



HAL
open science

Contextualization of Web contents through semantic enrichment from linked open data

Amit Kumar

► **To cite this version:**

Amit Kumar. Contextualization of Web contents through semantic enrichment from linked open data. Databases [cs.DB]. Normandie Université, 2021. English. NNT : 2021NORMC243 . tel-03561788

HAL Id: tel-03561788

<https://theses.hal.science/tel-03561788v1>

Submitted on 8 Feb 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Normandie Université

THÈSE

Pour obtenir le diplôme de doctorat

Spécialité INFORMATIQUE

Préparée au sein de l'Université de Caen Normandie

Contextualization of Web Contents through Semantic Enrichment from Linked Open Data

Présentée et soutenue par
AMIT KUMAR

Thèse soutenue le 03/12/2021
devant le jury composé de

M. PATRICE BELLOT	Professeur des universités, Aix-Marseille Université	Rapporteur du jury
M. ADAM JATOWT	Professeur des universités, Université de Innsbruck - Autriche	Rapporteur du jury
MME CECILIA ZANNI-MERK	Professeur des universités, INSA de Rouen Normandie	Membre du jury
M. PIERRE SENELLART	Professeur des universités, École Normale Supérieure de Paris	Président du jury
M. MARC SPANIOL	Professeur des universités, Université Caen Normandie	Directeur de thèse

Thèse dirigée par **MARC SPANIOL**, Groupe de recherche en informatique, image, automatique et instrumentation



UNIVERSITÉ
CAEN
NORMANDIE



Abstract

Thirty years of the Web have led to a tremendous amount of contents and the enormous growth is still ongoing, even accelerating. Thus, Web users are confronted with an abundance of information. While this is clearly beneficial in general, there is a risk of “information overload” and it is very hard for an ordinary user to access, contextualize and digest Web contents. Therefore, there is an increasing need of filtering, categorizing, summarizing, and/or interpretability of Web contents in order to get a proper contextualization. While contents of the early years have been predominantly “simple” HTML documents, more recent ones have become more and more “machine-interpretable” and contribute to the ever growing Linked Open Data (LOD) cloud. LOD provides us a multitude of research opportunities for investigating and harvesting insights about Web contents. In this thesis, we investigate a variety of tasks related to semantic contextualization of Web contents. Specifically, we address three facets in the context of distillation of the Web contents, namely, entity-driven content analysis, semantic annotation & retrieval, and semantic user tracing. Named entities - ideally explicitly and intentionally annotated - pave the way toward a semantic exploration and exploitation of the data. We hypothesize that named entities and their types present in a Web document convey substantial semantic information.

Due to the ample amount of data availability, a user can not go through the entire content of all the documents to identify their relevance. The main topics and semantics of a document can help users to identify the documents according to their relevance. Driven by the mentioned motivation, we perform semantic content analysis in order to gain deeper insight into the Web contents. To accomplish that, we introduce the CALVADOS framework, which utilizes entity-level analytics in order to capture a Web content’s semantics via Linked Open Data. The proposed framework classifies the main topic and reveals the semantic building blocks of a Web content. While we observe that the approach is a first step into a semantic contextualization, but there arises a need to focus the informative onto the essential semantics. To this end, we realize that the most relevant type(s) of named entities can reveal a highly concise characteristics of a Web content. To this end, we develop the PURE framework in order to find the most representative type(s) of an entity. This framework exploits the inherent structural relations among the entity types derived from a knowledge graph in order to derive the representative type(s). The viability of the approach is then validated by performing a thorough study on the most characteristics named entities from Wikipedia.

Having achieved a concise semantic annotation of Web contents, the “natural” next step is to investigate their semantic annotation and retrieval. In a first step, we therefore

address the task of semantic content annotation through the AnnoTag framework, which derives tags via PURE for a Web document and interlinking them to the semantic concepts of the LOD cloud. Based on the quantitative and qualitative evaluations on Web news articles, we prove the viability of our approach and the high-quality of the automatically extracted information. In a second step, we then utilize the semantic annotations for document search and retrieval. As we know that, capturing and exploiting a content's semantic is a key success factor for Web search. So, it is crucial to - ideally automatically - extract the core semantics of the data being processed and link this information with some formal representation, such as an ontology. Connecting both, we introduce the SEMANNOREX framework in order to provide semantically enriched access to a news corpus from Websites and Wikinews.

In our final study, we turn around our investigations and focus on semantic user representation instead of document representations. For that purpose, we explore the concepts derived from a knowledge graph based on the Web documents published/edited by a user. We hypothesize that the semantic characteristics of a user can be revealed by the concepts they are interested in. In order to model user interests, we introduce approach toward semantic user interest tracing called SUIT. In particular, the SUIT framework exploits structural patterns among concepts derived from a knowledge graph in order to analyze user behavior. Our experiments with Wikipedia user data (along with their edited Wikipedia articles) in multiple languages demonstrate proof of the applicability of our methodology.

In this thesis, we have displayed by employing multiple studies that projecting Web contents to the entity-level captures their fundamental semantics. Thus, it provides significant knowledge about the Web contents and, subsequently, comprehensibility. We report novel findings over diverse tasks in an attempt to accomplish our overall goal of a better contextualization of Web contents.

Contents

Abstract	i
1 Introduction	1
1.1 Motivation and Problem	1
1.2 Web Content Analytics	3
1.2.1 Entity-driven Content Analysis	3
1.2.2 Semantic Annotation & Retrieval	4
1.2.3 Semantic User Tracing	5
1.3 Approach and Contributions	5
1.3.1 Semantic Distillation of Web Contents	5
1.3.2 Representative Entity Type Classification	6
1.3.3 Semantic Tagging via LOD Tags	6
1.3.4 Semantic Search via Entity Types	7
1.3.5 Semantic User Interest Tracing	7
1.4 Publications in the Scope of the Thesis	8
1.5 Structure of the Thesis	9
2 Foundations and Technical Background	11
2.1 Knowledge Bases and Linked Open Data	12
2.1.1 Resource Description Framework	12
2.1.2 Resource Description Framework Schema	13
2.1.3 Large Scale Knowledge Bases	14
2.2 Named Entity Recognition and Disambiguation	14
2.2.1 Entity Recognition	15
2.2.2 Entity Disambiguation	15
2.3 Supervised Learning and Classification Methods	18
2.3.1 Random Forests	19

2.3.2	Support Vector Machines	21
2.3.3	Multi-label Classification	23
2.4	Deep Neural Networks	23
2.5	Evaluation	25
3	Related Work	29
3.1	Entity Type Classification Methods	30
3.1.1	Fine-grained Entity Type Classification	30
3.1.2	Target Entity Type Classification	30
3.2	Linked Open Data and General Knowledge	31
3.2.1	Knowledge Based Models	32
3.2.2	Entity-level Analytics	33
3.3	Graph Convolutional Networks	34
3.4	Semantic Annotation & Search	36
3.4.1	Semantic Annotation	36
3.4.2	Semantic Search	38
3.5	User Pattern Analytics	39
3.5.1	User Profile Generation	39
3.5.2	Doppelgänger Detection	40
3.5.3	Authorship Attribution	42
4	Semantic Analysis & Digestion	47
4.1	Overview on CALVADOS	48
4.1.1	Document Preprocessing	48
4.1.2	Semantic Fingerprinting	49
4.1.3	Semantic Exploration	49
4.2	CALVADOS Interface	49
4.2.1	Content Digestion via Semantic Distillation	49
4.2.2	Comparison of Documents Semantics	50
4.3	Findings on Digestion of Semantic Content	51
5	Representative Entity Type Classification	53
5.1	Conceptual Approach	54
5.2	Computational Model	55
5.3	Representative Type(s) Classification	57

5.3.1	Baseline Models	58
5.3.2	Entity Type(s) Classification	58
5.3.3	Random Forest Model	59
5.3.4	Graph Convolutional Network Model	60
5.3.5	PURE	62
5.4	Experimental Evaluation	62
5.4.1	Experimental Setup	62
5.4.2	Model Configurations	65
5.4.3	Classification Results	65
5.5	Findings on Representative Type Classification	67
6	Concise Content Annotation	71
6.1	Conceptual Approach	72
6.2	Semantic Content Tagging Assessment	74
6.2.1	Assessment Dataset & Measures	74
6.2.2	Qualitative Assessment	75
6.2.3	Quantitative Assessment	75
6.3	AnnoTag Interface	76
6.4	Findings on Content Annotation	76
7	Semantic Search via Entity-Types	79
7.1	Conceptual Approach	80
7.1.1	Document Collection	80
7.1.2	Semantic Annotation	81
7.1.3	Semantic Retrieval & Exploration	81
7.2	SEMANNOREX Interface	82
7.3	Findings on Semantic Search	86
8	Semantic User Interest Tracing via Entity-level Analytics	89
8.1	Approach and Contribution	90
8.2	CONCEPTUAL APPROACH	91
8.3	Sub-user Representation	92
8.4	USER INTEREST TRACING MODELS	93
8.4.1	Random Forest based Models	94
8.4.2	Graph Convolutional Networks Models	95

8.4.3	SUIT	96
8.5	EXPERIMENTAL EVALUATION	96
8.5.1	Experimental Setup	96
8.5.2	Model Configurations	98
8.5.3	Experimental Results	99
8.5.4	Sensitivity Analysis	101
8.5.5	Findings on User Interest Tracing	105
9	Conclusion and Outlook	107
9.1	Findings on Semantic Contextualization of Web content	108
9.1.1	Findings on Entity-driven Content Analysis	108
9.1.2	Findings on Semantic Annotation and Retrieval	109
9.1.3	Findings on Semantic User Tracing	110
9.2	Future Research Directions	110
9.2.1	Entity Type Appearance in Events	111
9.2.2	(Dis)-information Spread Detection	111
9.2.3	Semantic-aware Privacy Protection	112
	List of Figures	113
	List of Tables	115
	Bibliography	117

Chapter 1

Introduction

1.1	Motivation and Problem	1
1.2	Web Content Analytics	3
1.2.1	Entity-driven Content Analysis	3
1.2.2	Semantic Annotation & Retrieval	4
1.2.3	Semantic User Tracing	5
1.3	Approach and Contributions	5
1.3.1	Semantic Distillation of Web Contents	5
1.3.2	Representative Entity Type Classification	6
1.3.3	Semantic Tagging via LOD Tags	6
1.3.4	Semantic Search via Entity Types	7
1.3.5	Semantic User Interest Tracing	7
1.4	Publications in the Scope of the Thesis	8
1.5	Structure of the Thesis	9

1.1 Motivation and Problem

The beginning of the Web 2.0 era and World Wide Web platforms have advanced vital societal transformations. The development of information and communication technology (ICT) has provided broad access of the Internet to the global population. Thus, a wide range of communities are contributing in the generation and consumption of Web contents via press articles, social media, blogs, or other platforms. According to [statista](https://www.statista.com/chart/19058/number-of-websites-online/)¹, the total number of Websites reached the threshold of 1.88 billion, and is rapidly increasing. The increasing number of Web users² has inspired many organizations to make available their contents online. Thus, the Web records an immense amount of everyday activities from

¹ <https://www.statista.com/chart/19058/number-of-websites-online/>

² <https://www.internetworldstats.com/stats.htm>

society. This results in many well-acknowledged digitization projects like the Internet Archive³, The New York Times [Sandhaus, 2008], Wikipedia⁴, etc. The Internet Archive alone reports more than 580 billion Web pages and many other digitized contents⁵ since its beginning in 1996. The most popular and widely investigated encyclopedia Wikipedia covers more than 300 languages of the world⁶. English Wikipedia alone consists of more than 6 million articles. According to Wikipedia statistics⁷, 598 new articles are added to the English Wikipedia version each day, whereas 1.9 edits on the existing articles are performed each second.

The other spectrum of the Web consists of a set of organizations where data is a result of the interplay among users and Web applications. These Web applications include social media platforms, such as Twitter, Facebook, or online discussion forums like Reddit, Quora, etc. The most popular social media platform Facebook has more than 2.85 billion active users, whereas Twitter has 397 million active users⁸. Twitter alone is generating 500 million tweets per day⁹. Because of each societal activity across the world, the Web gets a reaction in terms of data generation by the influenced and participated entities. So, it would be fair to say that the Web is a mirror of the natural world in the context of societal events.

Although, the advancement of the digital society results in ample amount of Web contents generation and subsequently accessibility. This appears to be the golden sky toward a more human-centric Web, it not necessarily is. The abundance of data and broad access to the Internet can overwhelm an ordinary Web user with information. For an average Web user, it is very difficult to identify which information is relevant or irrelevant. Hence, in the era of an exponentially growing Web, organization and interpretation of Web contents are very important in order to access relevant information easily. Many recent advancements in the area of Web content management, such as classification of Web contents [Govind et al., 2018b], information diffusion [Govind and Spaniol, 2017], credibility of information [Nakov et al., 2017], etc., have been explored based on text and semantics of a document. This research has given rise to more sophisticated studies. However, at the same time, it provides light on a lot of challenges, such as better contextualization or representation of a Web content, concise information about entities or documents, etc., for qualitative storage, retrieval, recommendation, or interpretation. The Web records data about different aspects of societal events. Semantic enrichment via Linked Opened Data (LOD) can provide a deeper insight into Web contents. The availability of tools like AIDA [Hoffart et al., 2011b] or DBpedia Spotlight [Mendes et al., 2011], which can interlink text documents to LOD has provided us efficient means to capture the semantics of a plain text using the entity-level.

In this thesis work, we address a variety of prediction tasks and challenges, such as prediction of the appropriate type(s) for an entity, semantic annotation of a document,

³ <https://archive.org/>

⁴ <https://www.wikipedia.org/>

⁵ <https://archive.org/about/>

⁶ <https://en.wikipedia.org/wiki/Wikipedia>

⁷ <https://en.wikipedia.org/wiki/Wikipedia:Statistics>

⁸ <https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/>

⁹ <https://www.internetlivestats.com/twitter-statistics/>

semantic search over Web documents, and user behavior pattern generation. The appropriate type(s) prediction task discovers the most suitable entity type(s) for a given named entity, based on all its facets defined in a knowledge base (KB). Further, we extend our research to semantic annotation of a document via different entity-level analytics strategies. Next, we address the task of semantic search over a well-defined entity type ontology and several Web corpora. Finally, we shift our focus from Web document representation to Web user representation and generate several user behavior patterns based on individual contribution towards Web contents.

1.2 Web Content Analytics

Motivated by the above discussion, the central research theme of this thesis is to address various tasks related to contextualization of Web contents. Contextualization of Web contents can be well captured by incorporating entity-level analytics. Raising Web contents to the entity-level provides us with the capacity to incorporate knowledge about entities stored in knowledge bases (KBs). The availability of several publicly available large scale knowledge sources, such as DBpedia [Auer et al., 2007] or YAGO [Suchanek et al., 2007] contain a wealth of facts related to named entities. Moreover, the emergence of named entity disambiguation systems (NED) like DBpedia Spotlight [Mendes et al., 2011] or AIDA [Hoffart et al., 2011b], establishes the link between unstructured text documents and “Linked Opened Data (LOD)” by giving semantics to plain text and bringing it to the entity-level. Thus, the structural knowledge accumulated in these KBs can be exploited and aggregated to enhance the performance of the above-mentioned high level analytics tasks. To this end, we highlight the various research questions addressed in the context of the thesis, in the following subsections.

1.2.1 Entity-driven Content Analysis

We observe that, for a text describing an event, there are specific recurring patterns of entity types appearing together. For instance, in the case of ‘natural disasters’, entities like **organizations**, **countries**, **presidents** frequently appear together, whereas in the case of ‘political events’, entities like **parties**, **leaders**, **business-persons** appear together. The interplay among different entities can be well captured by the tools like AIDA or DBpedia Spotlight in cooperation with the KBs. Thus, we commence our research in this thesis by asking the following question:

RQ1 How to explore a Web content with respect to named entities mentioned in it?

Further, the exploitation of named entities and their types is a valuable asset in getting a better contextualization of a Web document. However, sometimes one might be overwhelmed by too much information. For example, a recent article¹⁰ containing Joe Biden

¹⁰ <https://www.bbc.com/news/world-us-canada-58591095>

deals with his position of president. But, Biden has 41 facets (as mentioned in YAGO, by solely considering WordNet types) like **lawyer**, **president**, **senator**, etc., which are not equally relevant. This leads us to the our next research question:

RQ2 Which facet(s) are the most expressive and representative for a named entity?

1.2.2 Semantic Annotation & Retrieval

After addressing the above tasks, the question arises, how to utilize these concise entity types in order to support a semantic access to the contents? We therefore ploy the type annotation of the named entities obtained in the previous study to represent a document. In particular, we aim at supporting semantic annotation and retrieval. Although semantic annotation has been addressed before [Bikakis et al., 2010, Giannopoulos et al., 2010, Afiontzi et al., 2013, Eckart de Castilho et al., 2016, Medeiros et al., 2018], our approach is purely based on entity-level information. To this end, we target the process of semantic annotation first by asking:

RQ3 How to concisely annotate a document and interlink it with concepts of LOD?

Document annotation and retrieval are two different dimensions in the context of Web analytics. Some studies focus on document representation and interpretation [Flekova and Gurevych, 2016]. However, it is challenging for an ordinary user to get the context of these representations and, subsequently, interpret those contents. Therefore, our study aims at supporting a semantically-driven document retrieval.

Based on the semantic annotation of documents introduced before, we provide a semantic search interface. Semantic search is a retrieval technique that results in documents based on intent and contextualization of query and documents. In previous works, there are few studies [Hoffart et al., 2014, Hasibi et al., 2017, Gupta and Berberich, 2019, Ho et al., 2020], which approach the task of semantic search. In contrast to the before mentioned approaches, we aim at leveraging solely entity type information and the taxonomic structure extracted from the underlying KB. To this end, we postulate the following research question:

RQ4 Which retrieval method to apply in order to support ontology-driven retrieval?

1.2.3 Semantic User Tracing

We extensively utilize entity-level analytics in combination with LOD to address the challenges and the solutions of the aforementioned tasks. In our previous studies, we demonstrated the viability of getting a better contextualization of a Web content. In our last study, we now “turn around” our analytics and aim at predicting the user interest in Web contents based on their semantics. To this end, we shift and focus our study on user behavior analytics. Thus, we generate contextualized user publishing/editing behavior patterns in order to trace and predict if Web contents will be absorbed by a user. This results in the following research question:

RQ5 How to predict user behavior by exploiting the semantics of contents they are interested in?

1.3 Approach and Contributions

The notable contributions of this dissertation are five-fold. The first is the semantic distillation of Web contents. The proposed system defines the content semantic by employing entity-level analytics and visualizes it. Our second major contribution is representative entity type classification. For this purpose, we present a framework that exploits the inherent semantic among the types to predict the most suitable type(s) for an entity. Thirdly, we propose an approach of concise content annotation as a means of supporting the process of digital curation via entity-level analytics. The fourth contribution is semantic search via entity types. To this end, an online demonstrator has been deployed. Our final contribution is the prediction of user interest tracing via semantic patterns. To accomplish this, we propose a methodology based on users’ publishing/editing behavior towards Web documents. This system utilizes concepts corresponding to Web contents for generating patterns belonging to a user. We provide a brief outline of each of the above-mentioned contributions in the following subsections.

1.3.1 Semantic Distillation of Web Contents

Semantic distillation is the task of automatically assessing the Web contents and filtering, classifying and/or summarizing Web contents semantically. In order to help consumers in efficiently deriving the semantics from Web contents, we develop the CALVADOS (Content AnaLytics ViA Digestion Of Semantics) system, an extension of [Govind et al., 2018b] which raises contents to the entity-level and digests its inherent semantics. CALVADOS is a semantic content analytics platform, which facilitates end users to explore the main topic and visualizes the elementary semantic blocks of a Web document. At the same time, it also allows a user to compare the contents of two Web documents semantically. The key intuition behind this work is as follows:

“A Web document can be attributed by the named entities it contains”. It can be broken down in the following postulates:

Hypothesis 1.1 Entity type information constitutes the semantics of a document.

Hypothesis 1.2 Contextualization of a Web content can be obtained by employing a compact representation like “semantic fingerprint”.

Hypothesis 1.3 The exploration and visualization of the semantics of a document assists in getting a deeper insight into the document.

1.3.2 Representative Entity Type Classification

The representative entity type classification task deals with assigning the most appropriate type(s) or label(s) for a given named entity. This task is highly significant in explaining the better contextualization of a Web content as named entities are predominantly annotated without any order of importance associated. In this work, we derive all the candidate entity type(s) from a knowledge base taxonomy (in this case, YAGO [Hoffart et al., 2013]) defined on the most populated types. We investigate several baselines based on types information. In particular, we introduce the PURE (Pattern Utilization for Representative Entity type classification) framework, which aims at exploiting solely structural patterns derived from the knowledge graph in order to “purify” the most representative type(s) associated with a named entity. Moreover, we publish a newly designed dataset for the task, which is publicly available, to encourage other researchers for more investigation. We formulate the following axiom for the current task:

“Named entities of certain types share a multitude of common and (at the same time) characteristic facets”. Further, it can be extended as:

Hypothesis 2.1 Named entities are annotated with a plenitude of types, but all of these types are not equally important.

Hypothesis 2.2 A graph convolutional network model assisted by structural information from LOD, can learn and is able to identify the typical patterns shared by the common and characteristics facets.

Hypothesis 2.3 Identified common patterns can further be used to derive the most governing type(s) for a named entity.

1.3.3 Semantic Tagging via LOD Tags

Semantic tagging is the task of generating related tags based on the content of a document. It has wide applications in the field of data and digital curation, such as digital library, document organization, categorization, or search. In order to support the human in data curation, we introduce an annotation tagging system called “AnnoTag”. AnnoTag aims at providing concise content annotations by employing entity-level analytics in order to derive semantic descriptions in the form of tags. In particular, we are generating “Semantic LOD Tags” (Linked Open Data) that allow an interlinking of the derived tags with the LOD cloud. We compare various entity-level annotation methods and highlight the importance of concise content annotation based on qualitative as well as quantitative evaluations. The

AnnoTag system has been deployed in order to support users in visualizing and exploring semantic tags of a Web document. We formulate the following hypothesis.

“Entity-level tag derived by the PURE framework can reveal the semantic characteristics of a document”. In turn, we assume the following postulates:

Hypothesis 3.1 Entity-level analytics characterizes a document by generating the most concise and, at the same time, the most relevant tags.

Hypothesis 3.2 Associated named entities and their most concise type(s) can be explored by exploiting NED and LOD.

Hypothesis 3.3 Alignment of the generated tags with the help of LOD can bridge the gap between human-understandable annotations and semantic concepts.

1.3.4 Semantic Search via Entity Types

The task of semantic search aims at enhancing search quality by incorporating the user’s objective and the contextual meaning of a query term in a searchable space. It has a broad range of applications (e.g., Web search), and at the same time, it is highly relevant to the research questions highlighted in the context of this dissertation. In the current study, we introduce the SEMANNOREX (SEMantic ANNOtaion, Retrieval, and EXploration) tool. SEMANNOREX is built upon a very fine-grained entity type taxonomy (extracted from the YAGO knowledge base [Hoffart et al., 2013]). It exploits the PURE framework [Kumar et al., 2020] for document annotation and utilizes several similarity measures to retrieve the documents. We also release an online SEMANNOREX interface where an individual can explore and visualize the semantics contributed to the retrieval of the documents. Our semantic search methodology is driven by the following hypothesis:

“Entity-level analytics can expose the various semantic characteristics of a query and a document”. Therefore, we hypothesize the following:

Hypothesis 4.1 Implicit characteristics of a document can be revealed by exploiting the types of the named entities contained.

Hypothesis 4.2 A semantic similarity technique based on entity types can provide highly concise results.

