pages as identified by WikiProject[7]. These page titles were included in the test set, and the model's confidence scores were obtained for each sample, for both the baselines and our system. These scores are aggregated based on the article's content quantity and quality, summarized by parameters - Assessment[8] and Importance[9], as defined by Wikipedia editors. Note that this analysis was performed based on the law of Historic recurrence[10], as the article created at a point in time depends only on reliable information about it on the web at that point. Hence, the time of

[7] https://en.wikipedia.org/wiki/Wikipedia:Lists_of_popular_pages_by_WikiProject

[8] https://en.wikipedia.org/wiki/Wikipedia:Content_assessment

[9] https://en.wikipedia.org/wiki/Wikipedia:Assessing_articles#Importance_ratings:_a_variety_of_definitions

[10] https://en.wikipedia.org/wiki/Historic_recurrence

Table 4.4: Comparative analysis based on prediction confidence, for WikiProject popular pages

Assessment	Generic class				Abstract class			
	Average views per year	**SGD score**	**Semant. embed. score**	**Our system score**	**Average views per year**	**SGD score**	**Semant. embed. score**	**Our system score**
FA: Professional, outstanding, and thorough	1624143	0.841	0.8212	**0.8953**	949943	0.6388	0.7307	**0.9094**
GA: Approaching (not necessarily equalling) a professional publication's quality	919580	0.7266	0.8132	**0.8731**	466386	0.7289	0.7336	**0.9524**
B: Readers are not left wanting, although content might not be complete enough	963599	0.7565	0.8194	**0.9034**	681334	0.4542	0.743	**0.8627**
C: Substantial but still missing important content	413750	0.7595	0.8049	**0.8982**	370163	0.5112	0.7432	**0.8632**
Start: Provides some meaningful content, but most readers will need more	174553	0.6761	0.7668	**0.8371**	128447	0.4032	0.7377	**0.8279**
Stub: Provides very little meaningful content	65599	0.6814	**0.7809**	0.7474	21694	0.2147	0.7453	**0.8249**

Importance	Generic class				Abstract class			
	Average views per year	**SGD score**	**Semant. embed. score**	**Our system score**	**Average views per year**	**SGD score**	**Semant. embed. score**	**Our system score**
Top: Subject is extremely important, even crucial, to its specific field	897455	0.8004	0.8379	**0.907**	693444	0.6553	0.7431	**0.9195**
High: Subject is extremely notable, but has not achieved international notability	505203	0.751	0.8132	**0.9071**	395328	0.4849	0.7403	**0.8627**
Mid: Subject is only notable within its particular field or subject	437280	0.7073	0.7788	**0.869**	253295	0.4086	0.7377	**0.8631**
Low: Subject is not particularly significant even within its field of study	394171	0.7145	0.7853	**0.8389**	206492	0.3899	0.7457	**0.7976**

article creation is immaterial, as the Notability test is the same in each case, which further justifies the generalizability of our system.

From table 4.4, it can be observed that the average confidence scores for each value of assessment/importance are relatively higher for our system, in comparison with the baselines. Further, the difference in prediction confidence across different levels of article assessment/importance is significant for our system, which assists in clearly distinguishing articles of high/low assessment/importance. These two aspects establish the superior performance of our model, as these statistics are generated for popular pages in WikiProject, justifying the higher confidence scores, as well as effective substantiation of the differences between articles' content via confidence scores. Note that the SGD baseline yields lower confidence scores while the semantic embed-

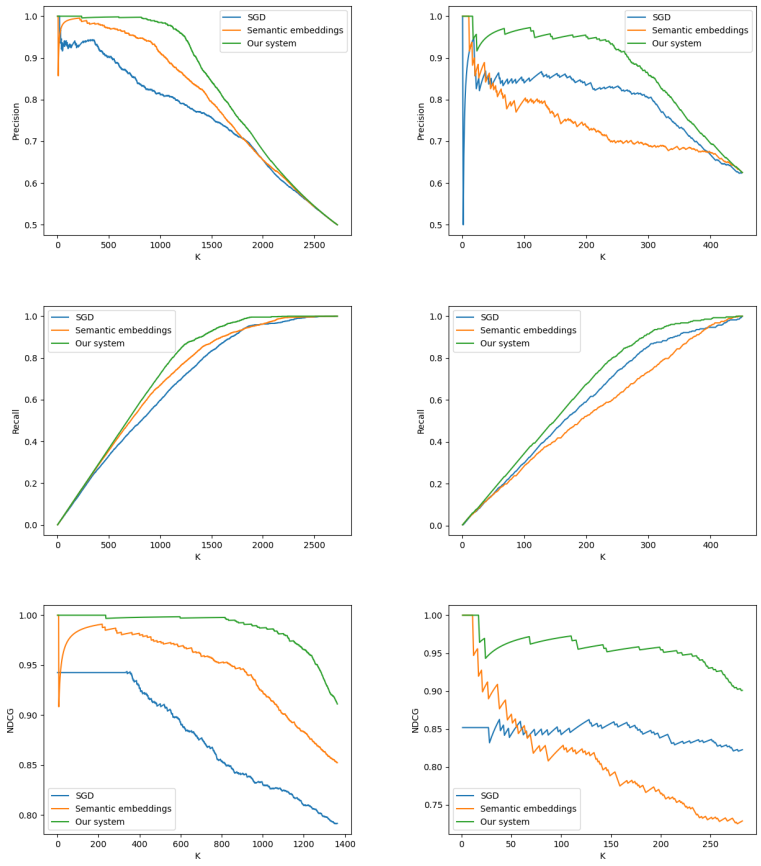

Figure 4.8: Precision, NDCG, Recall vs K for Generic (all plots on the left) and Abstract (all plots on the right) classes respectively

dings baseline doesn't sufficiently distinguish articles based on their assessment/importance, as reflected in the confidence scores.

Additionally, we utilize metrics such as precision, recall, and NDCG [28] (Normalized Discounted Cumulative Gain) are computed for top-K notable article recommendations generated and sorted by the confidence scores computed. The parameter K is varied (upper-bounded by test-sample count), and corresponding plots are generated for both baselines and our system (refer figure 4.8). Our system outperforms the baselines for all choices of K, for each of the metrics. This further establishes how our system of entity embedding construction functions effectively in the context of a retrieval model to identify and rank the most-notable and least-notable entities of a given entity list.

4.6 Limitations

While it can be observed that our system has proven effective in category-agnostic Notability determination based on our detailed analysis, it has certain limitations and biases. A key challenge is that it is sometimes non-trivial to label a category to belong to a given class (as discussed in the example of "Software"). This aspect is left for the user to decide and provide as input. The pipeline is complex and requires a significant computational effort for feature extraction. Further, the feature sets have a dynamic nature, such as Google - commercial search engine, news, and social media platforms, which affects the reproducibility as content about an entity is subject to change (this is expected behavior as interest in an entity changes over time). Bias is introduced by using pre-defined social media platforms, and online news websites, which function differently for different categories. Noise is also present in the system due to the heuristic mechanisms for entity-entry matching. Basic manual intervention is necessary to ensure noise in the system is minimized, in aspects such as verifying identification of web domains, data extraction, etc.

4.7 Summary

We have designed a web-centric entity-salience-based system to construct category-based entity embeddings for detecting the Notability of simple titles, which are further differentiated as belonging to the Generic/Abstract class, based on the nature of their title. We utilized feature extraction from various components in the web - reliable web sources for entity-related documents, Wikipedia ecosystem, query logs-based analysis, presence in social media, and relevance in online news websites. We designed the system architecture which utilizes BERT encodings and neural networks to construct entity embeddings for classification. The code is made publicly available. We validated our system with pre-defined baselines (section 4.2.1) and obtained a jump in performance accuracy by nearly 7% for the Generic class and 20% for the Abstract class. On thorough validation via correlations analysis, ablations, prediction confidence, the superior performance of our system is established. However, a few limitations exist in the system, such as noise, bias, and requiring slight manual intervention.

Chapter 5

Generating category-specific embeddings for Notability detection of Article titles having complex category-dependencies

5.1 Overview

In this chapter, we describe the system we designed to construct category-specific entity embeddings, to detect the Notability of complex titles, i.e., titles having non-trivial dependencies with a given category. Initially, we describe the feature extraction procedures, which contain how several relevant web-centric features have been collected for embedding generation of such complex titles. These web-based components, from which text-based salience features are extracted, include reliable web documents, the Wikipedia ecosystem, and query logs-based analysis. Necessary modifications are made to the system as against simple titles, to handle complex titles effectively.

We also discuss the baselines used, and our final classification architecture, which consists of a Graph neural network (GNN) that generates attention-enhanced category-specific embeddings for classification, with syntactic and semantic document graphs as inputs. We evaluated this system similar to the above system for simple titles and observed that it outperforms all baselines defined. The system's limitations, similar to simple titles, are also mentioned.

5.2 Feature extraction from Web-components to capture salience signals

Similar to the case of Article titles with Simple named entities, we design a supervised learning-based system architecture, to capture the salient features and dependencies to generate category-specific entity/title embeddings for classification. For each title in the dataset, signals about it on the web are captured by analyzing query logs, presence in the Wikipedia ecosystem,

and relevant documents on the web. Each of these components plays a role in providing insights about the titles coverage on the web, and its dependencies with the given category, as discussed in the case of Simple titles. We generate the required encodings for these components to create the category-specific entity/title embedding and pass them through a Feed Forward Neural network for classification. Note that despite the numeric features from web components functioning similarly to the case of Simple titles, there is an additional set of procedures used to generate specific encodings for classification, which work only for titles with complex categorical dependencies.

In defining the web-based feature set, we discarded reliable-web domain-based features that were present in the case of simple titles. This is because, for a given category, we cannot identify reliable web domains that consist of all complex titles corresponding to it. For instance, if we consider the category of "Birds", we cannot identify reliable web domains which comprise all titles that encompass all possible complex dependencies such as Islands, Wildlife sanctuaries, bird-related diseases, etc., which are all associated with the category of "Birds", but are not simple category instances. Hence, we rely on the approach of information distribution about the individual title on the web, as an alternative. We also discard social media and online news-based features that were used for simple titles, as it was observed that they did not boost the performance significantly, and were not particularly effective with the dataset of complex titles. We specifically ensured that the system is designed such that the non-trivial entity-category interconnections are effectively captured, for the accurate updation of the Wikipedia KG. Features extracted for the web components that are valid for complex titles and effective in the final embedding generation and classification are discussed in detail below. We make the code publicly available[1].

5.2.1 Information Distribution on the Web

We use the approach of "information distribution", used for simple titles (section 4.3.2), to extract relevant documents pertaining to any given article title, with respect to its category. A search query is formulated, with the title, category name, one prominent categorization of the title, and the keyword "information", to take into account all necessary phrases for extracting dependency-based relevant documents. An example query is "Wake Island" + "Islands of Oceania" + "Birds" + "information". This is performed for two types of queries - one with site extension "org" for reliable organization-based documents, and one without such restrictions to reflect generic content corresponding to the title on the web.

The retrievals for each set are examined and the top 8 results are considered, to ensure a sufficiently large initial document set. These result documents are further re-ordered, prioritizing documents with more title terms' mentions, as they would typically contain more title-related

[1] `https://anonymous.4open.science/r/Notability-Detection-Complex-Titles-1F01`

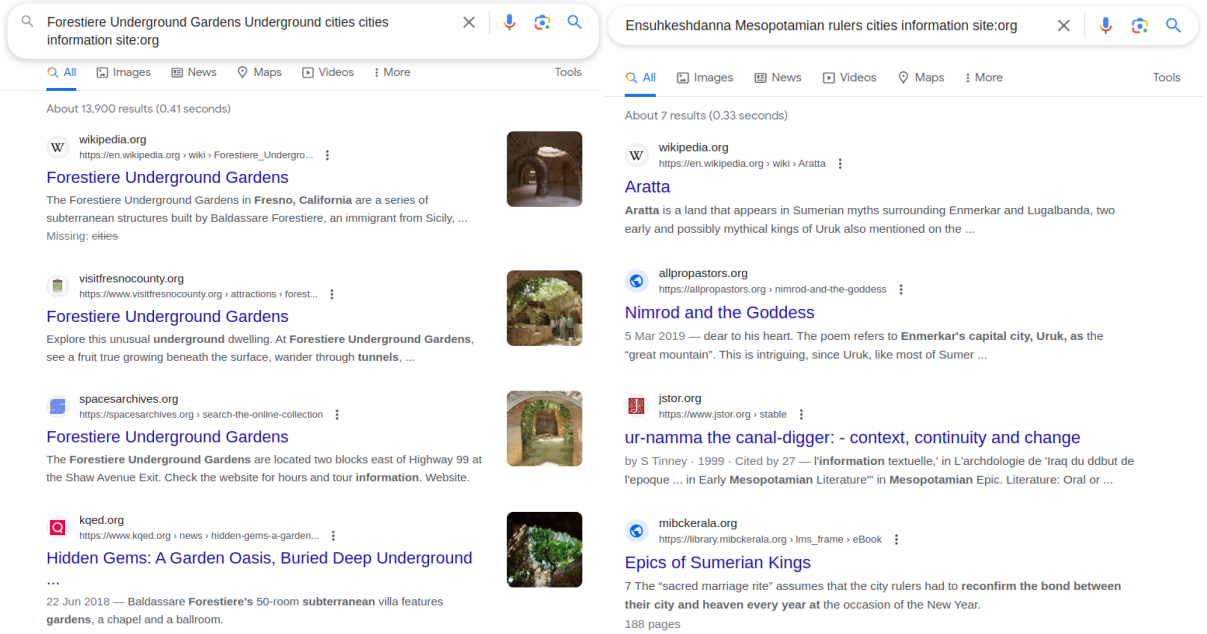

Figure 5.1: For a notable title (left), there is more relevant and title-centric information corresponding to it on the web in independent sources, in comparison to a non-notable title (right)

information. After re-ordering, the top 3 documents are chosen for further analysis, to increase the relevant information-density in the sample set. Numeric salience features E_p, E_f, E_i, and E_h (table 4.3.1) used by Yashaswi et al. [51] are extracted from above documents. Note that the values for parameters such as top-k results, documents considered, and the mechanism to choose documents are slightly varied in comparison with simple titles. These changes were made based on empirical observations, to improve the text content analyzed in further steps.

An example of the execution of this procedure is illustrated in figure 5.1, where we can observe the search results from reliable organizations (with site extension ".org") for two examples from the "Cities" category in the dataset. The queries are in the format: Title + prominent categorization + Wikipedia category + "information". For the notable complex title "Forestiere Underground Gardens", we observe more relevant and title-centric web documents, while mostly irrelevant documents are observed for the non-notable complex title "Ensuhkeshdanna". This establishes the importance of relevant web-coverage for a complex title.

Apart from direct web-coverage, other relevant key components of the web used in simple titles, which are applicable to complex titles, are also explored. These components and their features are discussed below.

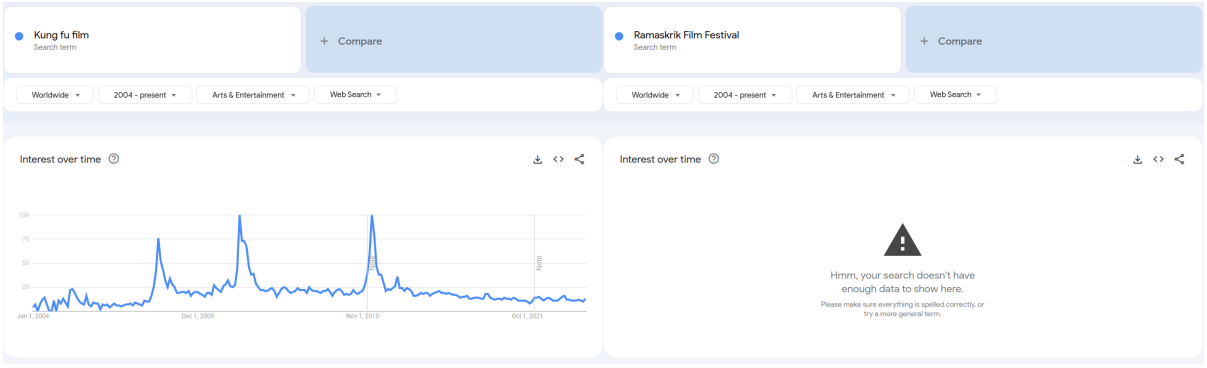

Figure 5.2: For a notable complex title (left), higher interest levels are recorded for a longer period of time in Google trends, in comparison to a non-notable complex title (right)

5.2.2 Wikipedia Ecosystem

We examine entries of titles in three primary components in the Wikipedia ecosystem - Wikidata, Wikicommons, and Wikipedia, just as in the case of simple titles (section 4.3.3). The procedures followed for extracting relevant features to encode the extent of coverage, are the same. However, there is a minor updation in the way search queries are formulated, for performing search on the 3 platforms. The search queries additionally contain one prominent categorization of the complex title, along with the title and category name. This additional information plays a key role in identifying more relevant search results that take into account the nature of the categorical dependency of the title.