Hypothesis 4.3 Semantic search should allow interactive access to the retrieved contents along with the visualization of the relevant semantic types.

1.3.5 Semantic User Interest Tracing

The user interest tracing task attempts to analyze user’s behavior based on relevant Web contents for each individual. This relevancy can be defined based on visited Websites, interaction among users, contribution in a Web content, etc. In the current work, we aim at generating user patterns by investigating individual’s contributing behavior towards

Web contents. This task has high importance as it provides insights about an individual’s interests, so it takes one step further in finding the societal relevance of a user. In order to model user’s interests, we introduce the SUIT (Semantic User Interest Tracing) framework. SUIT aims at exploiting the concepts associated with the Web contents contributed by a user. In particular, it utilizes the inherited structural relationships present among the concepts derived from the knowledge graph YAGO in order to identify the concerned Web contents corresponding to a user. As a vital part of this work, we also investigate our methodology beyond the scope of English language users and show that our approach is purely semantic, language-agnostic, and applicable to languages with less ample resources. Our user interest tracing model is based on the following hypothesis:

“User can be characterized by the concepts they are interested in”. This turns out to be in following hypothesis:

Hypothesis 5.1 User behavior can be captured by individual’s contribution towards Web documents assisted by the concepts from LOD.

Hypothesis 5.2 User patterns can be generated by exploiting the inherent semantic relations among the concepts derived from a knowledge graph.

Hypothesis 5.3 These patterns can further be utilized in identifying the relevant or interested documents for a Web user.

1.4 Publications in the Scope of the Thesis

The work studied in the scope of this thesis has been published at renowned international conferences. More specifically, we have focused the conferences addressing information extraction in the context of Semantic Web technologies and Web science. The study of semantic analysis of Web documents (*i.e.*, the CALVADOS framework) has been published at the 16th Extended Semantic Web Conference (ESWC) [Govind et al., 2019b]. This conference is regarded as a major venue for publishing scientific results and innovations in the field of Semantic Web technology. We submitted and presented our subsequent research methodology in Doctoral Consortium of the 42nd European Conference on Information Retrieval (ECIR) [Kumar, 2020]. ECIR is a premier forum addressing the innovations and new scientific results in the domain of information retrieval. The PURE framework for representative entity type prediction has been published at the 12th International Web Science Conference [Kumar et al., 2020]. The Web Science Conference is one of the leading venues for publishing the novel methods and findings to enhance knowledge about the Web and its impacts. The conference intends to gather researchers from a variety of domains, such as computer and information sciences, media studies, political science, etc. Our work on the semantic annotation of documents based on “AnnoTag” has been published at the 25th International Conference on Theory and Practice of Digital Libraries (TPDL) [Kumar and Spaniol, 2021a, Kumar and Spaniol, 2021b]. The TPDL conference is a reputed venue for works related to intersects with digital libraries, such as document annotation or tagging. Our work on semantic search, the SEMANNOREX framework is

published at 30th The Web Conference (WWW) [Kumar et al., 2021]. The Web Conference is a premier forum addressing the creativity and research about the evolution of World Wide Web. This conference also brings together researchers, developers, policy makers, etc.

Govind, Amit Kumar, Céline Alec, and Marc Spaniol (2019). CALVADOS: A Tool for the Semantic Analysis and Digestion of Web Contents. In Proceedings of the 16th Extended Semantic Web Conference (ESWC 2019), Slovenia, June 2–6, 2019, pages 84–89.

Amit Kumar (2020). Towards a better Contextualization of Web Contents via Entity-level Analytics. In Proceedings of European Conference on Information Retrieval (ECIR 2020), April 14–17, 2020, pages 613–618.

Amit Kumar, Govind, Céline Alec, and Marc Spaniol (2020). Blogger or President? Exploitation of Patterns in Entity Type Graphs for Representative Entity Type Classification. In Proceedings of the 12th International ACM Web Science Conference (WebSci '20), Southampton, UK, July 7–10, 2020, pages 59–68.

Amit Kumar, Govind, and Marc Spaniol (2021). Semantic Search via Entity-Types: The SEMANNOREX Framework. In Companion Proceedings of the 30th Web Conference (WWW 2021), April 12–23, 2021, pages 690–694.

Amit Kumar and Marc Spaniol (2021a). AnnoTag: Concise Content Annotation via LOD-Tags derived from Entity-level Analytics. In Proceedings of the 25th International Conference on Theory and Practice of Digital Libraries (TPDL 2021), September 13–17, 2021, pages 175–180.

Amit Kumar and Marc Spaniol (2021b). Semantic Tagging via Entity-level Analytics: Assessment of Concise Content Tagging. In Proceedings of the 25th International Conference on Theory and Practice of Digital Libraries (TPDL 2021), September 13–17, 2021, pages 97–105.

Amit Kumar and Marc Spaniol (2022). Semantic User Interest Tracing via Entity-level Analytics. (Submitted to The Web Conference 2022).

1.5 Structure of the Thesis

The rest of the thesis is structured as follows. In Chapter 2, we present a detailed description of fundamentals and technical prerequisites for work performed as the contribution of the thesis. We discuss the technicality of Linked Open Data (LOD) and large scale knowledge bases (KBs). Further, we provide an overview of different named entity recognition and disambiguation systems. Also, we discuss different machine learning techniques and graph convolutional networks as they are employed in our contributions. Additionally, we explain the evaluation metrics which are underlying our experiments.

Subsequently, in Chapter 3, we compile the prior works and research in detail that are relevant to our studies. We commence our survey with the work related to the identification of appropriate type(s) for an entity. Then, various works based on knowledge bases, ontology, and entity-level analytics are examined. We discuss several applications of graph convolutional networks in different domains. Further, we provide a detailed discussion on the annotation of documents and retrieval of Web documents based on semantic search. Finally, we wrap up this chapter by providing a review about user behavior and pattern.

Chapter 4 is related to the task of semantic analysis of Web documents. We discuss our proposed framework CALVADOS (Content AnaLytics ViA Digestion Of Semantics), an extension of [Govind et al., 2018b]. We present how entity-level analytics can be employed to automatically classify the main topic of a Web content and reveal the semantic building blocks associated with the corresponding document.

Chapter 5 is dedicated to the task of representative entity type classification. To this end, we introduce the PURE (Pattern Utilization for Representative Entity type classification) framework based on graph convolutional networks architecture. Here, we discuss different strategies to find the most representative type(s) exclusively based on structural information derived from a knowledge graph. We also mention the dataset crawled and construction methodology in this chapter. In the end, we conclude with the overall findings of the performed experiments over Wikipedia entities in different aspects.

Subsequently, we explain our study on concise content annotation in chapter 6. Here, we come out with the AnnoTag system that provides semantic LOD tags to a document via entity-level analytics. We also discuss other baseline approaches and report our quantitative and qualitative results. A Web interface has been released, which is freely available for the public and has been demonstrated in the current chapter.

Furthermore, in Chapter 7, we present our work on semantic search. Semantic search captures and exploits a content’s semantics to facilitate Web search. Thus, we introduce the SEMANNOREX (SEMantic ANNOtaion, Retrieval, and EXploration) framework in order to provide semantically enriched access to a news corpus from Websites and Wikinews. We discuss several strategies for documents retrieval and report quantitative and qualitative evaluations. As part of this work, a Web demonstrator has been deployed.

In Chapter 8, we address the task of user interest tracing via user contributing behavior towards Web contents. Here, we propose the SUIT (Semantic User Interest Tracing) framework based on novel graph convolutional networks. We derive user publishing/editing patterns solely based on the concepts corresponding to the contributed Web contents for the respective users as described in the chapter. This chapter also reports the results with Wikipedia users along with the sensitivity studies in multiple languages.

Finally, Chapter 9 concludes the entire studies performed in the perspective of this thesis. It also provides an outlook for the presumptions of future research direction that have been pointed out by this thesis work. Albeit, our findings based on the experiments performed have been summarized in each chapter, this chapter aggregates the entire work in a comprehensive manner and highlights a wider perspective.

Chapter 2

Foundations and Technical Background

2.1	Knowledge Bases and Linked Open Data	12
2.1.1	Resource Description Framework	12
2.1.2	Resource Description Framework Schema	13
2.1.3	Large Scale Knowledge Bases	14
2.2	Named Entity Recognition and Disambiguation	14
2.2.1	Entity Recognition	15
2.2.2	Entity Disambiguation	15
2.3	Supervised Learning and Classification Methods	18
2.3.1	Random Forests	19
2.3.2	Support Vector Machines	21
2.3.3	Multi-label Classification	23
2.4	Deep Neural Networks	23
2.5	Evaluation	25

In this chapter, we provide the foundation of technical background which is required to comprehend the tasks performed in the scope of this thesis. We commence with the description of knowledge bases and linked open data. Afterward, we provide an overview of named entity extraction and disambiguation systems. Next, we provide the description of different machine learning techniques for supervised classifications problems, followed by graph convolutional networks for classification. Finally, we conclude with a brief explanation of the relevant assessment measures.

2.1 Knowledge Bases and Linked Open Data

Knowledge bases (KBs) portray the common knowledge in a machine understandable format. The term knowledge follows the principle introduced by W3C¹¹ (World Wide Web Consortium), and is referred to the RDF (resource description framework) concepts¹². The Linked Data specifies a set of principles to publish and make a connection to the structured data available on the Web [Bizer et al., 2009a]. The linked data, which is publicly available (under an open license) to access and practice, is termed as “Linked Open Data (LOD)”¹³. The KBs store information about different entities, their characteristics and connection to other entities¹⁴. A unique identifier symbolizes each entity in the knowledge base. The facts and the associations between the entities are represented in a standardized form. These structured standard formats of knowledge are the bottom-line behind the concept of Semantic Web [Berners-Lee et al., 2001]. The term “Semantic Web” coined by Berners-Lee et al. conveys structure to the substantial Web contents [Berners-Lee et al., 2001]. Many knowledge bases have been designed based on the principles of Linked Data and inter connected to “The Linked Open Data Cloud”¹⁵. There has been growing interest in the KBs since the evolution of “Semantic Web” in the early 2000s. The KBs act as the backbone for various kinds of semantic aware applications, such as academic search [Xiong et al., 2017], question answering [Berant et al., 2013, Yahya et al., 2012], and semantic search [Bast et al., 2016].

2.1.1 Resource Description Framework

The Resource Description Framework (RDF) is the blueprint of a data model introduced by W3C¹⁶, which is a standard for the representation of the information available on the Web [W3C et al., 2014]. This standard data model acts as a mentor for the creation of a graph which is formed by `subject-predicate-object` triplets. The basic unit of these triplets are resources, predicates, and data-type literals, which are used to represent facts about resources. RDF datasets contain one default graph, and zero or more named graphs. A real world entity or an abstract concept is referred as “resource” in RDF. The set of all the resources in a KB is represented by \mathcal{R} . Literals are used for data values of several types such as numbers, strings, times, dates, etc., and represented by \mathcal{L} . Predicates \mathcal{P} are the other important part of the RDF data model, which represent the association between resources, or a resource and a literal. Then formally, a KB can be illustrated as the projection between the above-mentioned building units of the RDF data model. Thus, a KB \mathcal{K} can be interpreted by the following mathematical relation: $\mathcal{K} = \mathcal{R} \times \mathcal{P} \times (\mathcal{L} \cup \mathcal{R})$. In other words, a KB consists of a large set in which the elements of the set are in the form of triple: $t = \langle s, p, o \rangle$, where $s \in \mathcal{R}$, $p \in \mathcal{P}$, and object o stands for a literal or a resource,

¹¹ <https://www.w3.org/>

¹² <https://www.w3.org/TR/rdf-concepts/>

¹³ https://en.wikipedia.org/wiki/Linked_data

¹⁴ The terminology ‘knowledge graph’ represents the same concept. In literature, knowledge base and knowledge graph are used interchangeably, while the latter explicitly specifies that the information is represented as a graph.

¹⁵ <https://lod-cloud.net/>

¹⁶ <https://www.w3.org/RDF/>

i.e., $o \in \mathcal{L} \cup \mathcal{R}$. Table 2.1 shows an excerpt of RDF graph for resource `Alan_Turing` as in the DBpedia KB.

Subject	Predicate	Object
<code>dbr:Alan_Turing</code>	<code>rdf:type</code>	<code>dbo:Scientist</code>
<code>dbr:Alan_Turing</code>	<code>dbo:birthPlace</code>	<code>dbr:Maida_Vale</code>
<code>dbr:Alan_Turing</code>	<code>dbo:academicDiscipline</code>	<code>dbr:Computer_science</code>
<code>dbr:Alan_Turing</code>	<code>dbo:academicDiscipline</code>	<code>dbr:Cryptanalysis</code>
<code>dbr:Alan_Turing</code>	<code>dbo:almaMater</code>	<code>dbr:University_of_Cambridge</code>
<code>dbr:Alan_Turing</code>	<code>dbo:award</code>	<code>dbr:Smith's_Prize</code>
<code>dbr:Alan_Turing</code>	<code>dbo:knownFor</code>	<code>dbr:Turing_machine</code>
<code>dbr:Alan_Turing</code>	<code>dbo:knownFor</code>	<code>dbr:Cryptanalysis_of_the_Enigma</code>

Table 2.1: Excerpt of an RDF Graph for Resource `Alan_Turing` in DBpedia; `dbr` and `dbo`, symbolize a DBpedia Resource and the DBpedia Ontology Schema, respectively

2.1.2 Resource Description Framework Schema

The decisive step for the representation of real world information is to categorize the RDF resources by classes. The Resource Description Framework Schema (RDFS) is a set of classes with specific characteristics which provides elementary units for the explanation of ontologies [W3C et al., 2014]. RDFS is an augmentation of the basic data modeling vocabulary provided by the RDF data¹⁷. The RDF resources have been categorized into an intended structure by employing RDFS. As illustrated in Table 2.2, an RDFS class is described by a set of triples. Thus, a resource s is allocated to a class c by utilizing a predicate and the triplet is represented as $\langle s, predicate, c \rangle$.

Subject	Predicate	Object
<code>dbo:Scientist</code>	<code>rdf:type</code>	<code>owl:Class</code>
<code>dbo:Scientist</code>	<code>rdfs:subClassOf</code>	<code>dbo:Person</code>
<code>dbo:academicDiscipline</code>	<code>rdf:type</code>	<code>rdf:Property</code>
<code>dbo:award</code>	<code>rdf:type</code>	<code>rdf:Property</code>
<code>dbo:Agent</code>	<code>owl:disjointWith</code>	<code>dbo:Place</code>

Table 2.2: RDFS Class Definition - Excerpt of RDF Triples in DBpedia Ontology Schema

RDFS facilitates a precise structure of hierarchy among all the classes allocated to the resources belong to a KB. For instance, the resource `Alan_Turing` is a member of class `Scientist`. It can be depicted from the Table 2.1. Now, by exploiting the RDFS concepts and predicate `rdfs:subClassOf`, it can be derived that `Alan_Turing` is the member of `Person` class as well. To this end, RDFS grants an efficient mechanism to specify the structure and the convenient hierarchy among the classes.

¹⁷ <https://www.w3.org/TR/rdf-schema/>

2.1.3 Large Scale Knowledge Bases

Linked Open Data (LOD) has received a lot of attention with the declaration of standards of “Semantic Web”¹⁸ by W3C. Currently, the number of publicly available RDF data sets are in thousands¹⁹, and is exponentially increasing. Since the last decade, a considerable number of general purpose knowledge bases have been constructed, such as DBpedia [Auer et al., 2007, Bizer et al., 2009b, Lehmann et al., 2014], Freebase [Bollacker et al., 2008], NELL [Carlson et al., 2010], OpenIE [Banko et al., 2007], YAGO [Suchanek et al., 2007, Hoffart et al., 2013, Mahdisoltani et al., 2013, Pellissier Tanon et al., 2020], and Wikidata [Vrandečić and Krötzsch, 2014]. Indeed, the influence of knowledge bases can be observed in commercial search engines. Some of the industrial attempts to create generic and domain specific knowledge bases include Bing’s Satori [Qian, 2013], Google’s Knowledge Vault [Dong et al., 2014] and LinkedIn’s knowledge graph [He et al., 2016].

Knowledge bases can be created either automatically (e.g., NELL, OpenIE, etc.) or with the help of human curators in a semi-automatic fashion (e.g., DBpedia, YAGO, Wikidata, etc.). Information extraction techniques are exploited for the extraction of facts for the former group of KBs. Albeit the accuracy of these knowledge bases is sufficiently high, their qualities are still below in comparison with the human created knowledge bases [Suchanek et al., 2008]. For the latter class, the complexity of constructing a KB would be very high if few curators are contributing. The wide access to the internet in society and the betterment of crowd sourcing platforms have created relatively easy opportunities to take advantage of the crowd’s wisdom. Wikidata²⁰ is one of the examples of such a community project which is actively curated by volunteers from different geographical localities and cultures. Another well-known technique to build a KB is by automatically extracting meaningful facts from semi-structured data like Wikipedia. DBpedia and YAGO are two well-known examples of such KBs. These KBs contain facts about several millions of entities and store them in RDF triples format. These facts are accessible in more than 100 languages for both the KBs.

2.2 Named Entity Recognition and Disambiguation

This section provides an overview of the state-of-the-art methodology in the Named Entity Recognition and Disambiguation task, which is composed of the two subtasks: Named Entity Recognition and Named Entity Disambiguation.

Named Entity Recognition (NER) aims at the detection of named entities mentioned in a raw text (cf. Section 2.2.1). Furthermore, Named Entity Disambiguation (NED) already assumes that the recognition of the named entities has been completed and intends to map them onto their corresponding instance in a KB (e.g., DBpedia, Wikidata, or YAGO) (cf. Section 2.2.2). The combined task of the aforementioned subtasks is called Named Entity Recognition and Disambiguation (NERD) or Entity-linking. At first, this task recognizes the named entities mentioned in a text and then links them to the entries in a KB. Broadly,

¹⁸ <https://www.w3.org/standards/>

¹⁹ <http://sansa-stack.net/distlodstats/>

²⁰ <https://www.wikidata.org/>

two classes of systems have been developed for NERD. The first class of systems considers recognition and disambiguation tasks independently and develops separate models for both the subtasks. In the end, the developed independent models are combined. Thus, recognition of the named entities from the developed model or the already extracted (gold-standard) named entities act as the input for the disambiguation model. In contrast to the first class of systems, the second class develops an end-to-end model for the NERD task. These models receive the raw text as input and generate the disambiguated named entities as an output. They operate simultaneously on the detection of named entities and their disambiguation to the corresponding entry of a KB.

2.2.1 Entity Recognition

The Named Entity Recognition (NER) task is one of the fundamental information extraction tasks which aims at identifying the mention named entities in a given text and tag them to the most relevant class [Borthwick et al., 1998, Finkel et al., 2005, Prokofyev et al., 2014]. Some of these classes are person (PER), organization (ORG), location (LOC), time, and date. Broadly, the NER system acts in two steps: first, entity identification, and then, entity classification. Named entities mentioned in a text are identified in the first step, and subsequently, these identified named entities are assigned to one of the given classes.

Earlier, dictionary-based and rule-based methods have been developed for NER [Song et al., 2015]. Due to bad generalization properties of these methods, researchers have introduced methods based on machine learning techniques, such as Conditional Random Field (CRF) and Hidden Markov Model (HMM) [Zhao, 2017, Li et al., 2008]. The performance of classical machine learning systems depends on the amount and quality of the manually created features. The recent development of deep learning and its utility in multiple NLP tasks [Torfi et al., 2020] has encouraged researchers to adopt it for solving the problem of NER. It is driven by cognitive computing and avoids the complexity of creating the hand-crafted features. Some of the most popular NER systems include Stanford NER²¹ [Finkel et al., 2005], NLTK²² [Bird et al., 2009], and spaCy²³. A NER system does not deal with resolving ambiguity among the extracted named entities.

2.2.2 Entity Disambiguation

The alignment of the named entities mentioned in an unstructured text to a canonical entity in a knowledge base is known as Named Entity Disambiguation (NED) task. Inherited ambiguity among the mentioned named entities in a text is the main challenge of the NED task [Ling et al., 2015], since the same surface form of a word can refer to several entities in a text. For instance, in Figure 2.1, Turing can be referred as `Turing_test`, `Turing_Award` or the scientist `Alan_Turing` in this context. The online encyclopedia Wikipedia acts as the main repository for various disambiguation systems. The link anchors mentioned in

²¹ <https://nlp.stanford.edu/software/CRF-NER.html>

²² <https://www.nltk.org/>

²³ <https://spacy.io/>

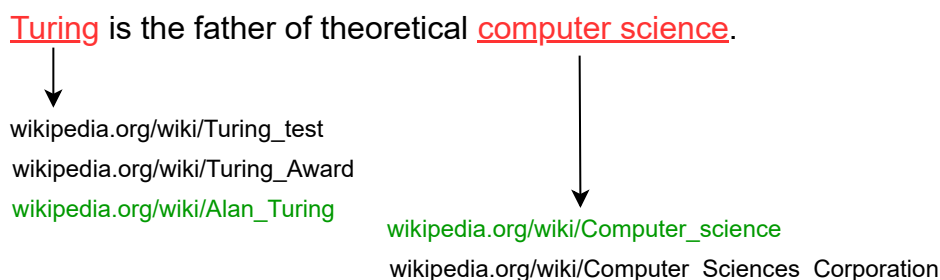


Figure 2.1: Example of Named Entity Disambiguation for Turing and computer science

the Wikipedia pages provide information about variations of mentioned entities for several systems. Furthermore, Wikipedia pages generally serve as the ground truth labels for physical entities.

Earlier works on named entity disambiguation relied upon the contextual information of a document and the referred entities [Cucerzan, 2007, Mihalcea and Csomai, 2007]. [Cucerzan, 2007] exploit category and contextual information extracted from Wikipedia for the candidate entities. The authors compute the agreement between the extracted information and the context of the document. The Wikify! system extracts the important concepts in a text using a keyword extraction technique and automatically maps them to the respective Wikipedia pages [Mihalcea and Csomai, 2007]. Subsequently, Milne and Witten proposed the techniques of *relatedness* (Equation 2.3) and *commonness* [Milne and Witten, 2008]. These two techniques are regarded as the most important attributes for entity disambiguation. In contrast to these approaches, which disambiguate one mentioned entity at a time, collective entity disambiguation explores the relatedness among the entities to conduct combined disambiguation of all the mentioned entities in a text [Han et al., 2011, Hoffart et al., 2011b, Kulkarni et al., 2009, Sen, 2012]. The approach of [Kulkarni et al., 2009] attempts to disambiguate the entities in a collective manner by the measurement of global coherence among the entities. Some of the most popular entity disambiguation systems include AIDA [Hoffart et al., 2011b, Yosef et al., 2011], DBpedia Spotlight [Mendes et al., 2011] and TagMe [Ferragina and Scaiella, 2010]. Recently, researchers have been incorporating distributional semantics and neural networks techniques. [Moreno et al., 2017] proposed the model, which learns joint embeddings of words in a text and named entities in a KB in order to disambiguate entities.

We subsequently provide an overview of the most recognized publicly available entity disambiguation systems.

DBpedia Spotlight

DBpedia Spotlight [Mendes et al., 2011] generates the annotations for a text document as DBpedia resources [Lehmann et al., 2014]. It allows end users the annotations according to their needs through DBpedia Ontology and other quality measures. DBpedia Spotlight operates mainly in four phases. The first phase, called *spotting*, recognizes all the phrases that might be a mention of possible DBpedia resource. The subsequent phase, *candidate selection* is performed to align the spotted phrases to the potential disambiguation for that phrase from DBpedia resources. The next phase *disambiguation* concludes the best

amongst the potential candidate resources by exploiting the neighboring context around the mention. In the end, the annotation operation can be customized by users according to their needs through the *configuration* parameters, such as destined resource set, resource prominence, topic relevancy, confidence of disambiguation, and contextual ambiguity.