5.2.3 Query Logs

Other than concrete information on web domains and Wikipedia ecosystem, general public's interest in a topic is captured by Query log features. These features of a title pertain to frequency analysis of search queries regarding the title, in a search engine (Google here). These are extracted in the same manner as done for simple titles (section 4.3.4). The difference in interest level scores for two film-related complex titles from the dataset is illustrated in figure 5.2, where the interest scores are higher for the notable complex title "Kung fu film", and there is negligible interest regarding the non-notable complex title "Ramaskrik Film Festival".

5.3 Additional Data Pre-processing using TextRank

Text data of web documents and relevant Wikipedia articles was extracted from URLs using the justext [52] tool. However, it was observed that additional text-processing was necessary to ensure content relevance. This was achieved by defining a specific text-cleaning procedure,

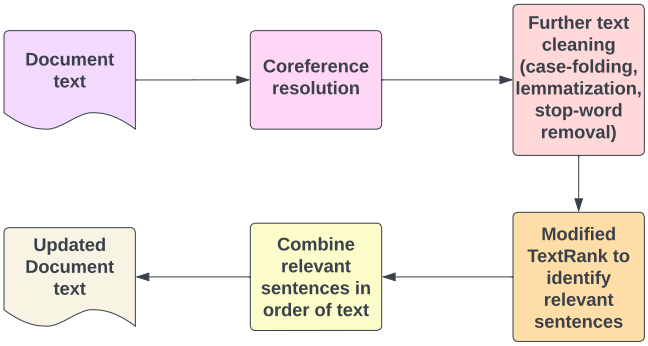

Figure 5.3: Additional pre-processing of documents for complex titles

which was augmented above the cleaning procedure used for simple titles, to improve the content in the documents to reflect more information related to the title and its dependencies with the category. We define a TextRank-based procedure to perform this task by cleaning up irrelevant sentences. This process is described as follows.

- Initially, for each document, coreference resolution is performed. Next, basic pre-processing such as case-folding, lemmatization, and stop-word removal is performed (as for simple titles' documents).
- Sentence Transformers [57] are used to obtain sentence embeddings for each sentence in the document, for accurately capturing semantic information for further steps.
- For each sentence, two types of semantic similarity scores are computed. One is a cosine-similarity score computed between the sentence's content and manually defined category keywords' (as discussed in section 3.5.1) embeddings, and another cosine-similarity score is between sentence content and title categorizations' embeddings. These two similarity scores indicate the sentence's correspondence with the title's meta-data.
- We define the mean of these two similarity scores for the i^{th} sentence as s_i. Consider cos_{ij} to be the cosine similarity between i^{th} and j^{th} sentence vectors.
- We construct the similarity matrix S for TextRank [42], where the element S_{ij} is given by equation 5.1. This matrix is defined in such a manner as to encapsulate sentence-level similarity as well the similarity between sentence and title meta-data.

$$S_{ij} = cos_{ij} \cdot (\frac{s_i + s_j}{2}) \cdot (1 - |s_i - s_j|) \tag{5.1}$$

- After constructing S, TextRank is applied for 500 iterations with a damping factor of 0.85. Sentences with low scores are eliminated, and highest-scoring sentences (until a word limit of 300 is reached) are retained in their order in the original text.

The flow of this pre-processing is visualized in figure 5.3. This method has significantly improved the quality and relevance of content in documents. This is because higher cosine similarity and higher sentence scores s_i imply high relevance to categorizations, which consequently yield higher similarity coefficients. Closer sentence scores also yield high coefficients, indicating that sentences have similar relevance with respect to the category keywords and title categorizations. Thus, the sentences finally remaining in the document have relatively less noise with respect to the complex title under consideration.

5.4 Baselines

We utilize three models as baselines to compare and validate the effectiveness of our system. The first two baseline models are similar baselines as used in the case of simple titles. We define another BERT-based baseline, which is similar to the final architecture used in the system for simple titles. These baselines are chosen based on their usage in similar downstream tasks, adherence to the definition of Notability, scalability, and generalizability, and to clearly establish how our designed system outperforms them for this specific task of complex titles.

5.4.1 Handcrafted entity-salience features: SGD

This baseline is similar to the one defined for simple titles (section 4.2.1). Based on the numerical feature data extracted, including count-based features from Google Trends, Wikipedia, and handcrafted salience metrics (table 4.3.1), binary classification is performed using Stochastic Gradient Descent classifier. Note that there is no special consideration for the nature of complex titles, to understand the impact of our defined system in effectively capturing these dependencies.

5.4.2 FastText-embedding similarity-based classification

This baseline is similar to the one defined for simple titles (section 4.2.2), which considers semantic context instead of heuristic features defined in the SGD baseline. For each document, FastText [30] word embeddings are obtained. Pairwise similarities of each document word embedding are computed against word embeddings of a short query defined for each sample, comprising the title, category keywords, and the title's categorizations. These similarity encodings, which capture the correspondence between title-dependency information and document text, are combined across documents and passed through a Multi-Layer Perceptron for performing binary classification. For this baseline, the only additional consideration from Simple titles is the title's categorizations, that provide basic additional information about the title.

5.4.3 BERT encodings for topic-salience

For each document text, a short query with title and meta-information (categorizations and category keywords) is prepended. BERT is used to generate word encodings for each such text, and the mean word encoding of each document is concatenated and passed through an MLP for classification.

This architecture is similar to that in the final system designed for simple titles. This system is used for comparison regarding how the performance can be improved, both theoretically and results-wise, for complex titles, at the cost of a slightly more complex architecture. This would help in better analyzing how these types of entity-category connections in the Knowledge graph are different from the case of the previous system.

5.5 Category-specific embeddings: Web-based count features + BERT + GNN

Based on the above set of features extracted from relevant web components, both count-based and text-based, we design an architecture to arrive at the Notability label for each complex title, by robustly generating category-specific entity embeddings. We use a GNN-based approach, for capturing complex dependencies between article titles and their categories and analyzing their coverage for notability prediction (figure 5.4). This is because of the effectiveness of graphs in such dependency-based tasks. Each document is represented as a graph to extract dependencies such as title terms among each other, title terms with category keywords, and title categorizations with keywords.

For each title t, the above-extracted numeric features are combined to form encoding n_t, and passed through a linear layer with sigmoid activation, to obtain encoding N_t (equation 5.2). This encoding consolidates the web-based heuristic features, summarizing the corresponding web-signals. Apart from such heuristic measures, we also rely on sufficient relevant text data for each title. As discussed previously, a total of 8 text documents are associated with each title. Here, 6 documents pertain to web information distribution (3 each for two types of queries - with the extension ".org" and without any restrictions), and 2 most relevant documents in the Wikipedia ecosystem.

We define an additional encoding which is essential to capture the key categorical dependencies of the complex title. For each of the documents obtained above, a key string s comprising the title's terms, category keywords, and title categorizations is prepended to the document. This key string is utilized in subsequent steps as it comprises all necessary terms whose dependencies are essential for the classification, to generate the category-specific entity/title embedding.

Each document is passed through BERT (Bidirectional Encoder Representations from Transformers), which is effective in such representation-based tasks, as it encodes text by attention

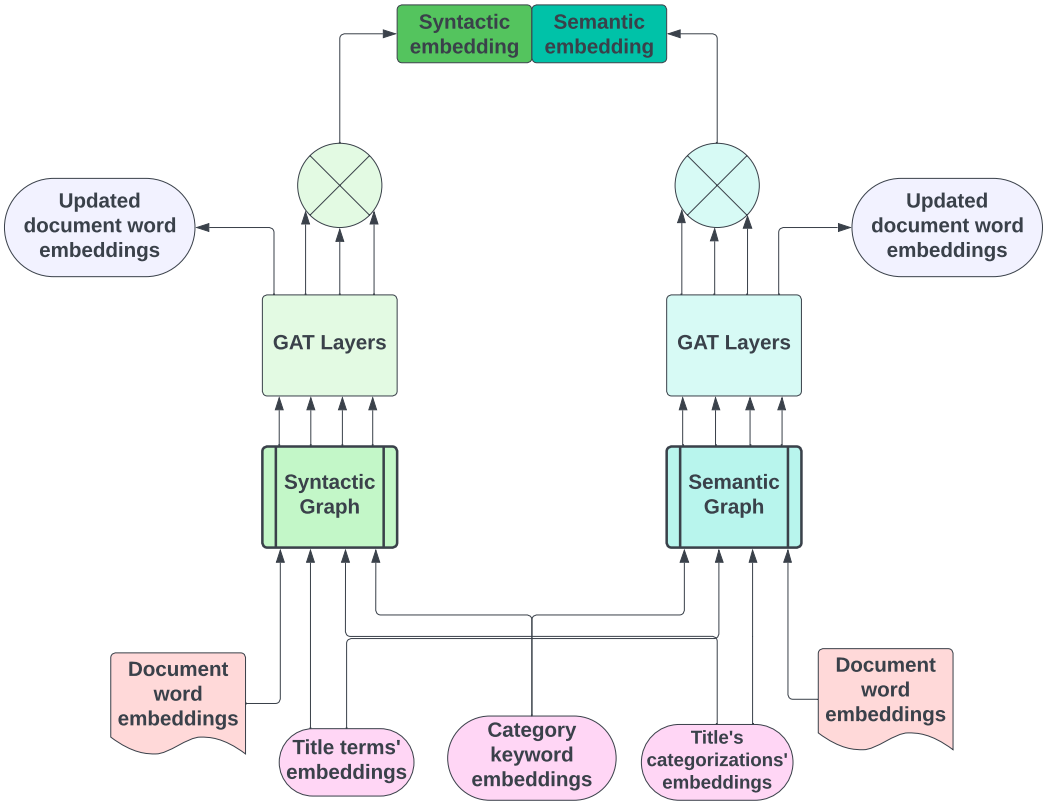

Figure 5.4: Document encoding generation through Graph-Attention mechanism for capturing non-trivial entity-category dependencies

mechanism. Consider the list of BERT word encodings obtained for the i^{th} document d_i, for a title t, to be W_{ti}.

We explore the graph-based representation for the above-collected data to model the required dependencies described in the key string s. For each document d_i of a given title t, we define two document graphs - syntactic graph G_{ti}^{syn} and semantic graph G_{ti}^{sem}, for capturing syntactic and semantic relationships among words in documents respectively.

For both types of document graphs, nodes are words of the document, and their BERT encodings W_{ti} are initial states. The edges in the syntactic graph G_{ti}^{syn} are constructed between nodes that are co-occurring in a window of size 5, capturing the syntactic dependencies based on the sentence structure. The edges in the semantic graph G_{ti}^{sem} are constructed based on semantically closest k-nearest neighboring nodes, for each node, using word encoding representations and cosine similarity. This value of k was chosen to be 10, based on empirical experimentation. This graph captures the semantic nature of the documents. Combining the features from both these graphs would provide a holistic view essential for embedding generation and classification, capturing the relevant associations between the title and its related meta-data.

After constructing these graphs for all relevant documents for a title, we employ the graph-attention mechanism to appropriately distribute weights to each document term based on its relationship with the complex title, both syntactically and semantically. Both graphs G_{ti}^{syn} and G_{ti}^{sem} are passed through two separate sets of Multi-Headed Graph Attention layers [77], where each Graph Attention (GAT) layer has 3 heads, and each set has 3 such GAT layers. Note that the GAT layer output obtained is after concatenating outputs from all its attention heads, which are used to encapsulate multiple forms of dependencies among the encodings. After passing through these layers, updated node encodings are obtained for each graph. These encodings capture complex dependencies defined that have been described above, as the attention weights flow through the graphs.

On obtaining the updated node encodings for both the syntactic and semantic documents graphs, we specifically analyze nodes corresponding to the key string s, as it comprises all terms whose dependencies are important for classification. Node encodings corresponding to s are mean-pooled separately for both G_{ti}^{syn} and G_{ti}^{sem} to highlight key features and reduce dimensionality, and we define the output obtained as g_{ti}^{syn} and g_{ti}^{sem} respectively. Both g_{ti}^{syn} and g_{ti}^{sem} across all documents d_{ti} are combined to obtain samples final Graph-Attention encodings g_{t}^{syn} and g_{t}^{sem} (refer equations 5.3, 5.4 and figure 5.4). In this manner, the required graph-based encodings are obtained, which completely capture all the required connections, and form a part of the final embedding for classification.

The classifier is further enabled to identify the differences in features across different categories. We define a categorical embedding e'_c for a category c, for differentiating content from diverse categories. It is obtained from the category's identifier id_c and embeddings matrix Emb_c (equation 5.5). Finally, all encodings obtained are concatenated to form the final category-specific entity/title embedding T' (equation 5.6). This embedding T' comprises appropriately encoded key web-based signals, which are useful to establish complex interconnections between the title and a given category, in the Wikipedia KG. This unified embedding T' is then passed through a Feed Forward Neural network comprising three linear layers, to obtain the final output o (equation 5.7). This output is obtained by combining all heuristic-based and text-based features, taking into account the nature of dependencies between the title and its category, and assigning appropriate weights via the feed-forward network for classification. Further specifications regarding the model definition and training are described below.

$$
\begin{aligned}
N_t &= \sigma(W_n^T n_t + b_n) && (5.2)\\
g_t^{syn} &= \Big|\Big|_i GAT(G_{ti}^{syn}) && (5.3)\\
g_t^{sem} &= \Big|\Big|_i GAT(G_{ti}^{sem}) && (5.4)\\
e'_c &= Emb_c(id_c) && (5.5)\\
T' &= [N_t, g_t^{syn}, g_t^{sem}, e'_c] && (5.6)\\
o &= \sigma(FFN(T')) && (5.7)
\end{aligned}
$$

5.5.1 Experimental Setup

The document word limit was set to 200, based on effectiveness after TextRank-based pre-processing. The word limit for the key string s was set to 25. We used Bert-base-uncased for generating initial node encodings. Each GAT layer had an input dimension of 768, and an output dimension of 256 (for each attention head). The model was trained for 2 epochs using Adam optimizer with a learning rate 2e-4, and batch size 2, due to fast learning of the model and computational constraints. Binary Cross-entropy was the loss function used.

The results of the baseline approaches, experiments, and our defined system are compared in detail for an overall analysis. This is explained in section 4.5.

5.6 Results and Discussion

The standard accuracy metrics used for binary classification such as Precision, Recall, F1 score, and Accuracy were computed for the baselines, ablation experiments, and the final system (Table 5.1). The test set distribution was ensured to have the same category-wise distribution as the dataset. Performance accuracy increased by nearly 5% compared to the BERT baseline (best-performing baseline). Despite the BERT baseline achieving a good accuracy score, our defined Graph-based approach is more theoretically sound when dealing with titles having complex categorical dependencies for generating embeddings, which is also evident from the boost in performance.

An ablation study, correlation analysis, and validation on existing Wikipedia pages were conducted to analyze our system's effectiveness. Note that the evaluation methods followed are similar to the evaluation of the system for simple titles, as the high-level problem statement is the same in both cases.

Table 5.1: Metrics for baselines, experiments, ablations and final system

System	ACC	PR	REC	F1
SGD	0.690	0.767	0.696	0.670
FastText-based similarity	0.822	0.826	0.820	0.821
BERT	0.914	0.915	0.915	0.914
- Semantic Graph	0.914	0.92	0.916	0.914
- Wikipedia Documents	0.937	0.937	0.937	0.937
- Web Documents	0.95	0.951	0.951	0.95
- Count features	0.951	0.953	0.952	0.951
- Categorical embedding	0.951	0.953	0.951	0.951
- Syntactic Graph	0.952	0.952	0.953	0.952
Final System	**0.959**	**0.959**	**0.960**	**0.959**

5.6.1 Ablation Study

Ablation experiments were conducted by removing components from the system (only one at a time), to analyze individual component contributions (Table 5.1) and their applicability. We can observe that semantic graph encodings g_t^{sem} were more essential in the final embedding than syntactic encodings g_t^{syn}, establishing how deeper semantic connections are captured by our model. Graph encodings of relevant documents in the Wikipedia ecosystem played a more significant role in classification than relevant web documents, justifying their presence in the final embeddings. Other components such as categorical embeddings, handcrafted features, etc. had relatively weaker contributions, as can be observed from the performance after their removal. To understand feature contributions at a more granular level, correlation analysis is performed.