AIDA

Accurate Online Disambiguation of Named Entities (AIDA) formulates the task of entity disambiguation problem as a dense sub-graph which estimates the best joint mention entity alignment [Hoffart et al., 2011b, Yosef et al., 2011]. The dense sub-graph is built upon the mentions weighted graph and candidate entities. AIDA utilizes Stanford NER²⁴ for distilling the mentions from a given text and YAGO2 [Hoffart et al., 2011a] knowledge base as the destined entity set. The entire framework is constructed by combining three measurements: the prior popularity of a mention entity, similarity computation between the context of a mention and its candidate entity, along with the coherence computation among all the candidate entities for all the mentions simultaneously. AIDA intends for the best disambiguation alignments by optimizing the combined objective function of the previously mentioned measures. The objective function is formulated in Equation 2.1. Here, each mention is represented by $m_j, j = 1, \dots, k$, select entity candidates p_{i_j} , one per mention, such that

$$\alpha \cdot \sum_{j=1..k} \text{prior}(m_j, p_{i_j}) + \beta \cdot \sum_{j=1..k} \text{sim}(\text{cxt}(m_j), \text{cxt}(p_{i_j})) + \gamma \cdot \text{coherence}(p_{i_1} \in \text{cnd}(m_1) \dots p_{i_k} \in \text{cnd}(m_k)) = \max! \quad (2.1)$$

where $\alpha + \beta + \gamma = 1$, $\text{cnd}(m_j)$ is the set of possible meanings of m_j , $\text{cxt}()$ represents the context of mentions and entities, respectively, and $\text{coherence}()$ is defined for a set of entities and formulated in Equation 2.2.

Figure 2.2 illustrates the mention-entity graph generated by AIDA. Mentions and entities serve as the nodes for the weighted mention-entity graph, which is undirected in nature. The weight between a mention and an entity is defined based on as similarity measure or an aggregation of similarity measure and popularity. The edge between entities is weighted based on the *coherence* computation (Equation 2.2) among the entities, or type distance, or some combinations of the previous two.

The *relatedness* between two entities or Wikipedia pages p_1 and p_2 is computed based on shared Wikipedia incoming links and formally defined by Equation 2.3. This concept was introduced by [Milne and Witten, 2008]. AIDA utilizes the concept of *relatedness* for *coherence* computation. In Equation 2.3, P_i represents the set of all incoming links for page p_i and P is the set of all Wikipedia pages. Based on the above-mentioned concepts AIDA provides three types of disambiguation schemes (*prior*, *prior + similarity*, and *prior + similarity + coherence*). Recently, AIDA-light has been developed, which utilizes less complicated features than AIDA and, thus, diminishing the computational cost [Nguyen et al., 2014].

²⁴ <https://nlp.stanford.edu/software/CRF-NER.html>

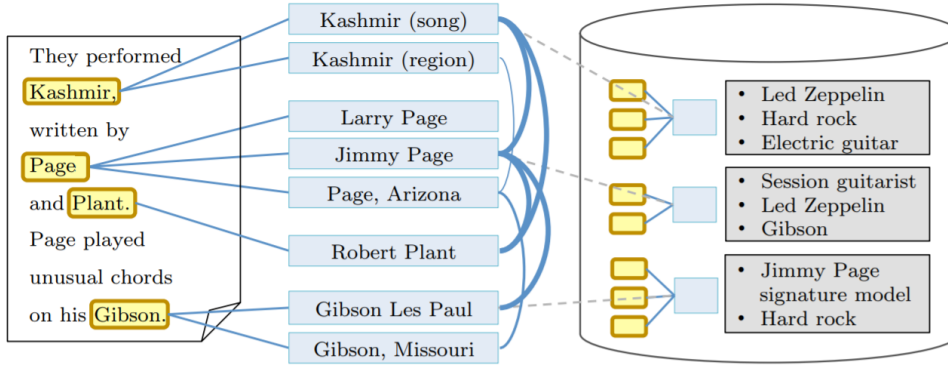


Figure 2.2: A Sample Text Snippet of Mention-entity Graph with Ambiguous Entities Mentioned from AIDA [Hoffart et al., 2011b]

$$coherence(p_1, p_2) = \begin{cases} 1 - relatedness(p_1, p_2) & \text{if } > 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.2)$$

$$relatedness(p_1, p_2) = \frac{\log(\max(|P_1|, |P_2|)) - \log(|P_1 \cap P_2|)}{\log(|P|) - \log(\min(|P_1|, |P_2|))} \quad (2.3)$$

TagMe

TagMe [Ferragina and Scaiella, 2010] is another well-known entity linking system. It is specifically designed for short text, such as social media comments, tweets, queries, etc., but is also frequently utilized for larger documents, such as news articles. It is extensively exploited by researchers as a large number of Web contents are smaller in size. For longer text, it is more effective than [Milne and Witten, 2008] and significantly more efficient than [Kulkarni et al., 2009]. TagMe follows the same scheme as for the above mentioned disambiguation systems and utilizes the anchor text mentioned in Wikipedia pages. The target sense (*i.e.*, Wikipedia page) of mentioned anchor a (*i.e.*, concept) in the given text is derived via the “collective agreement” of all other anchors b in the same text (\mathcal{A} is the set of all the mentioned anchors in the text and $b \in \mathcal{A} - a$). The score for each candidate sense w_a of anchor a is computed based on the voting from all other anchors b in the same text and their candidate sense w_b . If a mentioned anchor b is unambiguous, then relatedness between w_a and w_b defines the vote for w_a . On contrary, most related sense of w_a has an influence on the vote of w_a if b is ambiguous. TagMe utilizes the same *relatedness* score as proposed by [Milne and Witten, 2008], which is illustrated in Equation 2.3.

2.3 Supervised Learning and Classification Methods

Supervised learning is one of the main building blocks of the machine learning tasks. Applications in which the training instances consist of input vectors along with their target output labels are known as supervised learning tasks. It aims at learning a mapping function based on a given collection of input-output example pairs. Moreover, the trained

mapping function can be utilized to annotate unseen instances. In general, supervised learning task is formulated as follows:

Given a training set $T = \{(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \dots, (\mathbf{x}_N, t_N)\}$ of size N . (\mathbf{x}_j, t_j) represents the j^{th} example pair such as \mathbf{x}_j is the representative feature vector for j^{th} training instance and t_j corresponds to its target output label (*i.e.*, class label or some real number depending upon the nature of the task). When the target label of the training instances is a real number, it is called a regression task, whereas when the output label belongs to a class, it is called a classification task. A supervised learning algorithm aims at learning a mapping function $f : X \rightarrow Y$ from the training examples pairs, where X and Y belong to the input and the output space, respectively. The function which approximates the input space to the output space is called a hypothesis function. f is taken from the set of all possible hypothesis functions F (*i.e.*, $f \in F$), which is called the hypothesis space. The objective of the training operation is to minimize risk or a loss function in order to get a better generalization of a model over the training data.

In machine learning, the classification is a task of identifying the output label/class c from a set of known labels C for an unseen observation x , where $c \in C$. It decides the label based on the inherent patterns present within the seen observations. On the contrary, regression tasks deal with the real number as output based on previously observed instances. As the vital part of this thesis work, supervised learning and, specifically, the classification technique is more applicable and utilized extensively. In the following subsections, we discuss several classification algorithms which have been utilized to develop the classification models for the tasks studied as part of the thesis.

2.3.1 Random Forests

A random forest (RF) or random decision forest is an ensemble learning technique that exploits a collection of individual decision trees to make a better hypothesis over the training set [Breiman, 2001]. The intuition is that a committee's collection of relatively uncorrelated decision trees will perform better than any other individual decision tree. The random forest classifier integrates the bagging (also called bootstrap aggregation) ensemble technique and the random subspace method with random decision trees to better generalize and enhance stability (e.g., by reducing variance). A number of decision trees are allowed to contribute to the target label's decision for a test instance. The output for the regression task is the arithmetic mean of the values predicted by each of the individual trees, and in the case of a classification task, the output is determined by the majority vote technique. Each single decision tree model is highly prone to the issue of overfitting and, thus, results in a high variance problem. So, its performance is very vulnerable to unseen data due to poor generalization. To this end, the RF model handles the overfitting problem by compiling a set of decision trees. The key idea of creating the RF model is to apply sampling over the training data "with replacement" multiple times and create a decision tree for each individual sample. Each individual decision tree may overfit the sample training data on which it is built. However, via aggregating the decision of multiple trees, individual overfitting can be avoided. A unique decision tree can be trained using algorithms, such as CART [Leo Breiman, 1984], ID3 [Quinlan, 2004], C4.5 [Quinlan, 1993], etc. In order to create a decision tree, different algorithms

utilize different splitting criteria [Sheth and Deshpande, 2015] to optimize the separation in the sub-branches. Some of these splitting criteria include Gini impurity, Information gain, Variance reduction, Measure of “goodness”, etc. We utilize the Gini impurity criteria to get the best split within a decision tree to form the RF model. It measures the probability of a randomly selected sample being misclassified based on the distribution of samples at that node. Let C be the number of possible classes and p_i be the fraction of samples labeled with class i . Then, Gini impurity for a set of samples can be computed using Equation 2.4 and 2.5. At each node, the decision tree seeks for the features to split that results in the lowest Gini impurity value (*i.e.*, highest reduction in Gini impurity value). In the case of the RF model, the random subspace method governs the set of possible candidate features to be split.

$$Gini = \sum_{i=1}^C p_i^2 \quad (2.4)$$

$$Gini\ Impurity = 1 - Gini = 1 - \sum_{i=1}^C p_i^2 \quad (2.5)$$

The bagging and the random subspace methods supervise the creation of an ensemble of uncorrelated decision trees in RF based model, which are discussed as follows:

Bagging

Bagging, also called bootstrap aggregation, is an ensemble meta-algorithm. It is commonly designed to reduce variance and avoid overfitting. Bagging generates a random sample of instances from the training instances following the “with replacement” strategy (*i.e.*, an instance can be selected more than once). After generating multiple samples, multiple independent models are trained by utilizing each individual sample to form the basis of the ensemble model. Suppose n is specified as the number of estimates for the decision trees as part of creating the RF ensemble model and a training set of size N is specified as $T = \{(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \dots, (\mathbf{x}_N, t_N)\}$. The algorithm produces a separate training sample for each individual decision tree (*i.e.*, total n samples). The training set T_k for each decision tree is produced by doing uniform sampling “with replacement” from the actual training set T , where $T_k \in \{T_1, T_2, \dots, T_n\}$. The size of each sample training set is the same as the size of the original training set. Since the “with replacement” strategy is used for sampling, the training sets can include multiple copies of an instance from the actual training set. Different training instances associated with each decision tree maintain a lower order of correlation among themselves, and thus, an ensemble of these trees produces a better generalization.

Random Subspace Method

The idea of the random subspace method that aims to diminish the correlation among the estimated decision trees has been introduced by Ho [Ho, 1998]. It is also known by other terminology - feature bagging or attribute bagging. The random subspace technique differs from the basic bagging algorithm because it operates on the feature set while the former works on the training set. It follows the same strategy for sampling (*i.e.*, with replacement)

as used in the bagging algorithm, but on the feature set and utilizes a random sample of features instead of the entire feature set. It results in individual learners (*i.e.*, decision tree in our case) not emphasizing the highly discriminative/predictive features in the training set. These features might not be as discriminative/predictive for the unknown data point as for the training set.

2.3.2 Support Vector Machines

Support Vector Machines (SVMs) are another well-known supervised learning algorithm. The objective of an SVM is to identify a k -dimensional hyperplane (where k is the number of features) that precisely classifies the data instances [Cortes and Vapnik, 1995]. There can be many possible choices of selecting the hyperplane, which distinctly classifies the two sets of data instances. The objective is to identify the maximum margin hyperplane, *i.e.*, the hyperplane which has a maximum distance from the two sets of instances. The instances that impact the orientation and the position of the hyperplane are called the support vectors. An SVM is a kernel based technique, and its solutions are sparse. It means that a subset of training instances at which the kernel function is evaluated, decides the prediction of unseen instances.

As mentioned in the introduction of Section 2.3, $T = \{(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \dots, (\mathbf{x}_N, t_N)\}$ is a set of training instances of size N . (\mathbf{x}_j, t_j) represents the j^{th} example pair, such as \mathbf{x}_j is the representative feature vector for j^{th} training instance and t_j corresponds to its target output label. For a two-class classification problem, $t_j \in \{-1, 1\}$. Thus, a binary classifier problem can be derived using the linear models as in Equation 2.6. The sign of $f(\mathbf{x})$ decides the target label for some test instance \mathbf{x} . $\varphi(\mathbf{x})$ represents the feature space transformation, \mathbf{w} and b are the explicit parameters. \mathbf{w}^T represents the transposed vector of \mathbf{w} .

$$f(\mathbf{x}) = \mathbf{w}^T \varphi(\mathbf{x}) + b \quad (2.6)$$

Figure 2.3 illustrates a maximum margin hyperplane of an SVM classifier trained for a binary classification problem with two features. Data points falling on the margin of the hyperplane are called the support vectors.

If all the instances in the training set are linearly separable, each training instance should satisfy the condition mentioned in Equation 2.7. This is called the canonical representation of the decision hyperplane.

$$t_j(\mathbf{w}^T \varphi(\mathbf{x}_j) + b) \geq 1, j = 1, \dots, N. \quad (2.7)$$

In order to get the best generalization and separation of the two classes of training instances, we need to derive a maximum possible marginal decision hyperplane. Maximization of the margin distance provides more confidence about future instances. Therefore, there is a need to maximize the distance between the hyperplane and the margin boundaries. This turns out to be an optimization problem and equivalent to minimization of $\|\mathbf{w}\|^2$. Mathematically, it is formulated as Equation 2.8, subjected to constraints defined of Equation 2.7.

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \quad (2.8)$$

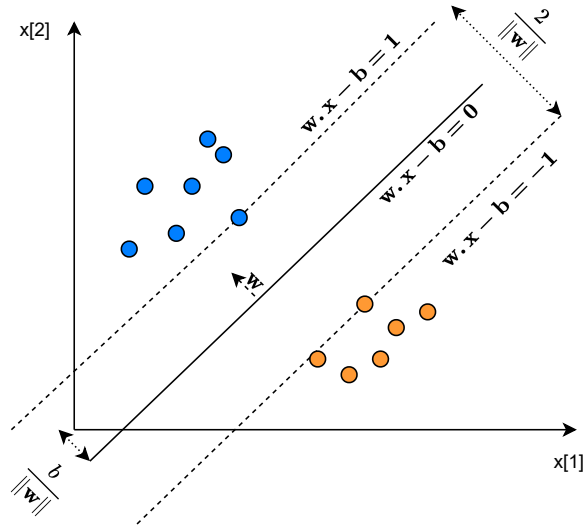


Figure 2.3: Binary Classification Problem SVM Hyperplane with Margin

In order to find a solution of the constraint optimization problem of Equation 2.8, we can convert the original problem into Equation 2.9 by utilizing Lagrange multipliers $a_j \geq 0$, with one multiplier a_j is defined for each constraint in Equation 2.7. To this end, we solve the following problem by optimizing with respect to \mathbf{w} , b , and Lagrangian multipliers \mathbf{a} .

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{j=1}^N a_j \{t_j (\mathbf{w}^T \varphi(\mathbf{x}_j) + b) - 1\} \quad (2.9)$$

where $\mathbf{a} = (a_1, a_2, \dots, a_N)^T$. Computing derivatives of $L(\mathbf{w}, b, \mathbf{a})$ with respect to \mathbf{w} and b and setting them equal to 0 provide us Equation 2.10 and 2.11.

$$\mathbf{w} = \sum_{j=1}^N a_j t_j \varphi(\mathbf{x}_j) \quad (2.10)$$

$$0 = \sum_{j=1}^N a_j t_j \quad (2.11)$$

Elimination of \mathbf{w} and b from $L(\mathbf{w}, b, \mathbf{a})$ in Equation 2.9 by utilizing these conditions provides a dual representation of maximum margin problem that needs to be maximized with respect to \mathbf{a} . The sign of $f(x)$ determines the class of a test instance, which is derived by substituting for \mathbf{w} from Equation 2.10 to Equation 2.6. It is formulated in Equation 2.12. $K(\mathbf{x}, \mathbf{x}_j)$ is called the kernel function and defined by $K(\mathbf{x}, \mathbf{x}_j) = \varphi(\mathbf{x})^T \varphi(\mathbf{x}_j)$.

$$f(\mathbf{x}) = \sum_{j=1}^N a_j t_j K(\mathbf{x}, \mathbf{x}_j) + b \quad (2.12)$$

2.3.3 Multi-label Classification

In contrast to the previously mentioned approaches where each instance is associated with only one label, multi-label classification is a variation where multiple labels may be assigned to each individual instance. With the continuous enhancement in available data, multi-label classification has become an omnipresent demand for many real world modern-day problems, such as music genre classification, protein function categorization, scene classification, etc., [Tsoumakas and Katakis, 2007]. A multi-label classification problem is handled in two ways: problem transformation and problem adaptation. The problem transformation approach converts a multi-label classification into several single-label binary classification problems, while the adaptation approach focuses on changing the loss/cost function of the binary classifiers (e.g., C4.5 decision tree). We utilize a one-against-all problem transformation technique [Rifkin and Klautau, 2004] as it is regarded more suitable for large scale classification problems [Tang et al., 2009]. The one-against-all transformation technique converts the multi-label problem into a set of binary classification problems. An individual learner is generated for each of the feasible target labels.

A test data point can have more than one associated target label $c \in C$ where C is the set of all feasible class labels. Suppose $\mathbf{x} \in \mathbb{R}^k$ represents the representative feature vector of dimension k for any test data. Then, we need to compute the target vector $f(\mathbf{x}) \in \mathbb{R}^{|C|}$ where $f(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_{|C|}(\mathbf{x})\}$. $f_c(\mathbf{x})$ determines the association of a test instance vector \mathbf{x} to a certain class $c \in C$. To this end, a binary classifier is developed for each individual class label. The output label of an unseen instance is determined by the collection of classifiers, which are $|C|$ in total. Each individual binary classifier learns to isolate the instances of one designated class label from rest of others. A decision function f for each class label is generated by transforming the training data such that data points belong to the designated class label are treated as positive examples. In contrast, all other data points are considered negative examples.

2.4 Deep Neural Networks

Neural network architectures (deep learning) have received attention because of their benefits towards multiple tasks in various domains [Dong et al., 2021]. These domains include, but are not limited to, computer vision, speech recognition, natural language understanding, medical domains, etc. Success behind deep learning can be credited to the rapid development of computing resources (e.g., GPU), availability of large amount of data, and the capability of deep neural networks to learn the inherent semantics from euclidean data, such as text, image, and video. Although deep neural networks efficiently extract the inherent semantics of euclidean data, multiple real world applications heavily depend on non-euclidean data (*i.e.*, the data which does not follow some underlying structure). The number of these applications is increasing day by day. Examples include but are not limited to the World Wide Web, knowledge graph, social networks, e-commerce, protein interaction networks, citation networks, etc. In recent years, researchers from diverse domains are showing interest in enhancing deep learning approaches for graph data [Wu et al., 2019]. Motivated by the earlier neural networks techniques, new gener-

alizations of neural networks approaches have been developed which directly deal with a graph structured data [Duvenaud et al., 2015, Li et al., 2016, Defferrard et al., 2016, Kipf and Welling, 2017, Veličković et al., 2018].

To this end, we provide a brief overview of one of the well-known graph neural networks architectures called “Graph Convolutional Networks” which has been utilized in solving the tasks studied in the scope of this thesis.

Graph Convolutional Networks

A Graph Convolutional Network (GCN) [Kipf and Welling, 2017] is a multilayer neural network that directly deals with the graph structure of the input and produces embedding vectors for each of the nodes based on the features of their immediate neighbors in the graph. Graph-level embedding or outputs can be generated using a pooling or readout technique [Wu et al., 2019]. The information flow is driven via the edges of the graph in each of the GCN layers. In other words, each node of the graph gets information from all its direct neighbors. Knowledge about the larger proximity is incorporated after stacking of numerous GCN layers. For example, when a node gets information from its direct neighbors in the second layer, the neighbors’ information already captures their direct neighborhood’s information. We can supervise the distance of information propagation by carefully selecting the number of GCN layers in the model.

Formally, assume $G = (V, \xi)$ is a graph, where V represents the set of nodes and ξ represents the set of edges. Every node is assumed to contain a self loop, i.e., there is an edge $(v, v) \in \xi$ defined for each of the nodes. Consider a matrix $X \in \mathbb{R}^{n \times d}$, which consists of features defined for each of the n nodes in the graph. Each row belongs to \mathbb{R}^d and defines a feature vector for an individual node of the graph with a dimension of d .

For a single-layer GCN, the updated d -dimensional node feature matrix $H^{(1)} \in \mathbb{R}^{n \times k}$ is calculated using following formula:

$$H^{(1)} = \psi(\tilde{A}XW_0) \quad (2.13)$$

where $W_0 \in \mathbb{R}^{d \times k}$ is the initial weight matrix, k is a hyperparameter, and ψ is a non linear activation function. If A is the adjacency matrix and D is the diagonal matrix of the graph, then $\tilde{A} = D^{-1/2}AD^{1/2}$ is called the normalized symmetric adjacency matrix. As mentioned before, one can integrate the wider neighborhood information by stacking several GCN layers:

$$H^{(l+1)} = \psi(\tilde{A}H^{(l)}W_l) \quad (2.14)$$

where l stands for the layer number, and $H^{(0)} = X$. Figure 2.4 illustrates the schematic evolution of a graph for multi-layer GCN. The graph-level classification task aims to predict label for the entire graph [Zhang et al., 2018, Ying et al., 2018b]. The end-to-end training for the task can be performed by the augmentation of the readout layer, and/or pooling layers followed by GCN layers. GCN layers are responsible for the high level node representation whereas readout layers generate the graph representation H_G by utilizing the node representation of each graph. The readout function ($R(\cdot)$) is defined in Equation 2.15. In practice, *sum/mean/max* pooling/readout layers receive more attention because of its effectiveness and simplicity. An end-to-end framework can be designed by the

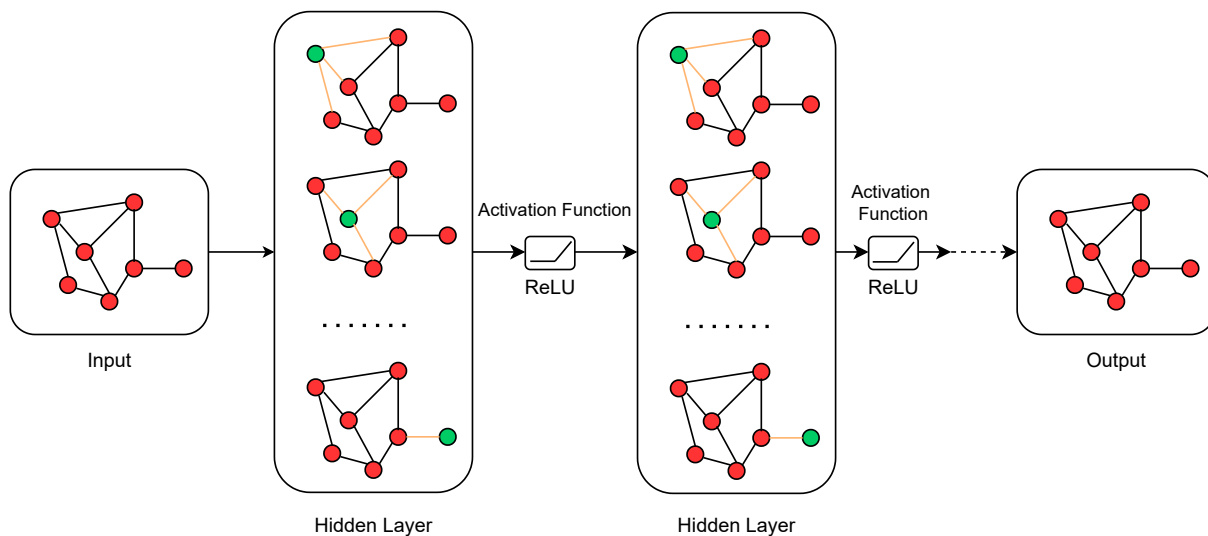


Figure 2.4: A Schematic Representation of Multi-layer GCN with First-order Filters

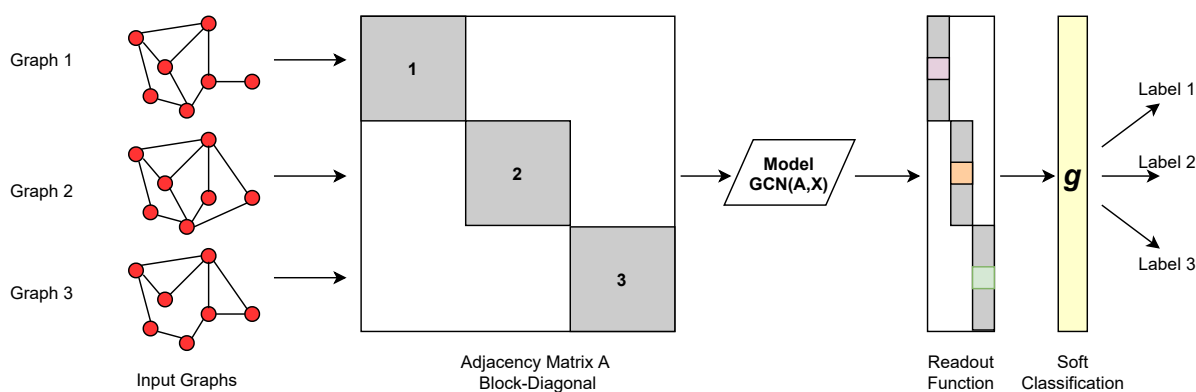


Figure 2.5: Schematic Depiction of GCN Batch-wise Graph Classification

addition of a softmax layer or multi-layer perceptrons followed by a softmax layer.

$$H_G = R(H_v^l, \forall v \in G) \quad (2.15)$$

Graph-level classification can be conducted in batch-wise learning over multiple graph instances. The size of these graphs can be potentially different from each other. A block diagonal adjacency matrix is created by exploiting the adjacency matrix of each graph. Each block of the newly created adjacency matrix stands for a single graph instance (cf. Figure 2.5).