5.6.2 Correlation Analysis

Correlation analysis is performed to validate the applicability of defined web-based features, similar to the case of simple titles, for understanding feature-level contribution in classification. The Pearson correlation coefficient was computed for each of the numerical features. The correlation scores for attributes with relatively higher values (at least 0.1) are enumerated in table 5.2, as they played a relatively significant role in embedding generation and classification.

It can be observed that the Wikipedia ecosystem-based features have a superior correlation score for both classes. The handcrafted entity-salience metrics of relevant Wikipedia documents and selected "information distribution" based documents also achieve relatively higher correlation scores. It can also be noted that documents obtained from the generic search have

Table 5.2: Pearson correlation coefficients for numeric features

Handcrafted metrics correlation scores							
Attribute	Score	Attribute	Score	Attribute	Score	Attribute	Score
Wikipedia-Eh-0	0.341	Wiki-1-Entity-name-Ep	0.34	Wiki-1-Category-Ef	0.329	Wikipedia-Ei-0	0.326
Wiki-1-Entity-name-Ei	0.325	Wikipedia-Ef-0	0.318	Wiki-1-Entity-name-Ef	0.317	Wiki-1-Category-Ep	0.285
Gen-2-Category-Ef	0.273	Gen-1-Category-Ef	0.272	Wiki-1-Category-Ei	0.259	Gen-3-Category-Ef	0.251
trends-max	0.246	Wikipedia-Ef-1	0.244	Wiki-2-Category-Ef	0.242	Gen-4-Category-Ef	0.221
Wiki-2-Entity-name-Ef	0.216	Org-1-Category-Ef	0.215	Gen-5-Category-Ef	0.208	Wikipedia-Ef-2	0.206
Wiki-1-Entity-name-Eh	0.202	Wiki-2-Category-Ei	0.201	Wiki-2-Entity-name-Ep	0.201	Wiki-3-Category-Ef	0.199
Wiki-2-Category-Ep	0.199	Wiki-2-Entity-name-Ei	0.198	Wiki-1-Category-Eh	0.197	Org-2-Category-Ef	0.185
Wiki-3-Entity-name-Ef	0.184	Wikipedia-Ei-1	0.181	Wikipedia-Eh-1	0.181	Wiki-3-Category-Ei	0.173
Wiki-3-Category-Ep	0.17	Gen-6-Category-Ef	0.169	Wiki-3-Entity-name-Ep	0.167	Org-3-Category-Ef	0.16
Wiki-3-Entity-name-Ei	0.159	trends-mean	0.157	Wikipedia-Ei-2	0.147	Gen-2-Category-Ep	0.146
wikidata-docs-count	0.145	trends-3/4	0.145	Gen-4-Entity-name-Ef	0.143	Gen-1-Category-Ep	0.143
Gen-3-Entity-name-Ef	0.137	Org-1-Category-Ep	0.136	Wikipedia-Eh-2	0.132	Org-4-Category-Ef	0.132
Gen-4-Category-Ep	0.131	Wiki-2-Entity-name-Eh	0.13	Wiki-2-Category-Eh	0.129	Gen-8-Category-Ef	0.128
Gen-7-Category-Ef	0.128	Gen-3-Category-Ep	0.128	trends-med	0.127	Gen-5-Entity-name-Ef	0.126
Gen-5-Category-Ep	0.119	Gen-2-Entity-name-Ef	0.118	Wiki-3-Entity-name-Eh	0.116	Org-5-Category-Ef	0.113
Org-2-Category-Ep	0.112	trends-1/4	0.11	Wiki-3-Category-Eh	0.107	Gen-1-Entity-name-Ef	0.102
Gen-6-Entity-name-Ef	0.1	Org-3-Category-Ep	0.096	Gen-6-Category-Ep	0.094	Gen-4-Entity-name-Ep	0.092

a higher correlation in comparison with documents obtained from only reliable organizations (site extension "org").

We can also observe how two types of entity-salience features are calculated for each document - mentions of a title (with the term "-Entity-name-" in the middle of attribute names of table 5.2), and mentions of its corresponding categorizations extracted (with the term "-Category-" in the middle of attribute names). The mentions of a title are similar to the case of an entity's mentions in documents of simple titles. The mentions of corresponding categorizations indicate the effectiveness of the additional information for each sample, which assists in understanding the complex title. The significant correlation scores obtained in both cases establish the superior performance of our model, as it relies on these features which are captured in a more refined manner via graph attention mechanism.

Among other web-component-based features, Query logs have a relatively lower correlation in comparison with the above-document-based entity salience features. However, as in the case of simple titles, we observe again that the attribute "trends-max" has a reasonably significant correlation, indicating that an entity is still notable if it was notable at a point in time (which is mentioned in the definition of Notability for Wikipedia).

From the above observations, we can infer that the presence of a title in the Wikipedia ecosystem, as well as title-centric information in selected web documents, corresponds the most to Wikipedia's notion of notability. Thus, our model has utilized the corresponding textual content and processed these salience encodings further, utilizing the graph-attention mechanism,

by taking both syntactic and semantic information into account to obtain the final embedding. This indicates the effectiveness of our designed system in capturing complex dependencies of titles with categories.

5.6.3 Validation on existing Wikipedia pages

In addition to the feature-level and component-level analysis, we perform an in-depth analysis by computing the system's confidence in prediction for pre-existing Wikipedia pages in the test-set. We exclusively analyze positive samples, to quantify performance on existing Wikipedia content. As in the case of simple titles, we do not specifically focus only on popular pages, as the overlap of such Wikiproject popular pages was lesser with complex titles in our dataset. Similar to simple titles, this analysis is performed based on the law of Historic recurrence[2], as the article created at a point in time depends only on reliable information about it on the web at that point. Model confidence scores were obtained for each sample, for the three defined baselines and our system. These scores are aggregated based on the article's content quantity and quality, summarized by parameters 'Assessment' and 'Importance', as in the case of simple titles.

From table 5.3, it can be observed that the average confidence scores for each value of assessment/importance are relatively higher for our system than for baselines. Further, the difference in prediction confidence across different levels of article assessment/importance is significant for our system, which assists in clearly distinguishing articles of high/low assessment/importance. As in the case of simple titles, these two aspects establish the superior performance of our model, as these statistics are generated for positive samples, justifying higher confidence scores, and substantiating differences in article content. Note that articles were grouped in both cases (FA-C and Top-Mid), because of a lack of sufficient samples in the cases of positive extremes (such as a highly notable article or a very informative and thorough article).

We have also computed additional metrics such as precision, recall, and NDCG (Normalized Discounted Cumulative Gain) to establish how our system functions effectively in the context of a retrieval model to identify and rank Wikipedia article titles based on quantity and quality of content. These metrics are computed for the top-K notable article recommendations generated, which are further sorted by the confidence scores computed. The parameter K is varied and plots are generated for the baselines and our system (refer figure 5.5). Our system (red line in plots) outperforms the baselines for all choices of K, for each metric, indicating the effective function of our system as a recommender of highly notable complex titles, based on the category-specific embeddings generated.

[2] `https://en.wikipedia.org/wiki/Historic_recurrence`

Table 5.3: Comparative analysis based on prediction confidence, for positive samples

Assessment	Page Views		Confidence scores			
	Average views per year	**Views last year**	**SGD score**	**FastText. sim. score**	**BERT score**	**System score**
FA-C: Ranging from professional, outstanding, and thorough articles, to substantial articles requiring more information	220859	259664	0.544	0.784	0.94	**0.988**
Start: Provides some meaningful content, but most readers will need more	40566	38292	0.504	0.758	0.919	**0.958**
Stub: Provides very little meaningful content	6080	6407	0.404	0.747	0.826	**0.896**
Importance	**Page Views**		**Confidence scores**			
	Average views per year	**Views last year**	**SGD score**	**FastText. sim. score**	**BERT score**	**System score**
Top-Mid: Ranging from subject being extremely important and internationally notable, to subject being notable only in its field	140031	159278	0.56	0.764	0.918	**0.964**
Low: Subject is not particularly significant even within its field of study	56196	60975	0.47	0.753	0.899	**0.948**

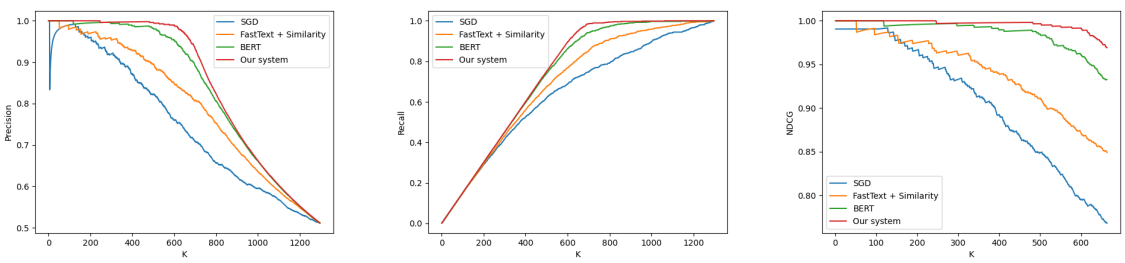

Figure 5.5: Precision, NDCG, Recall vs K for complex titles' system

5.7 Limitations

Despite the superior performance of our system as observed in the in-depth analysis performed above, our approach has certain limitations, which are mostly similar to those discussed in the system for simple titles (section 4.6). The pipeline requires high computational effort, and the feature sets have a dynamic nature, such as Google - a commercial search engine, which affects reproducibility as content about a title is subject to change. However, this is expected behavior as interest in a topic changes over time. Manual intervention is necessary for aspects such as annotation of title categorizations, verifying the correctness of unsupervised title classification, etc.

5.8 Summary

We have designed a graph-attention-based system to construct category-specific entity/title embeddings for detecting the Notability of complex titles, i.e., titles that have non-trivial and complex dependencies with a given category. We utilized feature extraction from various components in the web - reliable web documents via the "information distribution" procedure, Wikipedia ecosystem, and query logs-based analysis. We designed the classification architecture which constructs syntactic and semantic document graphs and uses a graph-attention mechanism to extract keyword embeddings that capture the desired dependencies in generated embeddings. The code is made publicly available. We validated our system with pre-defined baselines (section 4.2.1) and obtained a jump in performance accuracy by nearly 5% for the best-performing BERT-based baseline. On thorough validation via correlations analysis, ablations, and prediction confidence, the superior performance of our system is established. However, a few limitations exist in the system, such as noise, bias, and requiring slight manual intervention, as in the case of the system for simple titles.

Chapter 6

Conclusion and future work

In this thesis, we have designed systems that comprise novel frameworks for extracting category-specific entity embeddings that encode key signals about an entity/topic, and its association with a desired category, to establish connections for populating a Knowledge graph. We empirically proved the effectiveness of our system by utilizing it for the important task of detecting the notability of entities for the Wikipedia KG. This problem statement has broader implications as it could be extended to any Knowledge-graph downstream task such as entity identification, entity linking, text generation, classification, etc. that require capturing key dependencies between entities and categories. In this context, we have targeted the problem of Notability determination of Wikipedia pages in a category-agnostic manner. It is essential to tackle this problem as it directly controls the efficient functioning of the Wikipedia platform by omitting Wikipedia articles for topics that do not warrant them, i.e., which do not satisfy the definition of Notability as defined by Wikipedia editors. In general, this process requires huge manual effort to annotate each such topic as notable/non-notable. We attempt to design approaches that focus on automating this process of Notability detection of a Wikipedia page, in a category-agnostic manner, which is otherwise a cumbersome task. In doing so, we also take into account the nature and organization of content in Wikipedia, which is non-trivial because of the complex structure of the platform. The approach utilized could be extended to similar tasks related to entity embeddings and Knowledge graphs.

We designed systems to detect the Notability of titles for Wikipedia in an automated manner, irrespective of the category of the title, based on the understanding of the hierarchy of articles in Wikipedia, across various types of categories. Firstly, we organize titles in Wikipedia into different partitions based on the nature of the content of the Wikipedia page, and its association with the Wikipedia category it is a part of. We primarily focus on two categories - Simple and Complex, based on the dependency of the title with its defined category. For more robust functioning, Simple titles are further divided into Generic class and Abstract class, on the basis of the nature of the titles. We construct datasets for each partition of titles, comprising a diverse

set of categories. We also designed an efficient mechanism to differentiate between simple and complex titles in Wikipedia in a category-agnostic manner.

We have designed the Notability detection systems for both cases of Simple and Complex titles, in such a way as to capture key signals on the web corresponding to the notability of a title and generate entity embeddings incorporating these signals accordingly. Such web components include the Wikipedia ecosystem, relevant web documents, query logs, social media, and presence in news web domains. The entity embeddings we contributed were found to play a very effective role in determining the notability of a title, in both systems. These two systems as a whole were observed to exhaustively capture entity-category dependencies, which would assist in accurately populating the Wikipedia Knowledge Graphs (and could be further extended to similar systems). Both of our designed systems tackle the issue of non-generalizability which was in previous works. Further, in the case of simple titles, we also handle the challenges mentioned in previous work regarding abstract concepts, which is addressed by defining a new set of information distribution features; which has also been used in the system for complex titles. We make our datasets and code publicly available.

The proposed approaches are generalizable to any new categories, based on how they accommodated the variability and diversity of features extracted from different categories in the constructed datasets. This extendability eliminates the high manual effort spent on content creation for non-notable entities (irrespective of their category). Our approach is an automated alternative for notability labeling, rather than relying on manual verification of coverage of a title on the web. Further, the approaches we designed can be used for establishing connections between any given entity-category pair by analyzing their dependencies from web components effectively. This makes our system extendable to similar tasks related to Knowledge graphs. We describe a chapter-wise flow of the thesis below.

we highlight In this context, we motivate the need to perform the related downstream task of automating Notability detection for the efficient functioning of Wikipedia. We also introduce the key contributions of our work here.

We start this thesis by highlighting the need for populating KGs by encoding entity-category connections. In this context, we motivate the need to solve the issue of the increasing content creation rate of Wikipedia (English) in **Chapter 1**, and define the necessity for a Notability test, defined by Wikipedia editors, and the issues in automating it. This chapter also introduces the classification of titles, dataset construction, and the various systems proposed in this work which are discussed in the next chapters of this thesis.

In **Chapter 2**, we discuss some of the related works done in the past, in the field of Knowledge-graph population and notability detection. We also explore similar directions in-depth, such as entity and event salience-based works, summarization-based works, semantic-modeling-based works, Graph-based works for dependency extraction (complex titles), web popularity, etc.

In **Chapter 3**, we define various types of article titles that exist in Wikipedia and construct datasets accordingly for each such type and sub-type. We discuss the difference in Simple/Complex titles, and how Simple titles are further divided into Generic/Abstract classes. We also design an automated, category-agnostic, unsupervised approach for title classification as simple/complex.

Chapter 4 consists of a detailed description of the designed web-centric entity-salience-based system to generate attention-enhanced entity embeddings capturing categorical connections, and to detect the Notability of simple titles, which are further differentiated as belonging to the Generic/Abstract class. Web-based feature extraction is performed, and BERT encodings of text are used in conjunction with neural networks to generate entity embeddings for classification. We also discuss their performance, analysis and limitations.

In **Chapter 5**, we discuss the design of a graph-attention-based system to generate category-specific entity embeddings to detect the Notability of complex titles, i.e., titles that have non-trivial and complex dependencies with their corresponding categories. Apart from feature extraction from web-based components, we utilize document graph representations and the graph-attention mechanism to extract key dependencies and formulate embeddings to perform binary classification. Performance, analysis, and limitations are discussed in detail.