2.5 Evaluation

Evaluation is a common standard of information retrieval (IR) systems. Generally, the quality of developed systems is assessed through the performed experiments and by analyzing it against a standard test set. In this section, we provide a brief overview of the

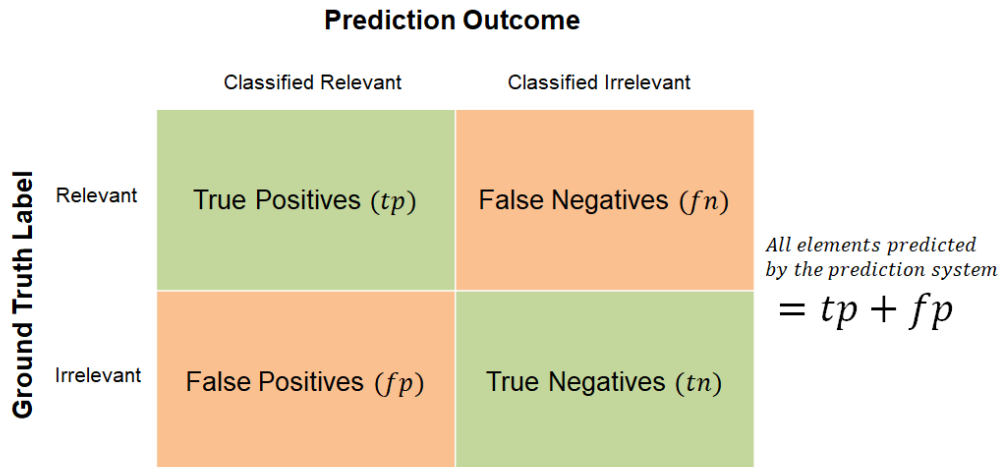


Figure 2.6: Confusion Matrix: Layout for Performance Visualization

most commonly used performance metrics, and which are suitable for the various tasks studied in the scope of this thesis.

A classifier/IR system aims at predicting/retrieving all the relevant labels/documents against a test/query instance. The most fundamental measures for evaluation of these tasks are Precision and Recall. To get a better insight into these fundamental measures, first we need to define the four sets of elements/items. These four sets are called true positive (tp), false positive (fp), true negative (tn), and false negative (fn). tp is the set of elements, which are correctly classified and are relevant. fp is the set of irrelevant elements, claimed relevant by the classifier/system. fn is the set of relevant items, that are incorrectly classified by the classifier/system. tn is the set of elements that are correctly classified as irrelevant. The above-mentioned sets are conceptually depicted in Figure 2.6. The evaluation matrix depicted in Figure 2.6 is called the “confusion matrix”.

- **Precision** measures the fraction of instances out of all the retrieved instances related to the query as formulated in Equation 2.16. Precision based on the “confusion matrix” is shown in Equation 2.17. Generally, Precision considers all the retrieved instances during the evaluation. But, it can also be evaluated based on the given threshold k by considering only the topmost responses retrieved. This measurement technique is called Precision@k.

$$precision = \frac{|\{relevant\ instances\} \cap \{retrieved\ instances\}|}{|\{retrieved\ instances\}|} \quad (2.16)$$

$$precision = \frac{tp}{tp + fp} \quad (2.17)$$

- **Recall** measures the fraction of relevant instances that are correctly retrieved. It is shown in Equation 2.18 and for “confusion matrix” in Equation 2.19, respectively.

$$recall = \frac{|\{\text{relevant instances}\} \cap \{\text{retrieved instances}\}|}{|\{\text{relevants instances}\}|} \quad (2.18)$$

$$recall = \frac{tp}{tp + fn} \quad (2.19)$$

- The **F-Score** constitutes a compromise between precision and recall measures. A general F-score is defined by Equation 2.20, where β represents a positive real value and recall is given β times more weightage than precision. The F-score with $\beta = 1$ is more common in practice and defined as the harmonic mean of precision and recall as shown in Equation 2.21.

$$F_\beta = (1 + \beta^2) * \frac{precision * recall}{\beta^2 * precision + recall} \quad (2.20)$$

$$F_1 = \frac{2 * precision * recall}{precision + recall} \quad (2.21)$$

Macro/Micro-Averaged Scores

As mentioned before, there exist several classification/prediction tasks in which more than one output labels can be assigned to the individual test instance. Macro and micro averaged based computations are proposed to handle such a scenario. The macro-averaged score treats each of the possible test classes/labels equally, whereas the micro-averaged based computation considers each of the test instances equally.

The ratio of collective true positives for all the classes to all the positive predictions can be defined as micro-precision score. In the same way, micro-recall can be defined as the ratio between collective true positives for all the classes and the actual positives labels. Suppose true positives for the j^{th} instance is denoted by tp_j , false positive by fp_j , and so on. $N_{test-set}$ represents the total number of test set instances. Then, micro-precision and micro-recall can be formulated as Equation 2.22 and 2.23, respectively.

Macro-averaged based computations are basically the arithmetic mean of the individual computation for each of the test instances as formulated in Equation 2.24 and 2.25 for macro-precision and macro-recall, respectively. The computation for the micro and the macro averaged F_1 -score is conducted by substituting precision and recall with macro-precision and macro-recall in Equation 2.21, respectively.

$$micro-precision = \frac{\sum_j tp_j}{\sum_j tp_j + \sum_j fp_j} \quad (2.22)$$

$$micro-recall = \frac{\sum_j tp_j}{\sum_j tp_j + \sum_j fn_j} \quad (2.23)$$

$$macro-precision = \frac{\sum_j precision_j}{N_{test-set}} \quad (2.24)$$

$$macro-recall = \frac{\sum_j recall_j}{N_{test-set}} \quad (2.25)$$

Mean Reciprocal Rank (MRR)

Mean Reciprocal Rank (MRR) is a static measure for the assessment of an information retrieval system which retrieves an index of possible responses with respect to a set of queries (cf. Equation 2.26). The reciprocal rank of a query is the multiplicative inverse of the rank of the first relevant response: 1 if a relevant response is retrieved at position 1, $1/2$ if relevant response is at position 2, $1/3$ for relevant response at position 3, and so on. MRR is the arithmetic mean of the reciprocal results for a set of Queries N , defined as:

$$MRR = \frac{1}{|N|} \sum_{i=1}^{|N|} \frac{1}{rank_i} \quad (2.26)$$

Chapter 3

Related Work

3.1	Entity Type Classification Methods	30
3.1.1	Fine-grained Entity Type Classification	30
3.1.2	Target Entity Type Classification	30
3.2	Linked Open Data and General Knowledge	31
3.2.1	Knowledge Based Models	32
3.2.2	Entity-level Analytics	33
3.3	Graph Convolutional Networks	34
3.4	Semantic Annotation & Search	36
3.4.1	Semantic Annotation	36
3.4.2	Semantic Search	38
3.5	User Pattern Analytics	39
3.5.1	User Profile Generation	39
3.5.2	Doppelgänger Detection	40
3.5.3	Authorship Attribution	42

This chapter provides an overview of related literature, which is relevant to our research problems highlighted in Chapter 1.

In more detail, we discuss prior works related to entity type classification in Section 3.1. These works are related to our study on the task of representative entity type classification (cf. Chapter 5). Section 3.2 provides the general techniques, such as knowledge-driven models and entity-level analytics which are related to all the studies performed throughout the thesis. Section 3.3 points out few applications of graph convolutional networks which is employed in solving the tasks of representative type classification (cf. Chapter 5) and user interest tracing (cf. Chapter 8). Next, we analyze the existing works related to semantic annotation of documents (cf. Chapter 6) and, subsequently, approaches for semantic search over Web documents (cf. Chapter 7). Finally, we review related studies regarding user behavior in Section 3.5, which is relevant to our contribution to the user interest tracing task (cf. Chapter 8). The detailed literature reviews are discussed in the following sections.

3.1 Entity Type Classification Methods

Type classification has been performed at several granularity levels. For example - entity type classification [Shimaoka et al., 2017] or document type classification [Govind et al., 2019a]. One of the main contributions of the thesis is to find the most representative type(s) corresponding to an entity (cf. Chapter 5). Thus, to this end, we segregate and review prior research related to fine-grained entity type classification and target entity type identification.

3.1.1 Fine-grained Entity Type Classification

Attributing the most suitable types(s) to an individual entity in a given text is a fundamental task and known as entity type classification. The most suitable type(s) are attributed from a set of fine-grained categories of types. We provide a brief overview of related works over type classification in Table 3.1.

FIGER, a fine-grained entity recognizer framework, predicts fine-grained entity type tags derived from Freebase for the entities mentioned in natural language texts [Ling and Weld, 2012]. The introduced framework utilizes various features, such as part-of-speech tags, contextual n-grams, distributional thesaurus, syntactical dependency features, etc., and a CRF sequential classifier for the prediction of tags. They further perform an extrinsic experiment on the relation extraction task to show FIGER’s capability.

HYENA is an entity label classification framework on a very fine-grained type taxonomy derived from the YAGO knowledge base [Yosef et al., 2012]. The proposed system exploits a multi-label hierarchical classifier along with a meta classifier to predict very fine-grained labels. It includes features, such as contextual, gazetteer, grammatical, etc.

Further, in [Shimaoka et al., 2017], the authors propose a model based on neural network architectures for the prediction of fine-grained type of an entity. They combine hand-crafted features (e.g., contextual, syntactical features, etc.) with a neural embedding for the proposed task and suggest that these features provide a balance in prediction. They also employ the attention mechanism. Their experiments on the FIGER (GOLD) and the OntoNotes datasets show the effectiveness of their approach.

Moreover, [Ghaddar and Langlais, 2018] introduce a large fine-grained entity types corpora based on English Wikipedia and derive tags from Freebase. They also validate their model on the OntoNotes and the FIGER (GOLD) datasets.

In a work, [Obeidat et al., 2019] introduce a zero-shot entity typing approach based on bi-directional LSTM and attention mechanism. The introduced framework exploits the entity-types distributional semantic representation derived from Wikipedia descriptions to assign a novel type emerged during the testing process without additional learning. Moreover, they also create a test set which incorporates more fine-grained entity types in comparison with the actual FIGER test data.

3.1.2 Target Entity Type Classification

The task of target entity type identification deals with discovering the most appropriate type for an entity. The appropriate type can be derived based on natural language text,

query, or relationship with other entities.

Balog and Neumayer [Balog and Neumayer, 2012] introduce the task of hierarchical target entity type identification based on a query. They propose two different models, one is type-centric and the other is entity-centric. They suggest that this task can be regarded as a learning to rank problem and propose several evaluation metrics. They also develop a dataset (query set and annotation types), which is publicly accessible.

The task of ranking entity types has also been addressed in [Tonon et al., 2013, Tonon et al., 2016]. The authors focus on ranking entity types based on the context by employing large collection statistics and relationship among different entities and types. To this end, they develop several models based on type hierarchy, context aware, and other entity related information. These type hierarchy approaches are based on the structural information among the types, while the context aware approaches exploit the other entities which co-occur with the current entity within the same text context. Finally, the entity related information approaches consider the relation between the current entity and the other entities. Further, this task has also been studied in the context of entity summarization by [Gunaratna et al., 2016].

In a study by [Garigliotti et al., 2019], the authors exploit hierarchical target entity type identification for solving the task of type-attentive entity retrieval. They derive type labels from a taxonomy and project the target entity type identification as a ranking task by employing random forest based regression algorithm as their supervised learning to rank approach. They exploit various entity-centric and type-centric features along with the features derived from a knowledge base and a given query.

Choi et al. [Choi et al., 2018] introduce a new entity typing scheme based on a collection of free form phrases and construct a new annotated dataset. This collection of phrases are ultra fine-grained in nature. They develop a model by employing attention based neural network architecture. The model integrates distant as well KB supervision by defining a multi-task objective function.

To this end, our task of representative entity type classification (cf. Chapter 5) has similarity to the target entity type identification task. The primary distinction is that we aim at finding the overall representative type for individual entities whereas the other task identifies the type with respect to a given context. Nevertheless, to find the representative type of an entity, the approaches in [Balog and Neumayer, 2012, Garigliotti et al., 2019] depend on a knowledge base (KB) as well as a query. In contrast, our approach solely relies on the type information of an entity from a KB.

3.2 Linked Open Data and General Knowledge

In this section, we focus on related studies that exploit structured and semi-structured knowledge through semantic enrichment from Linked Open Data (LOD) to enhance the performance of various tasks in natural language processing and information retrieval. In the following subsections, we also overview the related works that benefit from attaining semantic via entity-level analytics.

Reference	Approach	Features	Data	Remarks
[Ling and Weld, 2012]	CRF + Perceptron	Contextual + Syntactic + Distributional	Wikipedia + News reports	Multi-class + Multi-label
[Yosef et al., 2012]	Hierarchical + Meta classifiers	Contextual + Grammatical + Gazetteer	Wikipedia + FIGER (GOLD) + BBN	Multi-label + Meta classification
[Shimaoka et al., 2017]	Several neural encoders + Attention	Neural embedding + Hand-crafted	FIGER (GOLD) + OntoNotes	Hierarchical encoding
[Ghaddar and Langlais, 2018]	Neural network model	Mention + Context + Hand-crafted	Wikipedia + FIGER (GOLD) + OntoNotes	New corpus: WiFiNE
[Obeidat et al., 2019]	Bi-LSTM + Attention	Entity mention + Type + Wikipedia description	FIGER (GOLD)	Tackle novel type

Table 3.1: Comparative Study of Fine-grained Entity Type Classification

3.2.1 Knowledge Based Models

A plethora of tasks in information retrieval [Dalton et al., 2014] and natural language processing [Peters et al., 2019] incorporate information from knowledge bases (KBs) and openly accessible semi-structured knowledge such as Wikipedia²⁵. In the last decade, researchers from different domains have developed different knowledge acquisition techniques for the automatic construction and enhancement of knowledge graphs. This results in the advancement of many openly available KBs, such as DBpedia [Auer et al., 2007], Freebase [Bollacker et al., 2008], YAGO [Suchanek et al., 2007, Hoffart et al., 2013, Mahdisoltani et al., 2013, Pellissier Tanon et al., 2020]. Recently, a personal information management KB is introduced in [Montoya et al., 2018]. This KB integrates email messages, contacts, calendars, and location history information about a user. In addition, the large scale crowd sourcing efforts, such as Wikidata²⁶ have also been quite promising. Wikidata is the backbone of its Wikimedia sister projects. Wikipedia and Wikidata are both open sources of knowledge and cover more than 300 languages. The main characteristic of Wikidata is that data entered in any language is immediately available in other languages. These KBs provide temporal and spatial knowledge apart from the structural information. A plethora of tasks from different research domains have benefited by integrating this crucial world knowledge. In order to perform sentiment analysis on micro-blogging, [Hamdan et al., 2013] exploit the concepts extracted from DBpedia in combination with the feature extracted from WordNet and SentiWordnet. To this end, they develop an SVM and naive bayes models over the Sem Eval 2013 dataset. Some of other studies include question answering [Hao et al., 2018], disambiguation of named

²⁵ Wikipedia <https://www.wikipedia.org/>

²⁶ Wikidata <https://www.wikidata.org/>

entities [Usbeck et al., 2014], and social media topic identification [Cano et al., 2013].

Wide varieties of tasks related to information retrieval and natural language processing have accomplished improvement in the performance by exploiting ontologies [Benaouicha et al., 2015, Lytvyn et al., 2017]. Elberrichi et al. exploit WordNet concept for solving the task of content categorization [Elberrichi et al., 2008]. They extract generic concepts for each of the terms present in a text to derive a representative vector for the respective documents. A topic-specific ontology, Topic-OPA, is developed in [El Ghosh et al., 2020] for old press articles. Topic-OPA is built upon hierarchical and non-hierarchical schemes and is extracted from Wikidata by employing SPARQL based approach. In [Panigutti et al., 2020], the authors attempt to predict the next visit of the patient by exploiting the medical ontology and clinical history of a patient. They develop an agnostic model Doctor XAI, which can handle sequential, multi-labeled, and ontology based data. In order to provide semantics to the complex bimolecular network, [Ayadi et al., 2019] introduce the ontology, BNO. The authors validate the introduced ontology by employing different validation techniques, such as expert knowledge, automated consistency checking, etc. Moreover, their case study on Bacteriophage T4 G32 case shows the capability of the ontology. To recognize the human character, [El Bolock et al., 2020] propose an ontology CCOnto, which predicts users' behavior based on respective action in a given situation. This ontology is designed based on personality attributes, their elementary units & interaction among each other, and person reaction in different situations. In another study, [Kolbe et al., 2020] introduce an ontology ranking search dataset LOVBench to capture user's behavior and propose wide variations of features for ontology search.

3.2.2 Entity-level Analytics

The domain of entity-level analytics aims at enhancing semantic information by the assimilation of knowledge about an entity and/or its types. This has been widely accepted to be helpful in improving the performance of multitude of higher level tasks in Web science [Weikum et al., 2011], natural language processing [Sweeney and Padmanabhan, 2017], and information retrieval [Hong et al., 2011].

The incorporation of information about canonical entities has significantly enhanced the performance of prediction of event mentions in automatic content extraction [Hong et al., 2011]. They propose a model based on cross-entity reference, type information of entity mentions, and background knowledge about the entities to identify the events as well as role classification for the local entity. Their model is defined based on the intuition that similar types of events should involve similar entity types. Further, [Spitz et al., 2017] provide a Web-based interface and call it EVELIN, which is based on implicit networks of entities LOAD graph [Spitz and Gertz, 2016]. This system exploits the entities relation co-occurrence for event detection and summarization.

Issues related to Web archive data by employing longitudinal analytics are discussed in [Weikum et al., 2011]. In particular, they address the issue of entity tracking and detection in a Web page across the temporal scale to raise the whole analytics to the semantic level. In another work, [Ernst et al., 2016] provide an online demonstrator DeepLife which facilitates entity aware-search and exploration of health-related contents. This search interface is built on a large variety of resources, such as news articles, publications, etc.,

and integrates KBs like Knowlife and Unified Medical Language System.

The problem of event diffusion prediction into foreign language communities has shown encouraging results with the assimilation of knowledge about the entities contained in a document [Govind and Spaniol, 2017]. Here the introduced framework ELEVATE solely utilizes the information about the entities within a document and their associated location related data from YAGO. Experiments over multiple languages of Wikipedia communities demonstrate the viability of their approach. Subsequently, ELEVATE-live, a Web-based demonstrator raises the content of a Web news article to entity-level and visualizes its spread into the different geo-locations [Govind et al., 2018a]

The task of Web content fine-grained hierarchical classification is addressed in [Govind et al., 2018b, Govind et al., 2019a]. They hypothesize that a document is characterized by the named entities it contains. They propose the idea of the “semantic fingerprinting” method that expresses the overall semantics of a Web document by a compact vector. This compact vector is defined based on type information from YAGO for all the named entities contained within a Web document. In the end, it utilizes random forest and cosine similarity techniques to perform the fine-grained content classification.

[Duan et al., 2017] introduce a novel task of characterizing entity categories by generating histories of their named entities. They propose several unsupervised approaches by employing sentences, eras, topics, and topic correlation information and validate their approach on the Wikipedia category dataset.

Entity-level analytics is also effective in computational fact checking of information [Ciampaglia et al., 2015]. The authors claim that human fact checking can be achieved by finding the shortest path on a conceptually or semantically defined network, such as knowledge graphs (KGs). To validate their hypothesis, they exploit the knowledge graph consisting of RDF triples, which are originated from the facts of DBpedia.

To conclude, entity-level analytics always provides a depth insight into contents for a variety of tasks. Knowledge graphs carry a lot of information about entities, but all the information is not equally important for an ordinary user. The novelty of this thesis is to discriminate between relevant and irrelevant information for an average Web user by employing semantic enrichment from LOD and entity-level analytics.

3.3 Graph Convolutional Networks

In this thesis, we conceptually adapt the graph convolutional network (GCN) for solving the tasks of representative entity type(s) classification and user interest tracing. In this section, we give a brief overview of several applications which exploit GCN in order to address their issues.

Lately, the concept of Graph Neural Networks (GNNs) has been widely accepted by many researchers because it has demonstrated to be beneficial across several tasks in multiple domains [Wu et al., 2019]. In their seminal work [Kipf and Welling, 2017], the authors propose a simplified GNNs based model called Graph Convolutional Network (GCN), which accomplishes very encouraging results for several benchmark graph datasets.

A GCN model for text classification is introduced in [Yao et al., 2018]. They represent

an entire dataset by a heterogeneous graph. Documents and words are represented by one-hot encoding and act as the nodes of the graph. Edges are defined based on word appearance in the documents (document-word) and word co-occurrence (word-word) in the entire corpus. They also define different weights for different edges based on term frequency-inverse document frequency (Tf-Idf) and point-wise mutual information.

Moreover, [Bastings et al., 2017] incorporate the idea of GCN to address the issue of machine translation. They propose several models in combination with other encoders like RNN and CNN. The integrated GCN encoder extracts the characteristics of source sentences via syntactic dependency trees. They validate their efficacy of their methodology by performing English-Czech and English-German translation.