Despite our best efforts in automating the process of Notability detection for Wikipedia pages by generating effective embeddings, there is always scope for improvement in designing such systems because of the vast and dynamic nature of the Wikipedia platform and content on the web. The work in the field of Notability is in the nascent stages, but we believe that further efforts in this direction would help achieve more significant milestones in the context of the Wikipedia platform, because of the huge potential impact in the reduction of manual labor necessary for robust and efficient functioning of the platform. We discuss possible future directions related to the field of automating Notability detection and populating Knowledge graphs in general, in the below section.

Future Work

Multiple possible future works are possible in the work presented in this thesis. Below-mentioned points discuss these aspects in detail.

1. The approaches we have designed involve time-consuming steps related to data collection, because of how the features are defined. A further improvement in the scalability of the system could be an interesting direction to pursue. This could be explored either by performing additional feature-engineering experiments to replace time-consuming components, or by exploring alternative views in tackling the problem rather than a purely feature-based approach.

2. Augmenting a summarization-based understanding of the text to obtain the title's salience in documents, could be another extension in the pipeline of constructing entity embeddings for Notability detection. This could help in reducing the magnitude of text encoded per sample, thereby improving efficiency while retaining performance.

3. Incorporating more components of the notability guidelines, such as subject-specificity, would make the Notability detection more reliable and generalizable to any possible category. These aspects are harder to define for a category-agnostic system, but further research in these fields could help identify potential patterns in incorporating these aspects into the system.

4. Additionally, the system can also be extended to a multi-lingual setting, as the current focus is on content in English. This would assist in constructing entity embeddings with significant information in their corresponding regional language, even if there is not enough content on the web in English.

Overall, there is scope in some areas where this work can be extended to generate more effective entity embeddings, which could be utilized for related downstream tasks. Designing a complete notability system taking all potential factors into account, but being simultaneously scalable and robust, would assist in the long-term moderation of content uploaded to Wikipedia, irrespective of the type of its category, language, and other such criteria. Further, this would help in capturing the essence of more types of entities/topics, which would assist in populating Knowledge Graphs more effectively.

Related publications

- **Gokul Thota**, and Vasudeva Varma. **A Category-agnostic Graph Attention-based approach for determining Notability of complex article titles for Wikipedia**. In Companion Proceedings of the ACM Web Conference 2024 (WWW '24 Companion). Association for Computing Machinery, New York, NY, USA.
- **Gokul Thota**, and Vasudeva Varma. **Generating entity embeddings for populating Wikipedia Knowledge Graph by Notability detection**. Under review as a full paper at the 29th International Conference on Natural Language & Information Systems, 2024.
- **Gokul Thota**, Rahul Khandelwal, and Vasudeva Varma. **A Web-centric entity-salience based system for determining Notability of entities for Wikipedia**. In Proceeding of the Wiki Workshop ′23.

Bibliography

[1] S. Abbar, C. Castillo, and A. Sanfilippo. To post or not to post: Using online trends to predict popularity of offline content. In *Proceedings of the 29th on Hypertext and Social Media*, HT '18, page 215219, New York, NY, USA, 2018. Association for Computing Machinery.

[2] K. Akyol and B. Şen. Modeling and predicting of news popularity in social media sources. *Computers, Materials & Continua*, 61(1), 2019.

[3] N. Appiktala, S. Huang, B. Sankar, S. Tripathi, and E. Goldman. Identifying salient entities of news articles using binary salient classifier. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 1541–1549, 2021.

[4] R. Bandari, S. Asur, and B. Huberman. The pulse of news in social media: Forecasting popularity. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 6, pages 26–33, 2012.

[5] T. Caliski and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27, 1974.

[6] M. Carbonell, P. Riba, M. Villegas, A. Fornés, and J. Lladós. Named entity recognition and relation extraction with graph neural networks in semi structured documents. *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 9622–9627, 2021.

[7] H. A. Carneiro and E. Mylonakis. Google Trends: A Web-Based Tool for Real-Time Surveillance of Disease Outbreaks. *Clinical Infectious Diseases*, 49(10):1557–1564, 11 2009.

[8] J.-Y. Chang. A study on research trends of graph-based text representations for text mining. *The Journal of the Institute of Webcasting, Internet and Telecommunication*, 13, 10 2013.

[9] S. Chatterjee and L. Dietz. Bert-er: Query-specific bert entity representations for entity ranking. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 14661477, New York, NY, USA, 2022. Association for Computing Machinery.

[10] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979.

[11] H. De Meulemeester and B. De Moor. Unsupervised embeddings for categorical variables. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[13] B. K. Y.-R. L. Drew B. Margolin, Sasha Goodman and D. Lazer. Wiki-worthy: collective judgment of candidate notability. *Information, Communication & Society*, 19(8):1029–1045, 2016.

[14] J. Dunietz and D. Gillick. A new entity salience task with millions of training examples. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 205–209, Gothenburg, Sweden, Apr. 2014. Association for Computational Linguistics.

[15] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457479, dec 2004.

[16] M. Färber, A. Rettinger, and B. El Asmar. On emerging entity detection. In E. Blomqvist, P. Ciancarini, F. Poggi, and F. Vitali, editors, *Knowledge Engineering and Knowledge Management*, pages 223–238, Cham, 2016. Springer International Publishing.

[17] D. Fernàndez-Cañellas, J. Espadaler, D. Rodriguez, B. Garolera, G. Canet, A. Colom, J. M. Rimmek, X. Giro-i Nieto, E. Bou, and J. C. Riveiro. Vlx-stories: Building an online event knowledge base with emerging entity detection. In C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I. Cruz, A. Hogan, J. Song, M. Lefrançois, and F. Gandon, editors, *The Semantic Web – ISWC 2019*, pages 382–399, Cham, 2019. Springer International Publishing.

[18] D. Fernàndez-Cañellas, J. Marco Rimmek, J. Espadaler, B. Garolera, A. Barja, M. Codina, M. Sastre, X. Giro-i Nieto, J. C. Riveiro, and E. Bou-Balust. Enhancing online knowledge graph population with semantic knowledge. In J. Z. Pan, V. Tamma, C. d'Amato, K. Janowicz, B. Fu, A. Polleres, O. Seneviratne, and L. Kagal, editors, *The Semantic Web – ISWC 2020*, pages 183–200, Cham, 2020. Springer International Publishing.

[19] M. Gamon, T. Yano, X. Song, J. Apacible, and P. Pantel. Identifying salient entities in web pages. In *Proceedings of the 22nd ACM International Conference on Information amp; Knowledge Management*, CIKM '13, page 23752380, New York, NY, USA, 2013. Association for Computing Machinery.

[20] N. Golbandi, L. Katzir, Y. Koren, and R. Lempel. Expediting search trend detection via prediction of query counts. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, page 295304, New York, NY, USA, 2013. Association for Computing Machinery.

[21] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks, 2016.

[22] D. Gunawan, C. A. Sembiring, and M. A. Budiman. The implementation of cosine similarity to calculate text relevance between two documents. *Journal of Physics: Conference Series*, 978(1):012120, mar 2018.

[23] C. Guo and F. Berkhahn. Entity embeddings of categorical variables, 2016.

[24] D. Guthrie, B. Allison, W. Liu, L. Guthrie, and Y. Wilks. A closer look at skip-gram modelling. *Proc. of the Fifth International Conference on Language Resources and Evaluation*, 01 2006.

[25] L. Huang, D. Ma, S. Li, X. Zhang, and H. WANG. Text level graph neural network for text classification, 2019.

[26] K. Jacobs and A. P. de Vries. Phrase extraction models used for entity salience detection. 2019.

[27] A. M. Jarman. Hierarchical cluster analysis: Comparison of single linkage, complete linkage, average linkage and centroid linkage method. *Georgia Southern University*, 29, 2020.

[28] K. Järvelin and J. Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, page 4148, New York, NY, USA, 2000. Association for Computing Machinery.

[29] W. Jin and R. K. Srihari. Graph-based text representation and knowledge discovery. In *Proceedings of the 2007 ACM Symposium on Applied Computing*, SAC '07, page 807811, New York, NY, USA, 2007. Association for Computing Machinery.

[30] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. In M. Lapata, P. Blunsom, and A. Koller, editors, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain, Apr. 2017. Association for Computational Linguistics.

[31] A. Karami and A. Elkouri. Political popularity analysis in social media. In N. G. Taylor, C. Christian-Lamb, M. H. Martin, and B. Nardi, editors, *Information in Contemporary Society*, pages 456–465, Cham, 2019. Springer International Publishing.