The task of cross-lingual knowledge alignment by employing GCN has been addressed in [Wang et al., 2018]. Their system train GCN to produce a unified vector space to embed all the entities of each language (or different knowledge graph). The graph is constructed based on entities relations and entities characteristics in a knowledge graph. These information tend to generate close embedding for equivalent entities of different knowledge graph.

In order to identify the relations in clinical narratives, [Li et al., 2019] propose the combination of GCN and recurrent neural network (RNN) and call it Seg-GCRNs. In Seg-GCRNs, GCN captures the syntactic dependency information whereas RNN captures the sequential information of a text. They validate their approach on the i2b2/VA clinical relation dataset.

The graph convolution is also adapted in order to solve one of the most important daily life issues “traffic forecasting” in [Cui et al., 2020]. They develop model based on graph convolution operation in combination with long short-term memory (LSTM) neural network. Graph convolution operation exploits the traffic network topology while LSTM deals with dynamic and spatial dependencies within the traffic data. Experiments on real-world traffic data validate the effectiveness of their proposed approach.

[Ying et al., 2018a] employ GCN in a very well-known task Web-scale recommendation. They introduce PinSage algorithm based on GCN and efficient random walk. This system is deployed on a very large Pinterest dataset. They also introduce new techniques for training which significantly improve the performance of recommendation system tasks.

Further, [Zhang et al., 2020] propose GraphRfi framework which exploits GCN and neural random forest (NRF) to tackle the problem of stable recommendation and, at the same time fraudster detection. GCN captures structural information from users’ rating graph and users’ behavior based on Amazon and Yelp review system to predict ratings while NRF performs the fraudster classification. They perform their experiments on Yelp reviews and movies & TV reviews crawled from Amazon.

[Wang et al., 2020] develop a novel personality recognition model based on GCN. A heterogeneous graph is constructed by the whole user text corpus to learn the embedding of users, words, and documents jointly. Users, words, and documents are initialized with one-hot encoding and act as the nodes of the large graph. They define three different types of edges: user-document edges, document-word edges, and word-word edges. Tf-Idf and point-wise mutual information are utilized to define the weight of the edges.

A bi-directional GCN model for rumor detection on social media is introduced in [Bian et al., 2020]. This model consists of top-down and bottom-up directed graphs. Top-down

directed graph captures the information about patterns of rumor propagation whereas bottom-up directed graph captures structure of rumor dispersion information within communities. The bi-directional GCN model also concatenates the source post feature at each GCN layer to enhance its significance at each layer. They evaluate the performance of their experiments on three well-known rumor datasets: Twitter15, Twitter16, and Weibo.

Sentiment classification is one of the most popular tasks in NLP domain. Recently, [Zhang et al., 2019] attempt to address the issue of aspect based sentiment analysis by exploiting the GCN architecture. GCN has been deployed over a sentence dependency tree to extract long range word dependencies and syntactical information. They develop a model based on GCN in combination with a bi-directional LSTM embedding. Enhance performances on Twitter and SemEval dataset show the effectiveness of their approach.

Nowadays, researchers from different domains are showing interest in GCN to solve their problems because of its adaptability to easily capture the structural information present within the dataset. [Huang et al., 2020] aim to solve the task of video question answering by exploiting GCN. Contents of the video is represented by a location aware graph. They also incorporate the attention mechanism to show the effectiveness on several video question answering dataset. GCN has also become popular in medical domain. [Parisot et al., 2018] utilizes GCN for brain and predict brain disorders based on imaging as well as non-imaging information. [Han et al., 2019] present a GCN and matrix factorization based method for the identification of disease-gene association.

Table 3.2 provides an overview of few applications of GCN. In the context of this thesis, we utilize GCN to solve the tasks of representative entity type classification (cf. Chapter 5) and user interest tracing (cf. Chapter 8). We propose several models based on GCN. In particular, we introduce the PURE framework for the task of representative entity type classification and the SUIT framework for user interest tracing.

3.4 Semantic Annotation & Search

In this section, we provide an overview of related studies relevant to semantic annotation of a document. Also, we present prior works over semantic search.

3.4.1 Semantic Annotation

Semantic Annotation is the task of tagging documents with related concepts. These concepts are based on metadata or content of the documents and can be derived from a knowledge graph or an ontology. This helps in digesting an un-structured text in a “healthy manner”. Semantic content annotation has been widely investigated in the digital libraries (DL), information retrieval (IR) and natural language processing (NLP) communities. An overview over approaches that are aligned along four key sub-tasks, *i.e.*, Named Entity Recognition, Relation Extraction, Entity Linking, and Ontology Development can be found in [Liao and Zhao, 2019]. However, these approaches are not suitable for a coherent semantic annotation of an entire document.

GoNTogle [Bikakis et al., 2010, Giannopoulos et al., 2010] generates semantic annotation of a document. It utilizes kNN text clustering and is strictly fixed to the ACM

Reference	Task	Approach	Data	Remarks
[Yao et al., 2018]	Text classification	GCN	20NG + Ohsumed corpus + R52 + R8 + Movie reviews	Heterogeneous graph based on entire corpus
[Bastings et al., 2017]	Machine translation	Bi-RNN + GCN, CNN + GCN, BoW + GCN	En-De + En-Cs news commentary (WMT16 translation task, v11)	Graph based on syntactic dependency trees
[Wang et al., 2018]	Cross-lingual knowledge alignment	GCN	DBP15K datasets	Graph based on entities relationships and characteristics
[Li et al., 2019]	Relationship classification in clinical texts	GCN + Bi-LSTM	2010 i2b2/VA relation dataset	Graph based on syntactic dependency relation
[Cui et al., 2020]	Traffic forecasting	Graph convolution + LSTM	INRIX based GPS data + Data collected from Greater Seattle area	Traffic network based graph
[Ying et al., 2018a]	Web-scale recommendation	GCN + Random walk	Pinterest dataset	Bipartite graph based on set of pins & set of boards
[Zhang et al., 2020]	Recommendation + Fraudster detection	GCN + NRF	Yelp reviews + Amazon ratings (Movies & TV)	Graph based on users' rating and users' behavior
[Wang et al., 2020]	Personality recognition	GCN	myPersonality + essays dataset	Graph based on user-document, document-word and word co-occurrence relations
[Bian et al., 2020]	Rumor detection	Bi-directional GCN	Weibo + Twitter15 + Twitter16	Graph based on rumor propagation
[Zhang et al., 2019]	Aspect based sentiment analysis (ABSA)	GCN + Bi-directional LSTM	Twitter + SemEval 2014, 2015, 2016 ABSA dataset	Graph based on sentence dependency tree

Table 3.2: Summarization of Few Applications of GCN

ontology classes. In [Afiontzi et al., 2013], the authors propose the annotation for the research papers presented in JCDL and ECDL based on Digital Library Evaluation Ontology. However, neither a demonstrator nor an API has been released. WebAnno [Eckart de Castilho et al., 2016] is a generic web-based annotation tool for distributed teams. As such, it supports semantic annotation tasks, but not document tagging. Open Calais²⁷ provides services for named entity recognition, instance recognition and facts for certain predefined properties with a focus on news contents. It is ontology-based and returns extraction results in RDF, however the coverage with links to other Linked Open Data sets is very limited. TagTheWeb [Medeiros et al., 2018] aims at identifying topics associated with documents. It relies on the knowledge expressed by the taxonomic structure of Wikipedia, based on the generation of a fingerprint through the semantic relation between nodes of the Wikipedia Category Graph. Semantator [Tao et al., 2013] is a Protégé [Musen, 2015] plug-in that attempts to convert biomedical text to linked data. In particular, it provides facilities for creating and removing ontology instances, managing instance relationships, and annotating relationships. NAISC [McKenna et al., 2019] is an interlinking approach for the library domain. In particular, it supports the creation of interlinks between entities, such as people, places, or works, stored in a library dataset to related entities held in another institution.

Recently, a semantic annotation system called CySem tagger has been developed for Welsh language to serve the semantic level analysis for the large Welsh data [Piao et al., 2018]. This system is designed based on the USAS tagger framework & initially developed on large scale Welsh semantic lexicons and compatible with multiple Welsh part-of-speech taggers. In another study, [Stork et al., 2019] develop a semantic annotation tool for natural and cultural history archival collection based on the documentation of their provenance. Additionally, their system also produces the structural annotation for the named entities present in these historical collections.

The before mentioned approaches offer only to a very limited extent support for digital curation. Apart from that, the approaches are usually application specific solutions and, thus, limited to a dedicated domain or extremely generic (*i.e.*, part-of-speech tagger or NER). In contrast, our approach (*i.e.*, semantic content tagging cf. Chapter 6) is generally applicable (no domain constraints), purely semantic (derived from the named entities contained in a document), and concise (focus on the most relevant type(s) derived from entity-level analytics), at the same time. Hence, additional contextualization becomes possible due to seamless linkage with data in the LOD cloud. Further, we also provide the annotation as a RDF file in order to allow a seamless linkage with the LOD cloud.

3.4.2 Semantic Search

Semantic search aims to enhance the search quality by apprehending the intent and contextual meaning of a search query keywords unlike the traditional lexical search which intends to search for the literal matches for the query terms [Guha et al., 2003].

GoNTogle [Bikakis et al., 2010, Giannopoulos et al., 2010] supports semantic and keyword-based search over documents. However, none of the systems is solely built upon

²⁷ <https://www.refinitiv.com/en/products/intelligent-tagging-text-analytics>

entity related information. STICS [Hoffart et al., 2014] aims at semantic annotation and retrieval via named entities, but does not exploit conceptual or structural similarity. Nordlys [Hasibi et al., 2017] is an entity-oriented and semantic search toolkit which provides the functionality of entity catalog, retrieval of entity, linking mention to an entity in a KB, and target type identification of an entity over a search query.

GYANI supports structured search over large collection of annotated document [Gupta and Berberich, 2019]. The system supports regular expression within queries and allows to add semantics to words via part-of-speech, temporal expressions etc. It utilizes different indexing unit (N-gram indexes, Annotation indexes, etc.) to facilitate retrieval of the related documents. Further, Qsearch framework [Ho et al., 2020] supports semantic search which can handle the quantities measure present within a search query by employing similar cues between query and source documents.

In a work by [Pfahler and Morik, 2020], the authors develop a GCN based model for semantic search over scientific publications. Here, they design a search query based on mathematical expressions. They compile a large collection of dataset from arXiv.org to train the model and create an evaluation dataset from several domains.

[Zhang et al., 2017] propose an unsupervised approach in order to search analogical objects in the different geographical locations by employing general-term and topic-biased transformations. They derive the experimental dataset from Wikipedia.

The aforementioned studies allow semantic search queries either to be in the form of free text/keyword, or entity-name. On the contrary, the proposed framework SEMANNOREX (cf. Chapter 7) is entirely semantic. It solely utilizes the entity type taxonomy (extracted from YAGO), built upon the most prominent entity types to design the search query. The SEMANNOREX framework exploits the knowledge captured in the type hierarchy for structured exploitation.

3.5 User Pattern Analytics

User pattern analytics aims at predicting user behavior based on activities with respect to Web contents, social media platforms, or ordinary documents. Generally, these activities can be derived from several operations, e.g., publishing, appending, or visiting Web contents by a user or social interactions of a user on digital media platform. It also deals with the task of finding multiple identities of the very same user on different online forums based on his/her activities. Detection of multiple identities is also known as Doppelgänger identification, sockpuppet detection, or alias matching. Many approaches have been proposed for user pattern analytics by employing supervised as well as unsupervised techniques [Eke et al., 2019]. Moreover, it has wide application in various domains [Stamatatos, 2009, Pennekamp et al., 2019]. To this end, we briefly provide an overview of related work in the subsequent subsections.

3.5.1 User Profile Generation

Generally, user profiles contain knowledge about a user. This knowledge can be derived from user preferences and interests, user behavior, or social activity. The authors intro-

duce WebDCC [Godoy and Amandi, 2006], a clustering algorithm for documents that performs incremental unsupervised learning over the Web documents in order to capture user profiles. The intermediary results such as semantic of Web pages can be incorporated in an ontology for the Semantic Web. [Ramanathan and Kapoor, 2009] create user profiles by projecting Web documents to Wikipedia concepts at different resolution levels, such as tokens, keywords, sentences, paragraphs, a summary of the document, and the whole document itself. At the same time, the advancement of social media platforms has shifted the interest of profile generation systems towards user interactions on these platforms. These systems exploit either topic modeling [Weng et al., 2010] or bag-of-words [Chen et al., 2010] approaches to create the user profiles.

[Ottoni et al., 2014] investigate the user activities across social media platforms Twitter and Pinterest by employing a novel approach which compares text-based content (Twitter) to image based content (Pinterest). This comparison is based on categories of content. Pinterest provides categories of image while categories of tweets are defined based on Pinterest categories, topic modeling, and crowd sourcing. Their study suggests that users are involved in a wide range of categories in Pinterest while categories in Twitter have better prediction power. Moreover, this study also suggests that although Twitter is extremely popular, online social platforms, such as Pinterest also play a vital role in new ideas and contents.

Further, the authors in [Han and Lee, 2016] project the social media contents into the corresponding categories of a news corpus. They estimate user interests by considering both the features of social media contents and news categories. They propose a refined topic modeling approach which captures the explicit as well as implicit terms corresponding to topics and distill the inevitable features of categories. They evaluate their approach on a Twitter dataset and develop an application which recommends social media friends to a user based on similar interests.

In a study by [Kang et al., 2018], CNN and a bidirectional gated recurrent unit (biGRU) are combined to predicate user interests in the social media platforms Twitter and Facebook. To this end, they exploit pre-trained word embeddings to give input to the biGRU encoder and construct a sentence matrix by employing the biGRU output and pre-trained word embedding which acts as the input for the CNN architecture.

3.5.2 Doppelgänger Detection

A Doppelgänger represents a double or an apparition of an alive person in fiction or folklore. Following this, several social forums refer a duplicate account of a user by the term Doppelgänger in the forum. The task of finding multiple accounts of a user is called “Doppelgänger detection” or “sockpuppet detection”. Current approaches to find Doppelgänger either depend on the stylometric features or metadata information of a user. The stylometric features represent the writing style characteristics of a user and include features, such as syntactic, lexical, domain-specific, etc., whereas metadata information includes active time stamps for a user, concerned topic for a user, etc. We provide a brief overview of various Doppelgänger detection studies in Table 3.3.

Metadata based Doppelgänger Detection

The most generic approach for Doppelgänger detection based on metadata information relies on the time stamps of the user generated contents. The authors in [Johansson et al., 2014, Johansson et al., 2015] propose the combination of different time specific features for Doppelgänger detection and call it “Timeprints”. They perform several experiments based on timeprints [Johansson et al., 2014] and subsequently, in combination with stylometric features [Johansson et al., 2015] show the significance of time specific features. They suggest that the time specific features can act as a very important tool for author identification. Further, the authors in [Fernquist et al., 2017] utilize time and event profiles in order to identify the alias of individual users based on cell phone data. Time profiles are created based on the time stamps of communication or data transfer between users, while event profiles are defined based on the locations, applications used, and detected Bluetooth devices by the user phones. The aliases are identified based on the cosine similarity computation.

In [Li et al., 2020], the authors make use of friendship networks for user identification. They develop a ground truth dataset crawled from Twitter, Facebook, and Foursquare sites for the very same users, including users’ display names and friendships networks. Further, feature analysis has been undertaken on the *k-hop* neighbors. These experiments show that the contribution of 1-hop neighbors are much higher in user identification than the other similarities. Several experiments have been performed on friendships based features and in combination with display name-based features.

Zheng et al. [Zheng et al., 2011] exploit the communication between the users, active time stamps of users and the topics they have published to detect the sockpuppets. They validate their approach on posts from popular Hong Kong discussion forums (Uwants and HKdiscuss) in the same as well as cross forums. [Maity et al., 2017] exploit tweet features (entropy of tweets and normalized retweet frequency) as well as profile features (e.g., friends count, followers count, location, description, verified profile, etc.) for sockpuppets detection by employing several supervised learning models.

In another study, [Zhou et al., 2019] propose a time series sockpuppet identification approach based on the dynamic growth of social network of an individual. The dynamic growth of a sockpuppet is computed by a weight representation method. They convert the sockpuppet identification into time series analysis problem and validate the effectiveness of their approach on publicly accessible data from Sina Weibo.

Stylometry based Doppelgänger Detection

Contrary to the approaches based on metadata, stylometric approaches are purely based on the user generated content and attempt to detect the text excerpt that was generated by the same user. Stylometric features can be defined as the information that can be directly retrieved from the published text and, in the ideal case, it should be unique for each of the respective users.

[Abbasi and Chen, 2008] propose a rich set of stylometric features and develop the “Writeprints” technique for the identification of user’s identity. Features of their approach include lexical, stylistics, structural, syntactic, domain-specific, and idiosyncratic facets of the generated text. The “writeprints” technique is based on a Karhunen-Loeve transformation which utilizes a sliding window and operates pattern disruption with individual

author level features. Current development of online communication augmented the rich set of stylometric features by including domain-specific aspects, such as utilization of emoticons [Cristani et al., 2012], amount of favorable votes [Mihaylov et al., 2015], and word sentiment [Cristani et al., 2012]. [Perifanos et al., 2018] utilize neural distributional embedding in order to capture the users' writing style.

The Doppelgänger Finder algorithm [Afroz et al., 2014] extracts the stylometric features and generates a score based on the similarity of the writing style for each author pair in an unsupervised setting. The authors perform experiments on blogs and underground forum datasets. [Pennekamp et al., 2019] propose an extension of the Doppelgänger Finder by employing additional features, such as idiosyncratic features and modern online communication attributes. Their idiosyncratic features include upper-case word in a sentence and a comment, white space in a sentence and a comment, frequency of grammatical errors, etc., to distinguish writing styles. They develop models for news comments in two languages - English and German.

[Chatzakou et al., 2020] combine wide range of features for identification of multiple identities of a user in the same social media platform. These features are defined based on user profile, user posting activity, different linguistic features extracted from the user's posted content as well as the social networks of a user. They perform several studies by employing machine learning and deep learning on Twitter abusive and terrorism datasets.

3.5.3 Authorship Attribution

The task of finding a candidate author for a text of unknown authorship is called authorship attribution or authorship identification [Stamatatos, 2009]. In a typical authorship attribution setting, a candidate author is selected from a given set of authors for whom attributed text samples are available. Authorship attribution has a variety of applications in various domains [Marouf and Hossian, 2019, Kalgutkar et al., 2019]. Table 3.4 provides a brief illustration of several authorship attribution studies.

[Schwartz et al., 2013] introduce the concept of author signature. The features which appear in at least $n\%$ of author training samples are defined as *n-signature* of the respective author. They utilize character n-grams and word n-grams to define the signatures of an author. They also employ a set of features based on flexible patterns, which represent the context in which functional words appear. They develop several SVM based models by varying the size of authors and training samples on Twitter dataset.

[Zhang et al., 2014] utilize word dependency relations, sentence grammatical voice, and non-subject stylistic words to represent the writing style of an author. Moreover, they develop an unsupervised technique to extract the features and represent semantic patterns of sentences in a uniform vector space. Their semantic association model exploits principal component analysis and linear discriminant analysis to identify the author of unstructured texts.

[Sousa Silva et al., 2011] introduce a set of personalized and idiosyncratic stylistic markers, such as emoticons, punctuation, abbreviations, interjections, etc. to train an SVM model for authorship attribution of Twitter messages. In [Cristani et al., 2012], the authors propose an additional set of features based on conversations (e.g., turn taking) for authorship attribution. Further, the authors in [Villar-Rodriguez et al., 2016] propose a

Reference	Objective	Approach	Features	Data
[Johansson et al., 2014]	Author identification + Alias matching	SVM + Naive Bayes + Manhattan distance	Time specific	ICWSM dataset
[Johansson et al., 2015]	Author identification + Alias matching	SVM + Naive Bayes + Cosine distance	Stylometric + Timeprints	ICWSM dataset
[Fernquist et al., 2017]	User identification	Cosine Similarity	Event profile + Time profile	Reality mining dataset [Eagle and (Sandy) Pentland, 2006]
[Li et al., 2020]	User identification	Several supervised learning models	Friendship networks based features	Foursquare, Facebook, and Twitter users
[Maity et al., 2017]	Sockpuppet detection	SVM, Logistic Regression, RF	Tweet + Profile	Twitter dataset
[Abbasi and Chen, 2008]	Identity identification	Karhunen–Loeve transformation	Stylometric	Email, instant messages, feedback comments, and program code
[Afroz et al., 2014]	Authorship attribution + Doppelgänger detection	Sequential minimal optimization + Probabilistic model	Stylometric + Language-specific	Underground forum dataset
[Pennekamp et al., 2019]	Doppelgänger detection	Probabilistic model	Stylometric + Online communication specific + Idiosyncratic	News comments dataset
[Chatzakou et al., 2020]	Doppelgänger detection	Bayes Net, J48, RF, GRU + Dense layer	Profile + Activity + Linguistic + Network	Abusive tweets [Chatzakou et al., 2017] + Terrorism tweets (Arabic)

Table 3.3: A Comparative Illustration of Various Doppelgänger Detection Studies

feature selection algorithm over the linguistic features extracted for short messages. This algorithm derives a separate set of features for each of the senders. In [Shrestha et al., 2017], the authors propose a character n-grams based CNN model for the authorship attribution task. They develop unigrams and bigrams based CNN models for a Twitter dataset. In a study by [Boenninghoff et al., 2019], the authors propose a novel attention based Siamese neural network model for authorship verification in their newly published amazon reviews dataset.

DeepStyle is an embedding based framework that learns the user’s writing style by employing a deep learning technique [Hu et al., 2020]. They propose multi-level CNN for user post representation and utilize triplet loss [Cheng et al., 2016] to learn the posts embeddings. Further, they exploit an aggregation function to merge the posts embeddings of the same user and to learn the user’s writing style. They evaluate their approach on the Weibo and Twitter datasets.

[Marouf and Hossian, 2019] employ an authorship attribution technique to identify the lyricist of Bangla songs. They define different types of stylometric features for Bangla songs and propose several supervised models. Furthermore, [Sarwar et al., 2020] propose a set of Thai stylometric features. They introduce a new corpus and employ a probabilistic *k-nearest neighbors* classifier for author identification in Thai language.

The above-mentioned approaches either depend on the metadata information of users or linguistics features extracted from the written text. So, these approaches are language and domain dependent. On the contrary, our approach (cf. Chapter 8) solely utilizes the concepts [categories] of the documents and exploits the inherent semantics among these concepts [categories] in order to derive the user interest patterns. Thus, it is independent of any language or domain. Therefore, our approach addresses a similar problem but is not directly comparable to the previously mentioned approaches.

Reference	Approach	Features	Data
[Sousa Silva et al., 2011]	SVM	Stylometric + Stylistic markers	Twitter messages (Portuguese)
[Cristani et al., 2012]	Cumulative match characteristic (CMC)	Stylometric + Conversational features	Dyadic chat conversation (Italian)
[Schwartz et al., 2013]	SVM	Author's unique signature + Flexible patterns	Twitter dataset
[Zhang et al., 2014]	PCA + LDA + 1-NN	Structural + Lexical + Syntactic + Semantic	English books [Koppel et al., 2007] + RCV1 corpus [Houvardas and Stamatatos, 2006]
[Shrestha et al., 2017]	CNN	Character n-gram based embeddings	Twitter dataset [Schwartz et al., 2013]
[Boenninghoff et al., 2019]	Attention based siamese neural networks	Neural embeddings	Short Amazon reviews
[Hu et al., 2020]	Multi-level CNN + Triplet loss	Word + Character + Bi-gram + POS tag based latent representation	Weibo + Twitter datasets [Schwartz et al., 2013]

Table 3.4: A Comparative Illustration of Various Authorship Attribution Studies

Chapter 4

Semantic Analysis & Digestion

4.1	Overview on CALVADOS	48
4.1.1	Document Preprocessing	48
4.1.2	Semantic Fingerprinting	49
4.1.3	Semantic Exploration	49
4.2	CALVADOS Interface	49
4.2.1	Content Digestion via Semantic Distillation	49
4.2.2	Comparison of Documents Semantics	50
4.3	Findings on Digestion of Semantic Content	51

In order to get the initial contextualization of Web documents, we analyze them semantically and reveal the main topics and the semantic building blocks associated with the corresponding document. We start with addressing our first research question (cf. RQ1 in Chapter 1) - “How to explore a Web content with respect to named entities mentioned in it?” in this chapter. To this end, we present the CALVADOS framework for semantic analysis and visualization of Web documents by employing **semantic fingerprinting** [Govind et al., 2018b].

Even after celebrating the Web’s 30th anniversary in 2019, we still observe a gigantic growth in Web contents being created and, at the same time, being available for consumption. This novel data source is a blessing and curse at the same time. On the one hand side, we benefit from a vast amount of information accessible 24/7 all over the planet. On the other hand side, we might be overwhelmed by the sheer amount of data. To this end, efficient and smart approaches are required, in order to help us to “digest” this huge quantity of data in a “healthy manner”. Our hypothesis - therefore - is, that the named entities contained in a Web content carry its inherent semantics (cf. Hypothesis 1 in Chapter 1). In order to do so, we combine named entity disambiguation (e.g., AIDA [Yosef et al., 2011] or DBpedia Spotlight [Mendes et al., 2011]) with freely available knowledge bases (KBs) such as DBpedia [Auer et al., 2007] or YAGO [Hoffart et al., 2013]. As a result, **semantic fingerprinting** [Govind et al., 2018b, Govind et al., 2019a] has been previously introduced as a general purpose approach towards Web content classification.

In this chapter, we introduce the CALVADOS (Content AnaLytics ViA Digestion Of Semantics) system as an extension of **semantic fingerprinting**. CALVADOS is a novel approach that aims at distilling and visualizing semantics of documents by exploiting entity-level analytics for a user-friendly “digestion”. To this end, our work makes the following salient contributions:

- use of **semantic fingerprinting** to capture content’s (inherent) semantics
- visualization & exploration of (inter-) dependencies among entities contained
- provisioning of contextual KB data (e.g., types) supporting data digestion

4.1 Overview on CALVADOS

The goal of CALVADOS is to digest the semantics of a Web content and provide visualizations to facilitate content consumption. The backbone of the CALVADOS system is **semantic fingerprinting**, a fine-grained type classification approach based on the hypothesis that “*You know a document by the named entities it contains*”. The general approach is briefly explained in this section, more details can be found in [Govind et al., 2018b, Govind et al., 2019a]. In short, the semantics of a document is captured by the use of a **semantic fingerprint**, i.e., a vector that encodes the core semantics of the document based on the type information of entities contained. The ambiguity among named entities is handled using aforementioned standard NERD systems. The actual fine-grained type prediction via **semantic fingerprinting** can be described in the following two steps. First, the document’s **semantic fingerprint** is computed. For this purpose, we perform entity-level analytics of the entities contained and, in particular, exploit the type information from the knowledge base YAGO. Second, we employ a random forest classifier to predict the top-level type of our prediction. Once identified, the system aims to find the most suitable fine-grained sub-type. In order to do so, the cosine similarity between the **fingerprint** of the document and the representative vectors of the sub-types are computed, and the one with the highest score being selected. For example, an article about some football game can be predicted as an **event** in the top-level prediction, and further aligned to the more specific type **game** in the second step. CALVADOS utilizes the aforementioned **semantic fingerprints** to semantically analyze and digest Web contents.

The overall pipeline of CALVADOS works in three stages as depicted in Figure 4.1. We discuss each of the steps in detail, in the subsequent subsections.

4.1.1 Document Preprocessing

In the first stage, CALVADOS monitors the feeds of Web documents by a user. Preprocessing of the Web documents is performed to remove the HTML markup (Hypertext Markup Language) and any unwanted noise, such as online advertisements or unrelated contents on the page. We exploit the boilerplate removal technique to obtain the clean and relevant text.

Subsequently, we extract the named entities present in a Web document by employing DBpedia Spotlight [Mendes et al., 2011]. Then, we create a list of all the extracted canonical entities. By doing this, we raise the content of each document to entity-level.

4.1.2 Semantic Fingerprinting

The following stage then involves the computation of the `semantic fingerprint` for the concerned document, as discussed above. Subsequently, the relevant fine-grained types for the document are predicted from an entity type hierarchy. Here, it is worth mentioning that `semantic fingerprint` only exploits the type(s) information of named entities mentioned in a Web document, and thus, it is domain agnostic.

4.1.3 Semantic Exploration

In the final stage, the `semantic fingerprint` and predictions generated in previous stages are explored and visualized to serve the overall goal of simplified content digestion based on a semantic distillation. To this end, we provide a Web interface for the overall exploration. Visualizations are constructed by utilizing the JavaScript library Data-Driven Documents²⁸ (D3.js).

4.2 CALVADOS Interface

The goal of CALVADOS is to help users in digesting documents via entity-level analytics. To this end, entity information are extracted and visualized. In particular, various interactive visualizations are provided:

- the `semantic fingerprint` of a document
- the tag cloud based on the named entities contained
- statistics about similarity with other types showcasing the document’s “flavor”
- an opportunity to compare two documents based on their inherent semantics

As such, this work comprises two use cases of the CALVADOS system (<https://calvados.greyc.fr/> for an online interface). The first use case facilitates content digestion of an individual document (cf. Subsection 4.2.1). The second use case allows users to compare the semantics of two different documents (cf. Subsection 4.2.2).

4.2.1 Content Digestion via Semantic Distillation

The first use case of CALVADOS is content digestion via semantic distillation. The user can input the content by providing a reference URL to the document or by uploading the raw text itself. CALVADOS performs the entity-level analytics on the document via

²⁸ <https://d3js.org/>

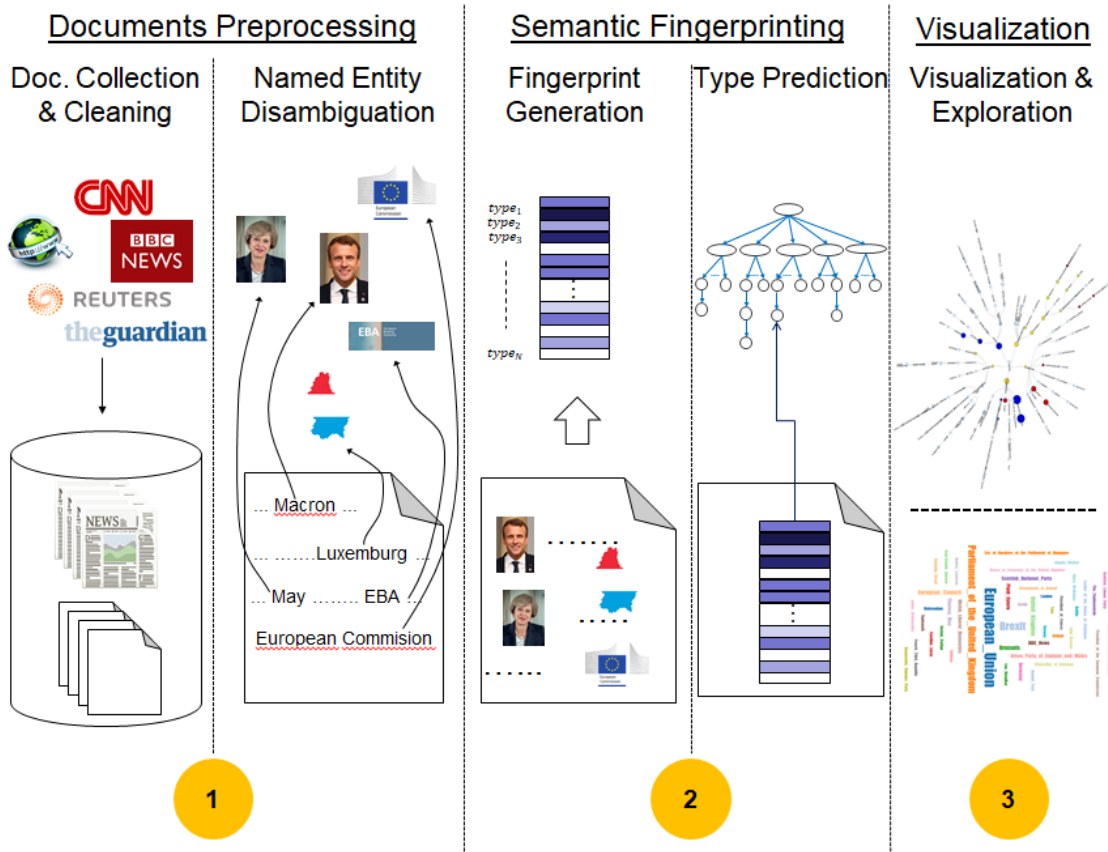


Figure 4.1: Conceptual Overview of the CALVADOS Pipeline

semantic fingerprinting. As a result, the system offers various visualization in order to provide a user-friendly content consumption. Focal point here is the visualization of the **semantic fingerprint** depicting the associated types based on the underlying type hierarchy. This graphical metaphor allows the user to understand the document’s constituents on a semantic level. For example, a news article about Theresa May²⁹ comprises a combination of various political parties, administrative districts, skilled workers, etc. Further, CALVADOS also provide an “entity cloud” based on the named entities contained, and highlights those types that are conceptually similar on the entity-level. Figure 4.2 displays a screenshot of the previously mentioned news article in CALVADOS.

4.2.2 Comparison of Documents Semantics

The second use case of CALVADOS is a semantic document comparison. To this end, we enable the end user to analyze the overlap and differences between two documents at the semantic level. This is achieved by visualizing the **semantic fingerprints** of both documents simultaneously as “overlay”. Further, an entity cloud on the intersecting named entities is provided. Here, it can be easily observed, that the **semantic fingerprints**

²⁹ <https://www.bbc.com/news/uk-politics-47627744>

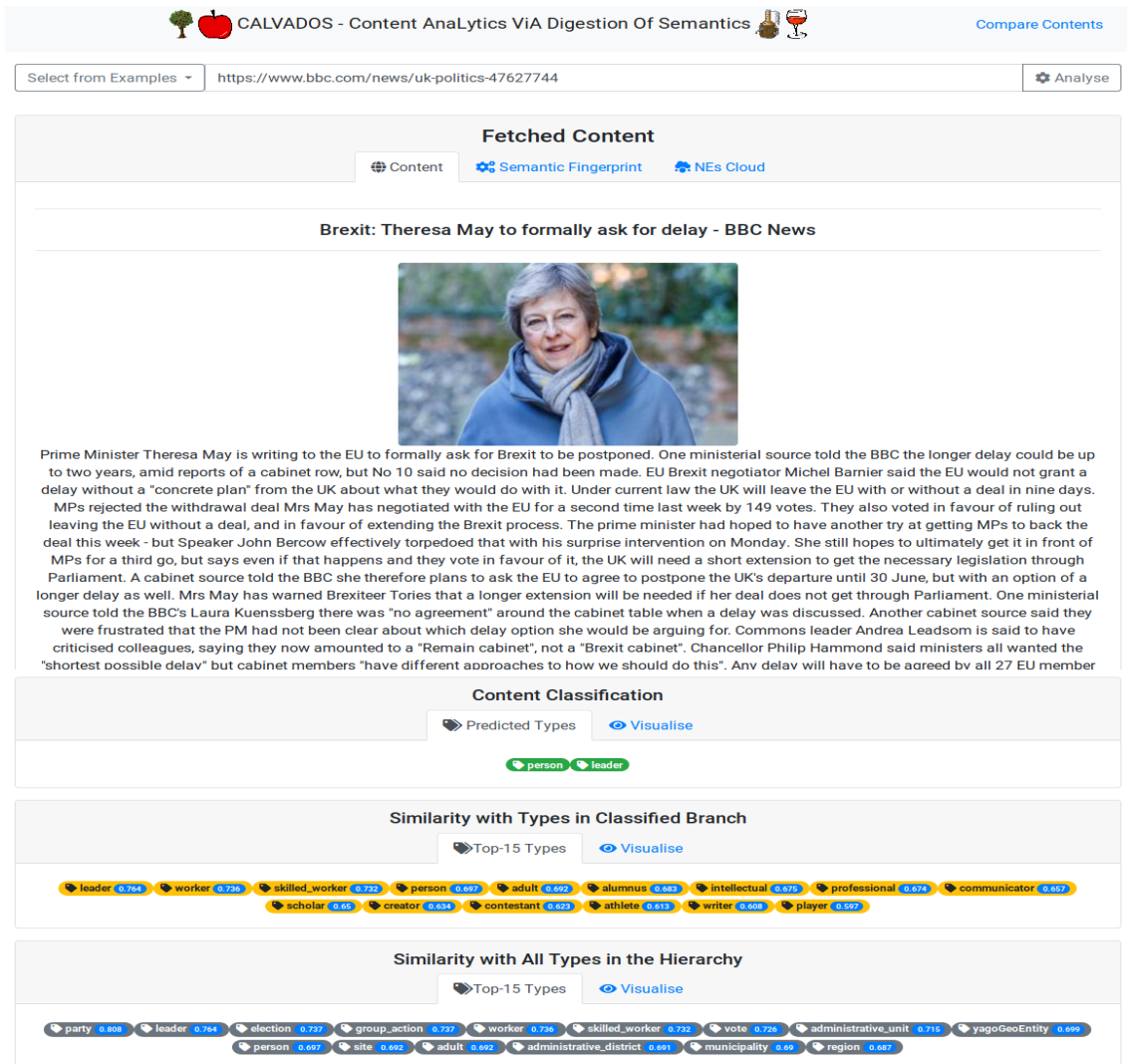


Figure 4.2: Semantic Digest of a Web Page

provide more insights in contrast to the plain entity mentions. In addition, information about the most similar types associated with both documents are provided in order to disclose their overall “flavor”. For example, when comparing the previous news article about Theresa May with a Manchester City FC article, there are visible differences. The former article being aligned towards political parties, skilled workers, etc., whereas the later one towards contests, clubs, etc. (cf. Figure 4.3). Finally, the quantified value of similarity based on the semantic fingerprints is indicated, as well.

4.3 Findings on Digestion of Semantic Content

In this chapter, we have introduced the CALVADOS framework, a Web-based tool for the semantic analysis and digestion of Web contents. CALVADOS lifts document analysis

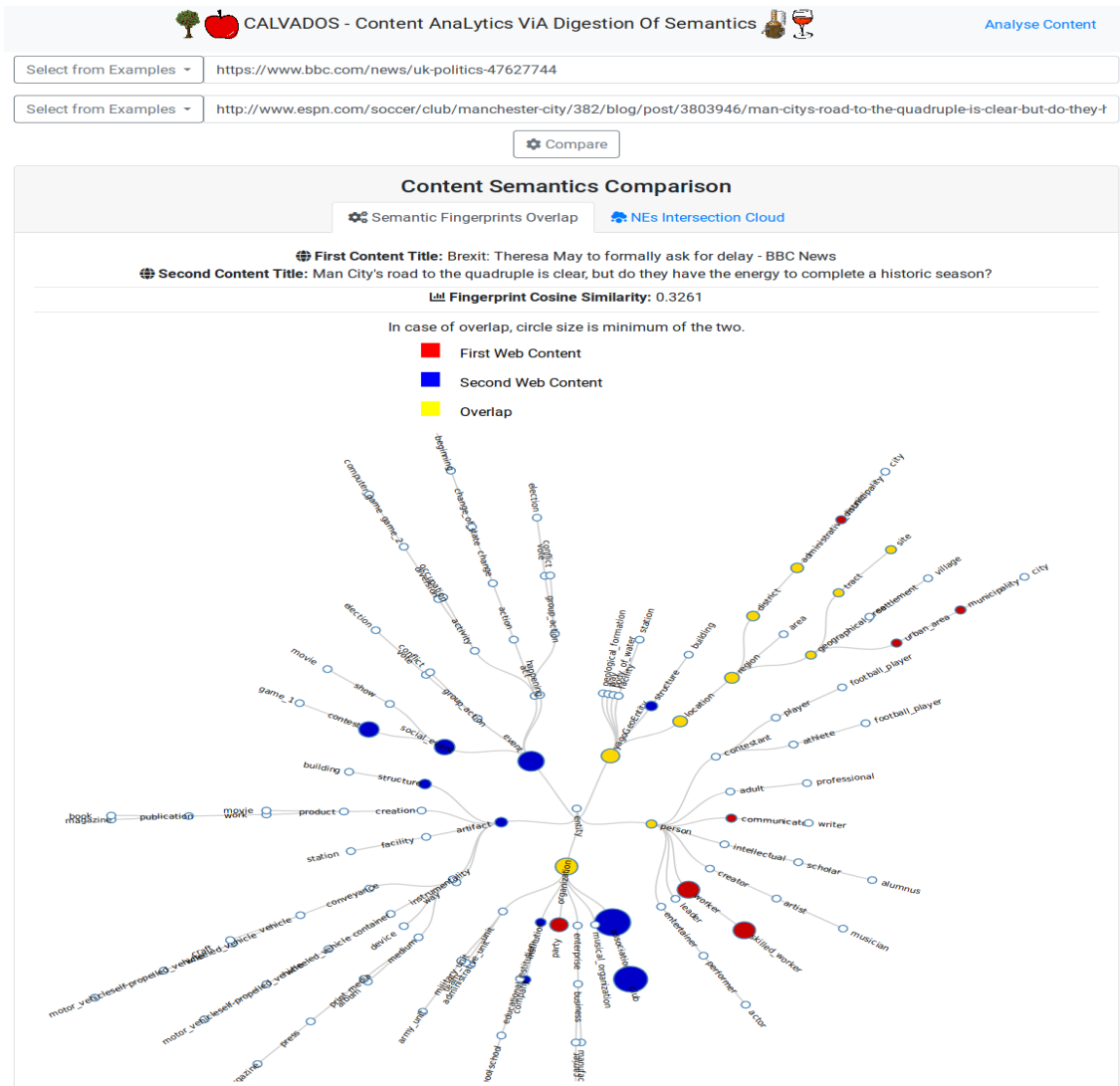


Figure 4.3: Semantic Comparison of Web Pages

to the entity-level and utilizes **semantic fingerprints** in order to capture the inherent semantics of a Web content. We observe that our system successfully captures and subsequently, provides the semantic of a Web content by employing named entities types information within it, which confirms our hypothesis (cf. Hypothesis 1 in Chapter 1). We enable end users to explore and visualize an individual Web document as well as comparison of two Web documents based on entity-level semantics. However, we observe that sometimes too much knowledge about entities can cause information overflow for a user, in particular with respect to the plentitude of types associated with named entities. To this end, there is a need of concise knowledge extraction about named entities in order to avoid dilution.

Chapter 5

Representative Entity Type Classification

5.1	Conceptual Approach	54
5.2	Computational Model	55
5.3	Representative Type(s) Classification	57
5.3.1	Baseline Models	58
5.3.2	Entity Type(s) Classification	58
5.3.3	Random Forest Model	59
5.3.4	Graph Convolutional Network Model	60
5.3.5	PURE	62
5.4	Experimental Evaluation	62
5.4.1	Experimental Setup	62
5.4.2	Model Configurations	65
5.4.3	Classification Results	65
5.5	Findings on Representative Type Classification	67

As highlighted in Chapter 1 and 4, information overload endangers the contextualization of a Web document, and concise type information about named entities might provide a deeper and more focused insight about the Web content. We believe that the relevant type(s) associated with an entity might help in getting the better contextualization of the Web contents. This leads us to address our second research question (cf. RQ2 in Chapter 1) - “Which facet(s) are the most expressive and representative for a named entity?” in this chapter. We, therefore, pursue the task of representative type classification in order to find the most relevant type(s) associated with a named entity. To this end, we introduce an approach towards Pattern Utilization for Representative Entity type classification called *PURE*.

As discussed in Chapter 1, the advent of the digital society has been driven by knowledge extraction and data extraction of “digital born” as well as digitized contents. Three

decades of the Web have further led to strategies of aggregating and cross-linking information from various knowledge sources. In particular, the maturation of high-quality knowledge bases (KBs) such as DBpedia [Auer et al., 2007] or YAGO [Suchanek et al., 2007, Hoffart et al., 2013] allows a semantic enrichment of contents. As such, literally, the “sky is the limit”. With the abundance of information available via the linked open data (LOD) cloud, there appears to be almost no limit in gathering, aggregating and presenting this data to users. Although a plenitude of information is - in general - desirable, the sheer amount of data might simply overwhelm the user, which becomes particularly relevant when tapping into the wealth of KBs and exploiting the rich information distilled about named entities. However, not each and every piece of information is equally important, given the type structure of KBs containing types such as PERSON, PRESIDENT or BLOGGER. While some types tend to be highly representative (e.g., PRESIDENT) with respect to a named entity, others are not (for instance, BLOGGER). To this end, we aim at structurally identifying the most representative type(s) associated with a named entity for a focused semantic exploitation supporting, e.g., a subsequent content enrichment or data curation to name just a few use-cases.

5.1 Conceptual Approach

KBs capture a wealth of information about each and every entity contained. For instance, YAGO’s transitive closure of entity types associated with *Donald Trump* contains 74 types, while there are 32 for the *Tower of London*. Undoubtedly, all of them are valuable and show the many facets of a named entity. In fact, the type graphs of named entities are extremely sophisticated. Due to properties such as the “subClassOf relation” the transitive closure of a named entity might become quite abstract. This might lead to dilution by too “generic” types, such as HEAD or LEADER, which are types in the upper-part of KBs. However, for a concise context exploitation, e.g., in the context of semantic fingerprinting [Govind et al., 2018b, Govind et al., 2019a], a focus on the most representative type(s) is essential in order to avoid dilution. Thus, the question arises: what are “representative” type(s) of a named entity? We postulate that named entities of certain types, e.g., PRESIDENT, share a multitude of common and (at the same time) characteristic types, such as POLITICIAN, BUSINESSPERSON and ALUMNUS (cf. Hypothesis 2 in chapter 1).

To this end, we employ a graph convolutional network (GCN) by adapting it to the characteristics of knowledge base type graphs in order to identify the predominant type patterns among all (sub-)types of a type hierarchy from YAGO composed of 5 top-level types (PERSON, ORGANIZATION, EVENT, ARTIFACT, and GEOENTITY). In particular, we perform our study on 299 different Types structured by the 5 before mentioned top-level types. Extensive experiments with named entities in Wikipedia demonstrate the viability of our approach and show a significant improvement in the performance of representative type(s) classification compared with the state-of-the-art competitors.

In the proposed work, we introduce *PURE* (Pattern Utilization for Representative Entity type classification). *PURE* aims at exploiting solely structural patterns derived from knowledge graphs in order to “purify” the most representative type(s) associated

with a named entity. In summary, the salient contributions of the current study are:

- a model for representative type classification;
- the creation of a gold standard of representative types for the most prominent named entities in Wikipedia;
- the adaptation of a *GCN* in order to structurally identify “predominant type patterns” of named entities;
- a comprehensive experimental study on identifying the most relevant entity types demonstrating the viability and the high quality of our method.