[32] N. Ketkar. *Stochastic Gradient Descent*, pages 113–132. Apress, Berkeley, CA, 2017.

[33] C. Kohlschütter, P. Fankhauser, and W. Nejdl. Boilerplate detection using shallow text features. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, page 441450, New York, NY, USA, 2010. Association for Computing Machinery.

[34] V. Kumar, J. K. Chhabra, and D. Kumar. Performance evaluation of distance metrics in the clustering algorithms. *INFOCOMP Journal of Computer Science*, 13(1):3852, Sep. 2014.

[35] S. T. K. Lam and J. Riedl. Is wikipedia growing a longer tail? In *Proceedings of the 2009 ACM International Conference on Supporting Group Work*, GROUP '09, page 105114, New York, NY, USA, 2009. Association for Computing Machinery.

[36] M. E. Lemieux, R. Zhang, and F. Tripodi. too soon to count? how gender and race cloud notability considerations on wikipedia. *Big Data & Society*, 10(1):20539517231165490, 2023.

[37] P. Li, Z. Wang, W. Lam, Z. Ren, and L. Bing. Salience estimation via variational auto-encoders for multi-document summarization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), Feb. 2017.

[38] H. Lin, Y. Jia, Y. Wang, X. Jin, X. Li, and X. Cheng. Populating knowledge base with collective entity mentions: A graph-based approach. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, pages 604–611, 2014.

[39] Z. Liu, C. Xiong, T. Mitamura, and E. Hovy. Automatic event salience identification. In E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1226–1236, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics.

[40] J. Lu and J. D. Choi. Evaluation of unsupervised entity and event salience estimation, 2021.

[41] F. Martini. Notable enough? the questioning of womens biographies on wikipedia. *Feminist Media Studies*, 0(0):1–17, 2023.

[42] R. Mihalcea and P. Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411, 2004.

[43] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space, 2013.

[44] A. A. Mohamed and S. Rajasekaran. Improving query-based summarization using document graphs. In *2006 IEEE International Symposium on Signal Processing and Information Technology*, pages 408–410, 2006.

[45] N. Moniz and L. Torgo. A review on web content popularity prediction: Issues and open challenges. *Online Social Networks and Media*, 12:1–20, 2019.

[46] F. Nielsen. *Hierarchical Clustering*, pages 195–211. Springer International Publishing, Cham, 2016.

[47] S. Niwattanakul, J. Singthongchai, E. Naenudorn, and S. Wanapu. Using of jaccard coefficient for keywords similarity. 03 2013.

[48] E. Palumbo, D. Monti, G. Rizzo, R. Troncy, and E. Baralis. entity2rec: Property-specific knowledge graph embeddings for item recommendation. *Expert Systems with Applications*, 151:113235, 2020.

[49] L. Pang, Y. Lan, J. Guo, J. Xu, J. Xu, and X. Cheng. Deeprank: A new deep architecture for relevance ranking in information retrieval. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, page 257266, New York, NY, USA, 2017. Association for Computing Machinery.

[50] S. G. K. Patro and K. K. Sahu. Normalization: A preprocessing stage, 2015.

[51] Y. Pochampally and K. Karlapalem. Notability determination for wikipedia. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, page 16411646, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee.

[52] J. Pomikálek. jusText, 2011. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

[53] Y. Qi, J. Zhang, W. Xu, and J. Guo. Salient context-based semantic matching for information retrieval. *EURASIP Journal on Advances in Signal Processing*, 2020, 07 2020.

[54] P. Radhakrishnan, G. Jawahar, M. Gupta, and V. Varma. Sneit: Salient named entity identification in tweets. *Computación y Sistemas*, 21, 2017.

[55] N. Rahman and B. Borah. A survey on existing extractive techniques for query-based text summarization. In *2015 International Symposium on Advanced Computing and Communication (ISACC)*, pages 98–102, 2015.

[56] N. Rahman and B. Borah. Improvement of query-based text summarization using word sense disambiguation. *Complex & Intelligent Systems*, 6:75–85, 2019.

[57] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.

[58] M. R. Saeed, C. Chelmis, and V. K. Prasanna. Not all embeddings are created equal: Extracting entity-specific substructures for rdf graph embedding, 2018.

[59] M. Saini, D. Sharma, and P. K. Gupta. Enhancing information retrieval efficiency using semantic-based-combined-similarity-measure. In *2011 International Conference on Image Information Processing*, pages 1–4, 2011.

[60] P. Saleiro and C. Soares. Learning from the news: Predicting entity popularity on twitter. In H. Boström, A. Knobbe, C. Soares, and P. Papapetrou, editors, *Advances in Intelligent Data Analysis XV*, pages 171–182, Cham, 2016. Springer International Publishing.

[61] C. Sammut and G. I. Webb, editors. *TF–IDF*, pages 986–987. Springer US, Boston, MA, 2010.

[62] J. Schneider, A. Passant, and S. Decker. Deletion discussions in wikipedia: Decision factors and outcomes. In *Proceedings of the Eighth Annual International Symposium on Wikis and Open Collaboration*, WikiSym '12, New York, NY, USA, 2012. Association for Computing Machinery.

[63] H. Sebei, M. A. Hadj Taieb, and M. Ben Aouicha. Popularity metrics normalization for social media entities. pages 525–535, 01 2018.

[64] K. R. Shahapure and C. Nicholas. Cluster quality analysis using silhouette score. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 747–748, 2020.

[65] S. S. Sonawane, P. A. Kulkarni, H. Balinsky, A. Balinsky, W. Jin, R. K. Srihari, F. Zhou, F. Zhang, F. Rousseau, and M. Vazigiannis. Graph based representation and analysis of text document: A survey of techniques. *International Journal of Computer Applications*, 96:1–8, 2014.

[66] D. Taraborelli and G. L. Ciampaglia. Beyond notability. collective deliberation on content inclusion in wikipedia. In *2010 Fourth IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshop*, pages 122–125, 2010.

[67] A. Tatar, M. D. De Amorim, S. Fdida, and P. Antoniadis. A survey on predicting the popularity of web content. *Journal of Internet Services and Applications*, 5(1):1–20, 2014.

[68] S. Trani, D. Ceccarelli, C. Lucchese, S. Orlando, and R. Perego. Sel: A unified algorithm for entity linking and saliency detection. In *Proceedings of the 2016 ACM Symposium on Document Engineering*, DocEng '16, page 8594, New York, NY, USA, 2016. Association for Computing Machinery.

[69] Q. Wang. The use of semantic similarity tools in automated content scoring of fact-based essays written by efl learners. *Education and Information Technologies*, 27(9):13021–13049, 2022.

[70] S. Wang and J. Jiang. Machine comprehension using match-lstm and answer pointer, 2016.

[71] P. Wijayatunga. A geometric view on pearsons correlation coefficient and a generalization of it to non-linear dependencies. *Ratio Mathematica*, 30(1), Feb 2017.

[72] C. Xiong, Z. Liu, J. Callan, and T.-Y. Liu. Towards better text understanding and retrieval through kernel entity salience modeling. In *The 41st International ACM SIGIR Conference on Research amp; Development in Information Retrieval*, SIGIR '18, page 575584, New York, NY, USA, 2018. Association for Computing Machinery.

[73] B. Yang, W. tau Yih, X. He, J. Gao, and L. Deng. Embedding entities and relations for learning and inference in knowledge bases, 2015.

[74] Y. Yang, Y. Liu, X. Lu, J. Xu, and F. Wang. A named entity topic model for news popularity prediction. *Knowledge-Based Systems*, 208:106430, 2020.

[75] S. Zhang and K. Balog. Ad hoc table retrieval using semantic similarity. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*. ACM Press, 2018.

[76] W. Zhang, B. Paudel, W. Zhang, A. Bernstein, and H. Chen. Interaction embeddings for prediction and explanation in knowledge graphs. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, page 96104, New York, NY, USA, 2019. Association for Computing Machinery.

[77] W. Zhang, Z. Yin, Z. Sheng, Y. Li, W. Ouyang, X. Li, Y. Tao, Z. Yang, and B. Cui. Graph attention multi-layer perceptron. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 45604570, New York, NY, USA, 2022. Association for Computing Machinery.

[78] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.