5.2 Computational Model

We consider a knowledge base (KB) as a pair $(\mathcal{T}, \mathcal{A})$ where \mathcal{T} is a set of terminological axioms (TBox) and \mathcal{A} is a set of assertional axioms (ABox). In our context, the TBox is an entity type hierarchy composed of entity types and `subClassOf` relations. Entity types are represented by τ as shown in Equation 5.1. τ_h ($1 \leq h \leq H$) defines the top-level types of the hierarchy and t_i ($1 \leq i \leq I$) defines all the successors, i.e., the subclasses, of all the top-level types in the hierarchy. All the types present in τ form a directed acyclic graph (DAG). Formally, the TBox \mathcal{T} can be defined as a pair $(\tau, R \subseteq \tau \times \tau)$, where τ represents the set of nodes. If t_i is a superclass of t_k , then there is a directed edge between t_i and t_k defined by the relation R . The ABox \mathcal{A} contains named entities and their associated types. It is a pair $(\mathbb{E}, P \subseteq \mathbb{E} \times \tau)$, where \mathbb{E} defines a set of named entities (cf. Equation 5.2) and P the type relation expressed between an entity and its type(s). The entity types directly associated with a named entity e_j in the ABox \mathcal{A} are represented by τ^{e_j} , as shown in Equation 5.3. Let $\hat{\tau}^{e_j}$ denote all the direct entity types along with their transitive closure as shown in Equation 5.4. All the predecessors, i.e., the superclasses, of a type are defined as the transitive closure of this type. For any entity e_j , we project $\hat{\tau}^{e_j}$ onto the sub-hierarchy of its top-level type τ_m . In case of multiple top-level types, we consider several projections, one for each top-level type. This projection is represented by $\Pi_{\tau_m}(\hat{\tau}^{e_j})$ and shown in Equation 5.5. All the types associated with $\Pi_{\tau_m}(\hat{\tau}^{e_j})$ form the directed acyclic graph $\tau_m^{e_j}$ (cf. Equation 5.6), where the set of nodes is $\Pi_{\tau_m}(\hat{\tau}^{e_j})$.

$$\tau = \{\tau_1, \tau_2, \dots, \tau_H, t_1, t_2, \dots, t_I\} \quad (5.1)$$

$$\mathbb{E} = \{e_1, e_2, \dots, e_J\} \quad (5.2)$$

$$\tau^{e_j} = \{\tau_1^{e_j}, \tau_2^{e_j}, \dots, \tau_h^{e_j}, t_1^{e_j}, t_2^{e_j}, \dots, t_i^{e_j} \mid h \in [1, H], i \in [1, I]\} \quad (5.3)$$

$$\hat{\tau}^{e_j} = \{\tau_1^{e_j}, \dots, \tau_h^{e_j}, \tau_{h+1}^{e_j}, \dots, \tau_y^{e_j}, t_1^{e_j}, \dots, t_i^{e_j}, t_{i+1}^{e_j}, \dots, t_z^{e_j} \mid y \in [h, H], z \in [i, I]\} \quad (5.4)$$

$$\Pi_{\tau_m}(\hat{\tau}^{e_j}) = \{\tau_m, t_1^{e_j}, t_2^{e_j}, \dots, t_n^{e_j} \mid t_k^{e_j} \text{ subClassOf } \tau_m, 1 \leq k \leq n\} \quad (5.5)$$

$$\tau_m^{e_j} = \{(\Pi_{\tau_m}(\hat{\tau}^{e_j}), r \subseteq \Pi_{\tau_m}(\hat{\tau}^{e_j}) \times \Pi_{\tau_m}(\hat{\tau}^{e_j})) \mid r \subseteq R\} \quad (5.6)$$

Entity Type Graphs

After projecting $\hat{\tau}^{e_j}$ onto one of the top-level types τ_m , we get the entity type graph $\tau_m^{e_j}$ for an entity e_j . This DAG consists of entity types associated with only one top-level type. The graphic in Figure 5.1 highlights an excerpt of the entity type graph for *Donald Trump*. The entity type graph of *Donald Trump* consists of 74 types in total, out of which are 28 leaf types. As shown, the top-level type is PERSON and leaf types include, but are not limited to, e.g., PRESIDENT, HOTELIER, OFFICEHOLDER and BLOGGER.

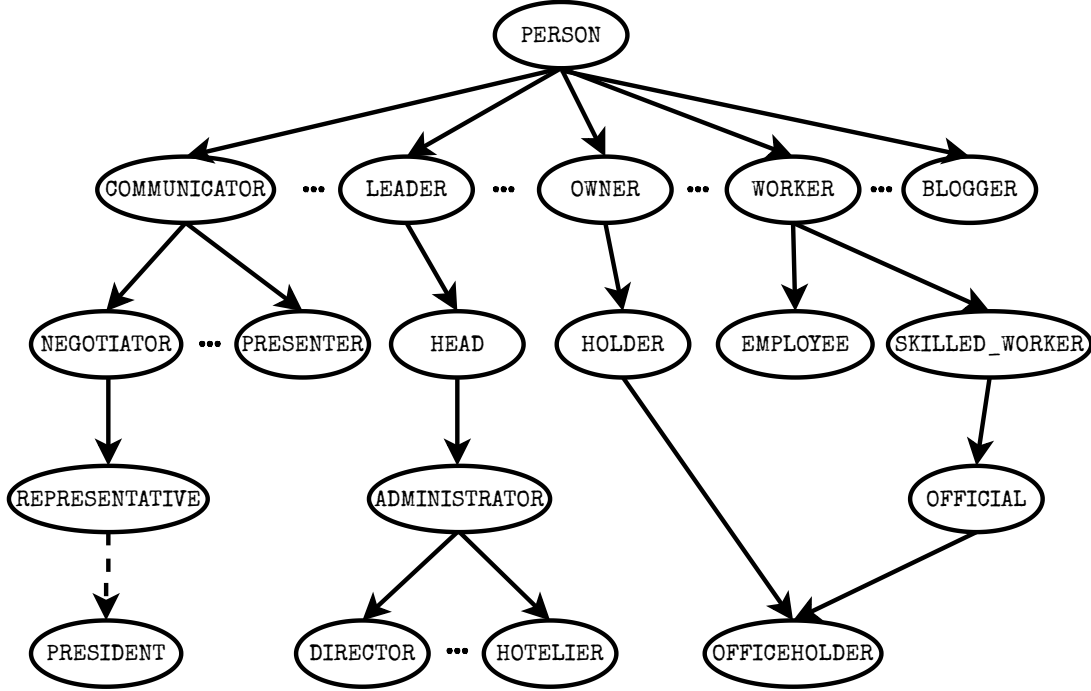


Figure 5.1: Entity-type Graph Excerpt for *Donald Trump*

Further, we define a function ϕ , which scores the representativeness of any type t^{e_j} with respect to an entity e_j , where $t^{e_j} \in \Pi_{\tau_m}(\hat{\tau}^{e_j})$. Equation 5.7 describes the formulation.

$$\phi(t^{e_j}, e_j) = \begin{cases} 1, & \text{if } t^{e_j} \text{ is representative w.r.t. } e_j \\ 0, & \text{if } t^{e_j} \text{ is not representative w.r.t. } e_j, \\ \forall t^{e_j} \in \Pi_{\tau_m}(\hat{\tau}^{e_j}) \end{cases} \quad (5.7)$$

Problem Definition

Given an entity-type graph $\tau_m^{e_j}$ for an entity e_j , predict the representative types $\sigma(\tau_m^{e_j})$, i.e., the set of types that are the most representative for the entity e_j . Formally, it can be defined as:

$$\sigma(\tau_m^{e_j}) = \{t_k \in \Pi_{\tau_m}(\hat{\tau}^{e_j}) \mid \phi(t_k, e_j) = 1\} \quad (5.8)$$

Figure 5.2 depicts the conceptual approach of the representative type(s) prediction framework. The generic classification process involves four steps to compute the representative entity type(s) $\sigma(\tau_m^{e_j})$ described as follows:

1) Direct KB Type Retrieval

In a first step, we retrieve all the directly associated entity types τ^{e_j} for an entity e_j in the ABox \mathcal{A} from the KB. In addition, we fetch the underlying taxonomy, i.e., TBox \mathcal{T} for this KB.

2) Transitive Type Computation

In a second step, the `subClassOf` relations of the TBox are used to retrieve all transitive types $\hat{\tau}^{e_j}$ for e_j .

3) Top-level Type Projection

Next, we project all the direct KB types along with their transitive types onto their top-level type(s) ($\Pi_{\tau_m}(\hat{\tau}^{e_j})$). In the context of our experiments (cf. Section 5.4), the top-level types are: `ARTIFACT`, `EVENT`, `ORGANIZATION`, `PERSON`, and `GEOENTITY`.

4) Representative Type(s) Prediction

In the final step, the representative types $\sigma(\tau_m^{e_j})$ for individual named entities are predicted. The prediction is done as per the underlying model (σ) (cf. Section 5.3). Depending on the model, a type in the entity type graph qualifies to be a representative type, or not. We describe various approaches for representative entity type classification in the following section.

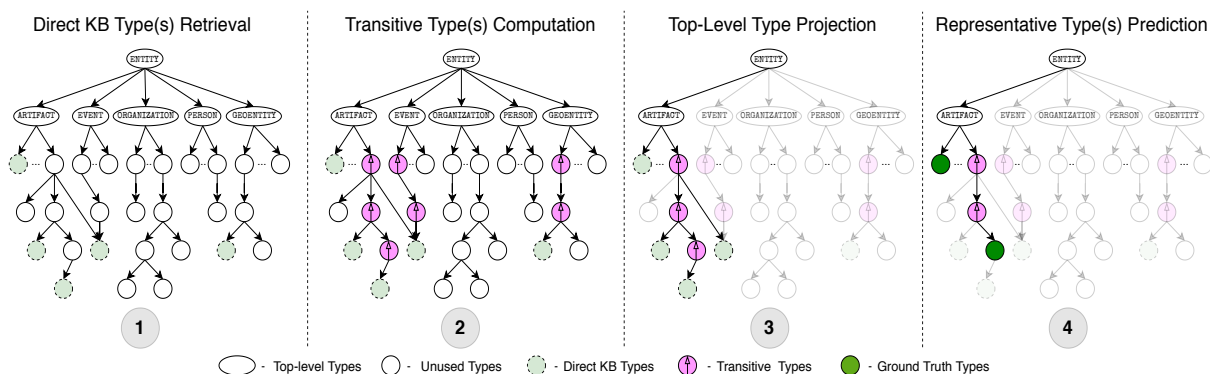


Figure 5.2: Conceptual Approach for Representative Type(s) Prediction of *Entities*

5.3 Representative Type(s) Classification

In this section, we introduce various entity representative type(s) classification approaches. To the best of our knowledge, no previous work is directly comparable with the task addressed by this work (cf. Chapter 3 for details). To this end, we describe several baseline models driven by heuristic considerations, as well as, state-of-the-art machine learning methods applied to our problem setting.

5.3.1 Baseline Models

In a first approach, we address our problem by considering suitable heuristics in identifying the representative type(s) in entity type graphs. To this end, we aim at exploiting the representative type(s) of a named entity via its leaf type(s). The intuition is that the leaf types in an entity type graph explicitly state the most specific characteristics of an entity while intermediate types do not. For instance, in Fig. 5.1 **PRESIDENT** is considered to be more specific (respectively representative) than, e.g., **COMMUNICATOR**. To this end, we introduce **Most Specific Type(s)** (σ_{MST}), **Most Generic Type(s)** (σ_{MGT}), and **Average Direct KB Type(s)** (σ_{ADKBT}) as variations of leaf-based representative type(s) prediction models.

Most Generic Type(s) Model

The σ_{MGT} model aims at representing a named entity e_j via its most generic leaf type(s) within the entity type graph $\tau_m^{e_j}$. For that purpose, the level of the highest leaf type(s) is identified. To avoid potential outliers and a potentially unfair comparison, we filter out those entity type(s) that are not seen in the set of all possible gold standard representative types G . Finally, the remaining highest leaf type(s) are predicted (e.g., **BLOGGER** in Fig. 5.1).

Most Specific Type(s) Model

The σ_{MST} model aims at capturing the representative type(s) of a named entity e_j from its most specific type(s). To this end, the leaf type(s) at the deepest level in the entity-type graph $\tau_m^{e_j}$ are identified. As before, the outlier type(s) are filtered out and the remaining ones are predicted (e.g. **PRESIDENT** in Fig. 5.1).

Average Direct KB Type(s) Model

In order to find a “compromise” between the two extremes introduced before, the third heuristic σ_{ADKBT} aims at identifying the number of representative type(s) with respect to the overall size of the underlying graph belonging to a named entity. To this end, we consider all the leaf types $L(\tau_m^{e_j})$ associated with a named entity e_j . Again, leaf type(s) not appearing in the set of possible gold standard representative type(s) G are filtered out. Then, rather than predicting all of these leaf type(s) $L(\tau_m^{e_j})$ as representative types, we aim to learn a subset whose size θ is adjusted with respect to the size of entity type graphs. More precisely, in order to determine θ for a named entity e_j , we consider all the entities from the training set that have a type graph of same size as e_j . We then compute θ as the average number of gold standard representative type(s) for those entities. In case there is no other graph of exactly the same size existing, an average count is computed. Based on θ a random sample is taken from the leaf types $L(\tau_m^{e_j})$ and predicted for entity e_j (cf. Algorithm 1 for the pseudo code of σ_{ADKBT}).

5.3.2 Entity Type(s) Classification

The heuristics introduced before are based on intuition by formulating a few handcrafted rules. However, they do not have capability of learning the patterns in entity type graphs

Algorithm 1 σ_{ADKBT} - Average Direct KB Type(s)

Input: Entity type graph $\tau_m^{e_j}$; set of possible gold standard types G ; training set TS **Output:** Representative type(s) $\sigma_{ADKBT}(\tau_m^{e_j})$ for entity e_j

```

1:  $N \leftarrow |TS|$ 
2:  $T \leftarrow L(\tau_m^{e_j}) \cap G$ 
3:  $entity\_count \leftarrow |\{e \in TS : |\Pi_{\tau_m}(\hat{\tau}^e)| = |\Pi_{\tau_m}(\hat{\tau}^{e_j})|\}|$ 
4:  $total\_GS\_types \leftarrow \sum_{e \in TS: |\Pi_{\tau_m}(\hat{\tau}^e)| = |\Pi_{\tau_m}(\hat{\tau}^{e_j})|} |G_e|$ 
5: if  $entity\_count = 0$  then
6:    $\theta \leftarrow \frac{\sum_{e \in TS} |G_e|}{N}$ 
7: else
8:    $\theta \leftarrow \lceil \frac{total\_GS\_types}{entity\_count} \rceil$ 
9: end if
10: if  $|T| > \theta$  then
11:    $\sigma_{ADKBT}(\tau_m^{e_j}) \leftarrow rand\_sample(T, \theta)$ 
12: else
13:    $\sigma_{ADKBT}(\tau_m^{e_j}) \leftarrow T$ 
14: end if

```

to make predictions. To address these limitations of heuristics based approaches, we formulate the entity representative type(s) classification as a machine learning task. As an individual named entity can have more than one representative type associated, we employ multi-label classification approach. To this end, we transform the representative type(s) classification into a set of sub-classifications task by utilizing the one-against-all strategy. This entails that if there are $|G|$ possible gold standard types, then we train $|G|$ different classifiers.

A separate classifier is trained for each type by considering the examples that have the concerned type in their gold standard as positive. For any entity, if the type is present in the corresponding entity type graph, but not in the gold standard of the entity then that example is considered as negative example. In order to make the prediction of representative type(s) for an entity, the decision is taken based on the outputs of all trained $|G|$ classifiers. Finally, the strategy for making representative type predictions with respect to the classifiers output varies among the models. We explain the prediction process in detail along with the respective models.

5.3.3 Random Forest Model

In the random forest based representative entity type(s) classification σ_{RF} we employ a random forest model [Breiman, 2001]. Random forest (RF) is a decision tree based ensemble model that utilizes a collection of decision trees to fit over the training dataset. The model aims to better generalize with the help of bootstrap aggregating for the training set and random subspace method for the feature set.

As the random forest classifier does not provide any direct input of graph based data, we encode the entity type graphs in form of feature vectors. For instance, for the dataset of top-level type PERSON, the size of the feature vector is equal to the number of all possible

types in the PERSON branch. Now, in order to encode an entity type graph $\tau_m^{e_j}$, we put a “1” in the feature vector at the corresponding locations of types present in the entity type graph and rest of the entries are initialized to “0”. Once the entity type graphs are encoded, a set of *RF* classifiers are trained using the one-against-all strategy as discussed in subsection 5.3.2. The classifiers learn to make a decision of “Yes” or “No” corresponding to each of the possible representative types in gold standard. For example, a classifier trained for the type PRESIDENT will solely decide whether PRESIDENT is a representative type for an entity e_j or not. To this end, corresponding types for which the classifiers decide “Yes” with respect to the entity type graph $\tau_m^{e_j}$, are predicted as the representative types for e_j .

One of the primary limitations of the aforementioned *RF* model is that it is not capable to exploit the inherent semantics from the hierarchical relations among types. Moreover, the representation of entity type graphs as feature vectors is quite sparse and not very qualitative in nature.

5.3.4 Graph Convolutional Network Model

In order to address the limitations of random forest model (*RF*) outlined in the previous section, we propose graph convolutional networks as model for representative type(s) classification. To this end, we introduce the basic framework for entity representative type(s) classification task via the adaption of graph convolutional networks. We denote the basic representative type(s) classification model by σ_{GCN} . We also present an enhancement over the basic *GCN* model denoted by σ_{PURE} in the subsequent section. The *GCN* based models aim to learn the patterns in entity type graphs of named entities with similar representative type(s).

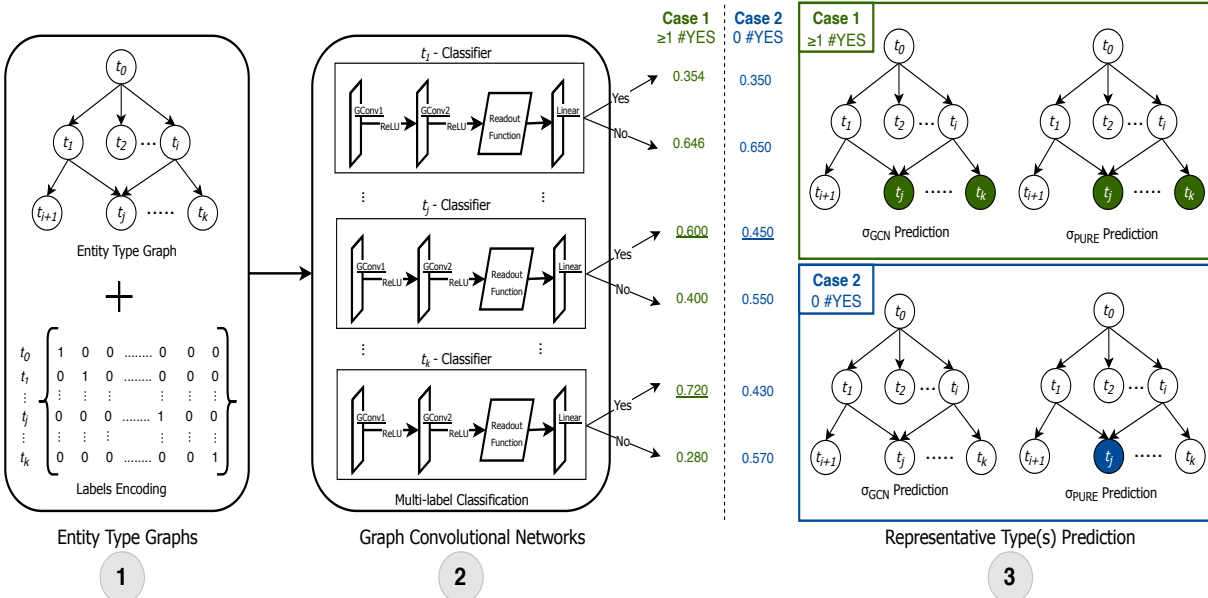


Figure 5.3: Illustration of the Entity Representative Type(s) Classification Pipeline for *GCN* based Approaches

A Graph Convolutional Network (*GCN*) [Kipf and Welling, 2017] is a multi-layer neural architecture that straightly operates on graph designed dataset. For adapting the graph convolutional networks in order to be applied for entity representative type(s) classification, we utilize an entity type graph for each of the entities as input. The number of nodes in the graph is $|\Pi_{\tau_m}(\hat{\tau}^{e_j})|$. We set label encoding v_t for the type t as the initial feature and explain more about the label encoding in the following section. Then, this information is fed into a two-layer *GCN* followed by a linear layer and SOFTMAX. In the two-layer *GCN*, the maximum exchange of information is possible between the nodes that are two hops away. We use the AGGREGATE function as suggested in [Kipf and Welling, 2017]. The mean of all nodes embedding is used as a READOUT function for representing the entity type graph. In our early experiments, we observed that the performance of a two-layer *GCN* is better than a one-layer *GCN*, while increasing the number of layers further did not considerably improve the performance (cf. Algorithm 2 for details).

Algorithm 2 Graph Convolutional Network Model

Input: Training set TS of entity-type graphs; Labels encoding $v_t, \forall t \in \tau_m$; *GCN* depth \mathcal{L} ; aggregator functions $AGGREGATE_l, l \in \{1, \dots, \mathcal{L}\}$; neighbourhood function η ; *READOUT* function; *LINEAR* function

Output: Representative type decision with confidence scores

```

1: for  $\tau_m^{e_j} \in TS$  do
2:    $h_t^{(0)} \leftarrow v_t, \forall t \in \Pi_{\tau_m}(\hat{\tau}^{e_j})$ 
3:   for  $l = 1, \dots, \mathcal{L}$  do
4:     for  $t \in \Pi_{\tau_m}(\hat{\tau}^{e_j})$  do
5:        $h_t^{(l)} \leftarrow AGGREGATE_l(h_v^{(l-1)}, \forall v \in (\eta(t) \cup t))$ 
6:        $h_t^{(l)} \leftarrow ReLU(h_t^{(l)})$ 
7:     end for
8:   end for
9:    $Z_{e_j} \leftarrow READOUT(h_t^{(l)}, \forall t \in \Pi_{\tau_m}(\hat{\tau}^{e_j}))$ 
10:   $Z_{e_j} \leftarrow LINEAR(Z_{e_j})$ 
11:   $decision \leftarrow SOFTMAX(Z_{e_j})$ 
12: end for

```

Label Encodings

The *GCN* based entity representative type(s) classification models take entity type graph $\tau_m^{e_j}$ as the input to make prediction for the entity e_j . As discussed previously, the entity type graphs are directed acyclic graphs with nodes annotated with types. It is important to provide the *GCN* with not just the graph structure but also the labels of the nodes, i.e., types. We encode the entity type node labels by using the one-hot vector encodings. For instance, while performing experiments for top-level type PERSON, we identify all the types present in the training set for PERSON branch. Let m be the total number of types identified in the previous step. Then, we create one-hot vector of dimension m where each entry in the vector corresponds to one of the types. The type vector consists of entries with value “0” except only one position being assigned to “1”. To this end, all the types are encoded as one-hot vectors with entry “1” at their designated position in the vector.

To this end, we train the *GCN* based classification model σ_{GCN} by utilizing the multi-label classification strategy described in Section 5.3.2. Figure 5.3 illustrates the conceptual pipeline of σ_{GCN} model. To make a prediction for an entity e_j , the entity type graph $\tau_m^{e_j}$ along with encoding of labels v_t are provided as input to all the $|G|$ classifiers and their corresponding decisions are gathered. The types whose classifiers made the “Yes” decision, are considered as the representative type(s) prediction for the entity e_j .

5.3.5 PURE

The σ_{PURE} model aims to address the limitations of basic *GCN* model σ_{GCN} . As previously discussed, the basic *GCN* based model selects the type(s) for which the corresponding classifiers output a “Yes” decision. These classifiers make predictions without any collaboration among them. This might lead to the cases where all the classifiers produce a “No” decision, and, thus not a single representative type for an entity is chosen. The σ_{PURE} model handles such cases with a more intelligent strategy. Specifically, when all of the $|G|$ classifiers predict “No”, the σ_{PURE} model selects the type whose classifier assigns the lowest score to “No” (or in other words, highest score to “Yes”). The intuition behind this strategy is that every entity should have at least one representative type. The model therefore exploits the level of confidence shown by the underlying classifiers when all of them do not make any positive prediction. To this end, the σ_{PURE} model aims to capture the inherent patterns in entity type graphs and employs an intelligent strategy to make prediction of representative type(s).

5.4 Experimental Evaluation

In the following, we present the experimental setting. To this end, we introduce the experimental set-up before presenting the experimental dataset, evaluation methods and classification results.

5.4.1 Experimental Setup

In the current work, we aim at identifying the most representative type(s) of a given named entity solely based on its type graph extracted from a knowledge base. For our experiments, we employ YAGO, which is a large scale automatically constructed knowledge base where named entities are populated from Wikipedia [Suchanek et al., 2007, Hoffart et al., 2013]. It contains (as of today) more than 17 million entities and 350,000 types. Due to the vast amount of available types, we limit ourselves in the first place to a realistic setting of types in order to keep it clean. Hence, we employ the WordNet [Miller, 1995] types of YAGO, which are 68,423 in total, structured according to 5 top-level types. In all our experiments, we utilize YAGO version 3.1³⁰ and English Wikipedia dump from July 01, 2019.

³⁰ <https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/downloads>

Named Entities Extraction

In order to collect named entities for our experiments, we aim at focusing on those that are particularly “multi-faceted”. That means, the named entities for our experiments should have a large number of types associated with them, out of which only a small number might be considered as representative. To this end, we selected the 100,000 named entities of type PERSON that have the most types associated with them. Since the number of named entities in YAGO is heavily varying and lower for the remaining 4 entity types, we fixed the same ratio in order to obtain comparable results across all entity types. As 100,000 named entities of type PERSON correspond to 6.7% of this type, we extracted the same ratio of named entities ARTIFACT, EVENT, ORGANIZATION, and GEOENTITY. Table 5.1 summarizes the key aspects of the data employed for the experiments. From the statistics, it can be observed that the PERSON type is the most populated and the most multi-faceted (due to its highest average size of the type graph per named entity).

Dataset	Entities Extracted	#Entities with ≥ 1 GS Types	Entity Type Graph Avg Size
PERSON	100,000	83,215	31
EVENT	29,289	15,053	11
ORGANIZATION	25,318	21,869	10
ARTIFACT	54,452	33,796	8
GEOENTITY	74,552	61,599	11

Table 5.1: Gold Standard (GS) Statistics

As the entity type graph derived from YAGO is a directed acyclic graph in nature, this implies that a named entity might be connected to more than one top-level type. In fact, this pattern mostly occurs among named entities of type ARTIFACT and GEOENTITY (e.g., the *Eiffel Tower*). Other connections to more than one top-level type exist, though they are rare. In our experiments, we focus on a named entity’s directly associated types and their transitive closure, by projecting respectively onto the branch of each of their top-level types. To this end, a classification is done for the representative type(s) of the identified top-level type.

Gold Standard Dataset Construction

Based on the previously collected multi-faceted named entities from the KB, we are now going to describe how the Gold Standard has been constructed. In order to define the most representative type(s) of a named entity, we utilize its Wikipedia page. In particular, we make use of the Wikipedia-specific writing style of articles: it can be observed, that the most-important and, thus, representative types can usually be found within the first or second sentence of its article in Wikipedia. For the extraction, we consider the named entities that are derived from Wikipedia in English language and have implemented top-level type-specific regular expressions, which are shown in Table 5.2. As a result, we obtain potentially many *candidate text strings* that might serve as a gold standard label for this entity. In order to link a *candidate text string* onto a specific type of the KB, we perform tokenization on word-level and check that the word exists in the types of the corresponding

entity or not. The mapped types are defined as the gold standard labels for that entity. The previously described approach results, for example, in the *candidate text strings*: actor, filmmaker, businessman, author, bodybuilder and politician; being extracted for *Arnold Schwarzenegger*. Applying the subsequent matching leaves the *candidate text string* “author” unmatched and results in the labeling by the following types: ACTOR, FILM-MAKER, BUSINESSMAN, BODYBUILDER and POLITICIAN. The overall statistics of the gold standard construction are summarized per top-level type in Table 5.1 and 5.3.

Dataset	Regular Expression
PERSON	, and , and who was the who is the served as appointed as worked as as the as a known as known for was a is a is an was an is the was the was is
EVENT	, and , and which was the which is the which were the which is a which was a which is an which was an as the as a known as known for was a is a is an was an is the was the were a were an were the
ORGANIZATION	, and is a is an is the was a was an was the are was is were
ARTIFACT	, and is a is an is the was a was an was the are was is were
GEOENTITY	, and is a is an is the was a was an was the are was is were

Table 5.2: Regular Expressions for Candidate Text String Extraction (‘|’ Represents ‘or’)

Dataset	Training Set Size	Test Set Size	#Possible GS Types	#Avg Types	#Min Types	#Max Types
PERSON	74,852	7,588	161	2	1	11
EVENT	7,700	819	20	2	1	6
ORGANIZATION	18,993	2,023	33	2	1	5
ARTIFACT	21,244	1,993	43	1	1	3
GEOENTITY	53,511	5,609	42	2	1	5

Table 5.3: Experimental Dataset Statistics

Experimental Dataset

The experimental dataset is constructed from the gold standard dataset, by performing a split of (90 : 10) for each top-level type. The 10% dataset is used for testing. In the 90% dataset, is split up again into 70% training and 30% for validation. In order to perform our experiment, we consider only those entity types that have at least 100 positive examples in the training dataset as explained in Section 5.3.2. To this end, Table 5.3 depicts the training and test dataset statistics after performing the aforementioned steps.

5.4.2 Model Configurations

We employ the Scikit-learn³¹ and PyTorch³² libraries for the implementation of the aforementioned models. The random forest based model is trained with the bootstrapped samples of the training set and the gini impurity criterion to measure the quality of a split. The number of estimators (i.e., decision trees) used is 100. For the implementation of the *GCN*, we use the DGL library³³. We use two hidden convolutional layers followed by one linear layer to achieve the pre-softmax logits. The number of neurons in the hidden convolutional layers are determined as per the geometric pyramid rule [Masters, 1993]. The network is trained using the Adam optimizer with a learning rate of 0.001 and for a number of 50 epochs. We also performed the experiments with different settings and chose the previously mentioned configurations as they provide better generalization over the training set.

5.4.3 Classification Results

Using the previously described experimental setup, we perform extensive experiments with respect to different approaches (cf. Section 5.3) and across several top-level Types (cf. Section 5.4.1). As indicated in Table 5.1 and 5.3, properties vary heavily depending on the top-level type. We evaluate the three heuristic based approaches namely σ_{MGT} , σ_{MST} and σ_{ADKBT} , one random forest based σ_{RF} , and the two *GCN* based approaches, i.e., σ_{GCN} and σ_{PURE} . In the following, we present the results conducted on the models introduced in Section 5.3. To this end, we present results for *entity-centric* as well as *type-centric* evaluation.

Entity-centric Evaluation

Table 5.4 and 5.5 summarize the *entity-centric* comparative analysis based on the macro- and micro-averaged evaluation respectively. We observe that both *GCN* based approaches dominate over the *RF* as well as the heuristic-based approaches. In particular, σ_{PURE} outperforms its competitors by from 2% to around 6% in macro-averaged F-measure as it can be seen in Table 5.4. The better performance of *GCN* based approaches is attributed to their capability of better encoding the structural type information in entity graphs and capturing the inherent patterns among similar entities. In addition, σ_{PURE} performs considerably high, because of its conceptual adaptation in prediction of the output label(s). Since σ_{GCN} collects the decisions of individual classifiers and predicts the type labels in “isolation”, this frequently leads to predictions where no label at all is being assigned. However, results are often very close to a positive decision and it should be enforced, that at least the highest scoring type should be assigned in order to ensure a classification as implemented in σ_{PURE} . In-line to the previous observations, *GCN* based approaches also improve the micro-averaged F-measure value by around 1% to 3% except in cases of **ORGANIZATION** and **GEOENTITY** datasets as reported in Table 5.5. Here, the performances of σ_{PURE} and σ_{RF} are around the same with σ_{RF} being higher by a margin of less than

³¹ <https://scikit-learn.org/>

³² <https://pytorch.org/>

³³ <https://www.dgl.ai/>

1%. The micro-averaging of scores has the risk of being affected from few high performing examples whereas the macro-averaging of scores gives equal importance to each of the examples in the test set. This makes the macro-averaged scores a bit more prominent in the current scenario.

Dataset	Metrics	σ_{MGT}	σ_{MST}	σ_{ADKBT}	σ_{RF}	σ_{GCN}	σ_{PURE}
PERSON	Precision	0.3007	0.2750	0.3263	0.5820	0.5826	0.6428
	Recall	0.2155	0.2290	0.3858	0.5221	0.5473	0.5948
	F-measure	0.2511	0.2499	0.3535	0.5504	0.5644	0.6179
EVENT	Precision	0.6800	0.1257	0.6963	0.8166	0.8202	0.8495
	Recall	0.5427	0.1015	0.7399	0.7709	0.7821	0.8096
	F-measure	0.6037	0.1123	0.7175	0.7931	0.8007	0.8290
ORGANIZATION	Precision	0.1212	0.3147	0.4198	0.8648	0.8752	0.8925
	Recall	0.1164	0.2685	0.5076	0.8583	0.8353	0.8506
	F-measure	0.1188	0.2898	0.4595	0.8615	0.8548	0.8711
ARTIFACT	Precision	0.3856	0.5903	0.6231	0.8296	0.8238	0.8886
	Recall	0.4052	0.6033	0.5926	0.8184	0.8238	0.8849
	F-measure	0.3952	0.5968	0.6075	0.8240	0.8238	0.8867
GEOMETRY	Precision	0.2313	0.4012	0.4203	0.7521	0.7400	0.7897
	Recall	0.2799	0.5192	0.6058	0.7681	0.7411	0.7839
	F-measure	0.2533	0.4527	0.4963	0.7600	0.7405	0.7868

Table 5.4: Macro-average Scores for *Entity-centric* Evaluation

Type-centric Evaluation

In order to quantify the complexity of identifying patterns linked with an individual **types**, we further conduct a **type-centric** assessment (cf. Table 5.6 and 5.7 for details). In macro scores, we observe here that *GCN* based approaches perform better for all top-level types (except for **ORGANIZATION**) with an improvement of around 2% to 6% in σ_{PURE} . In case of **ORGANIZATION**, the performances of σ_{RF} and σ_{PURE} are approximately the same with a minor advantage of 0.5% for σ_{RF} . There is an improvement of the micro-averaged F-measure value by around 1.5% to 3% for σ_{PURE} except for **ORGANIZATION**. In the latter case, the gain of σ_{RF} over σ_{PURE} is 0.63% only.

Apart from that, it can also be seen that the machine learning approaches perform better than the heuristics based approaches in general. The heuristic-based approaches rely on few handcrafted rules and do not have the capability to automatically learn from the training dataset. This limits their performance on predicting the representative type(s) for multi-faceted entities, and hinders a proper generalization over training examples.

Dataset	Metrics	σ_{MGT}	σ_{MST}	σ_{ADKBT}	σ_{RF}	σ_{GCN}	σ_{PURE}
PERSON	Precision	0.3885	0.3579	0.2977	0.6342	0.6083	0.6013
	Recall	0.2010	0.2258	0.3569	0.4952	0.5213	0.5542
	F-measure	0.2649	0.2769	0.3246	0.5562	0.5614	0.5768
EVENT	Precision	0.7820	0.6746	0.6435	0.8155	0.8194	0.8151
	Recall	0.4639	0.0842	0.6827	0.7481	0.7609	0.7789
	F-measure	0.5823	0.1497	0.6625	0.7803	0.7890	0.7966
ORGANIZATION	Precision	0.1744	0.5530	0.4481	0.8592	0.8681	0.8651
	Recall	0.0969	0.2625	0.5044	0.8447	0.8136	0.8257
	F-measure	0.1246	0.3561	0.4746	0.8519	0.8400	0.8449
ARTIFACT	Precision	0.5456	0.7439	0.6602	0.8193	0.8212	0.8151
	Recall	0.4033	0.5914	0.5825	0.8020	0.8081	0.8686
	F-measure	0.4638	0.6590	0.6189	0.8106	0.8146	0.8410
CREENTITY	Precision	0.3328	0.4556	0.3888	0.7718	0.7724	0.7585
	Recall	0.2505	0.4947	0.5552	0.7446	0.7139	0.7494
	F-measure	0.2858	0.4744	0.4574	0.7580	0.7420	0.7540

Table 5.5: Micro-average Scores for *Entity-centric* Evaluation

Dataset	Metrics	σ_{MGT}	σ_{MST}	σ_{ADKBT}	σ_{RF}	σ_{GCN}	σ_{PURE}
PERSON	Precision	0.2534	0.3403	0.3999	0.5492	0.5372	0.5578
	Recall	0.1564	0.2882	0.4093	0.4155	0.4535	0.4834
	F-measure	0.1934	0.3121	0.4046	0.4731	0.4918	0.5179
EVENT	Precision	0.5161	0.3715	0.5985	0.7926	0.7487	0.8325
	Recall	0.4000	0.1938	0.7001	0.7620	0.7700	0.7953
	F-measure	0.4507	0.2547	0.6453	0.7770	0.7592	0.8135
ORGANIZATION	Precision	0.2069	0.4870	0.5529	0.8072	0.8106	0.8236
	Recall	0.1771	0.4044	0.6341	0.7504	0.7117	0.7278
	F-measure	0.1908	0.4419	0.5907	0.7778	0.7580	0.7727
ARTIFACT	Precision	0.5761	0.6665	0.7105	0.8376	0.8454	0.8747
	Recall	0.4100	0.4712	0.5152	0.7125	0.7166	0.7917
	F-measure	0.4791	0.5521	0.5973	0.7700	0.7757	0.8312
CREENTITY	Precision	0.4779	0.5490	0.5679	0.7667	0.7686	0.7799
	Recall	0.4714	0.3611	0.6551	0.6924	0.6787	0.7086
	F-measure	0.4746	0.4356	0.6084	0.7277	0.7209	0.7425

Table 5.6: Macro-average Scores for *Type-centric* Evaluation

5.5 Findings on Representative Type Classification

In this chapter, we have presented *PURE*, a novel approach towards Pattern Utilization for Representative Entity type classification. To the best of our knowledge, this is the first

Dataset	Metrics	σ_{MGT}	σ_{MST}	σ_{ADKBT}	σ_{RF}	σ_{GCN}	σ_{PURE}
PERSON	Precision	0.4139	0.3761	0.3195	0.6544	0.6311	0.6401
	Recall	0.2010	0.2258	0.3837	0.4952	0.5213	0.5418
	F-measure	0.2706	0.2822	0.3487	0.5638	0.5709	0.5869
EVENT	Precision	0.8183	0.7225	0.6821	0.8511	0.8525	0.8551
	Recall	0.4639	0.0842	0.6827	0.7481	0.7609	0.7766
	F-measure	0.5921	0.1508	0.6824	0.7963	0.8041	0.8140
ORGANIZATION	Precision	0.1845	0.5616	0.4656	0.8711	0.8808	0.8820
	Recall	0.0969	0.2625	0.5044	0.8447	0.8136	0.8229
	F-measure	0.1271	0.3578	0.4842	0.8577	0.8459	0.8514
ARTIFACT	Precision	0.6218	0.7965	0.7666	0.8892	0.8890	0.8952
	Recall	0.4033	0.5914	0.5825	0.8020	0.8081	0.8621
	F-measure	0.4893	0.6788	0.6620	0.8434	0.8466	0.8783
GEOENTITY	Precision	0.3491	0.4672	0.4090	0.7894	0.7901	0.7959
	Recall	0.2505	0.4947	0.5552	0.7446	0.7139	0.7399
	F-measure	0.2917	0.4806	0.4710	0.7664	0.7501	0.7669

Table 5.7: Micro-average Scores for Type-centric Evaluation

ever approach aiming at extracting information about the most representative type(s) of a named entity solely via structural information from an entity’s KB type graph. Our study on representative type classification validates our hypothesis that named entities of certain types share a multitude of common and (at the same time) characteristic facets (cf. Hypothesis 2 in Chapter 1). In our extensive experiments, we have shown that *PURE* significantly outperforms competitors, including an implementation of a (plain) *GCN*. In particular, our experiments show that *PURE* performs best in macro-type assessment on the *PERSON* type, which is inherently complex and diverse.

As part of this study, we also observe the following notable findings with our *PURE* framework which have been discussed in the following subsections.

Dataset for Representative Type(s)

As part of this comprehensive study, we introduce a gold standard dataset for representative type(s) identification of named entities in knowledge bases. The task as well as the prepared gold standard data is novel in itself and might encourage other researchers to perform various future studies. To this end, the dataset is available from the project page of *PURE*³⁴.

Prediction Performance of different Top-level Types

The results in the previous subsections show that the prediction for *PERSON* entities is more complex than for the remaining top-level types. This has various reasons. First, *PERSON* entities are inherently multi-faceted and versatile at the same time. While patterns are

³⁴ *PURE* Dataset: <https://spaniol.users.greyc.fr/research/PURE/PURE.zip>

obviously learnable via the graph structures, “exceptions from the rule” are more frequent here. Second, in line with the previous observation, the number of possible types for a PERSON is considerably higher, i.e., 161 possible types for PERSON and below 50 for others (cf. Table 5.3). Lastly, type graphs for PERSON entities are intrinsically more complex because of the presence of a relative more fine-grained type hierarchy. It is also worth pointing out that *PURE* considerably outperforms all other approaches in case of PERSON, which confirms our hypothesis that the graph neural networks can better capture the predominant type patterns in entity graphs.

Data Curation Aspects

During the assessment of our experimental results, we figured out that results of *PURE* have been negatively impacted by the (mostly) incremental nature of Wikipedia. For instance, we identified several ACTRESSES, such as *Kelly Clarkson*, for which our approach did a prediction, but the Wikipedia version employed for our experiments did not yet contain the corresponding information. Due to various cases of that kind, we also believe in a potential of our method in the context of data curation.

Graph Convolution Neural Networks

Our thorough experiments show that graph convolutional neural networks can take into account the representation of entity type graphs, and thereby be employed to predict the representative type(s) for entities. Thus *GCN* based approaches (and σ_{PURE} in particular) outperform the other machine learning as well as heuristic based approaches for all types except ORGANIZATION by a considerable margin. This confirms our initial hypothesis that graph neural networks can be adapted to identify the entity representative type(s) by using solely the structural information from the entity type graphs. As the model does not use any other features except the hierarchical structure of entity type graphs, this makes the approach purely semantic and at the same time demonstrates the viability of our approach.

Our comprehensive study and performed experiments show that our approach successfully captures the patterns inherent in entity graphs in order to predict the representative type(s). However, there are certain cases for which *PURE* faces challenges in predicting the representative type(s). As before mentioned, the performance of *PURE* might suffer because of incremental nature of Wikipedia. It can also affect the learning of *PURE*. However, these kinds of cases are rarely observed in the practice and, thus, do not hamper the overall viability of the system.

After identifying the most relevant type(s) for the named entities, the question arises - How to utilize this information for better contextualization of the Web content?. The most “generic” approach is the exploitation of the entity types for document annotation and curation. To this end, there is a need for automatic annotation of a document.

Chapter 6

Concise Content Annotation

6.1	Conceptual Approach	72
6.2	Semantic Content Tagging Assessment	74
6.2.1	Assessment Dataset & Measures	74
6.2.2	Qualitative Assessment	75
6.2.3	Quantitative Assessment	75
6.3	AnnoTag Interface	76
6.4	Findings on Content Annotation	76

In the previous chapter, we focused on representative entity type classification. A “natural” next step is to utilize the concise entity information now in the context of document annotation. Thus, the main question is - “How to concisely annotate a document and interlink it with concepts of LOD?” (cf. RQ3 in Chapter 1). For this purpose, we assess in this chapter the approach of concise content annotation as a means of supporting the process of digital curation. In particular, we compare various entity-level annotation methods and highlight the importance of concise semantic tagging based on qualitative as well as quantitative evaluations. To this end, we introduce AnnoTag which aims at providing concise content annotations by employing entity-level analytics in order to derive semantic descriptions in the form of tags.

Concise content annotations are an indispensable prerequisite for efficient and effective digital curation. In particular, it is crucial to capture the essence of a document by extracting its semantic. To this end, experts such as librarians or curators index contents by keywords and (potentially) connect them with an underlying taxonomy (or ontology) in order to facility structured search and retrieval. However, this process is time-consuming and labor-intensive. At the same time, digital preservation and digitization leads to a sheer abundant amount of data to be curated. In an era of artificial intelligence (AI) the question therefore arises: how to support digital curation and assist curators in concisely annotating the data? One the hand side, we observe a need for an a flexible as possible tagging-like content annotation [Macgregor and McCulloch, 2006] while, on the other hand side, a way of linking these annotations with an underlying taxonomy (or ontology) is desired in order to support “guided” retrieval. With the availability of automatically generated knowledge

bases knowledge (KBs) such as DBpedia [Auer et al., 2007] or YAGO [Suchanek et al., 2007] software for named entity disambiguation such as Open Calais³⁵ or AIDA [Hoffart et al., 2011b] has emerged. Thus, it becomes now possible to “distill” the semantic of a document by identifying the named entities contained and analyzing them. As a result, a document might be “summarized” by its named entities and the type(s) they belong to. For instance, a document containing entities of type *ATHLETE* and *PLAYER* might be associated with *SPORTS*, while another document holding entities of type *POLITICIAN* and *LAWYER* might be linked with *POLITICS*. However, YAGO contains around half a million of types. Thus, digital curation requires the right balance between too fine-grained and too abstract annotations by focusing onto the most concise types. We therefore employ our PURE framework (cf. Chapter 5) in order to identify the most concise types out of the abundance of information captured in KBs about (prominent) entities such as *Kamala Harris* or the *International Monetary Fund*. To this end, we address in this study the assessment of concise content tagging based on entity-level analytics. In particular, the prominent contributions of this work are as following:

- incorporating the PURE framework to derive semantic tag of a document
- generating “Semantic LOD Tags” (linked open data) that allow an interlinking of derived tags with the LOD cloud
- providing an online interface to explore & visualize the named entities contained along with their concise annotation and their corresponding LOD tags

6.1 Conceptual Approach

In the following, we introduce the conceptual approach of entity-driven semantic tagging. Figure 6.1 presents the consecutive steps employed in entity-driven semantic tagging. The process begins with a document upload (cf. ① in Figure 6.1) to a collection or a digital archive. The two main steps relevant for digital curation follow then subsequently by automatically exploiting the document’s inherent semantic from the named entities contained. To this end, we employ the named entity disambiguation tool AIDA [Hoffart et al., 2011b] (cf. ② in Figure 6.1). Thus, we are able to extract the named entities contained in YAGO [Suchanek et al., 2007, Hoffart et al., 2013]. Apart from the information about the canonicalized named entities themselves the KB contains a wealth of additional facts about them, too. In the context of digital curation, the underlying ontology’s type hierarchy is of particular interest, because it gives insights for a more fine-grained content annotation. Considering the example highlighted in Figure 6.1, we observe, for instance, for US vice-president *Kamala Harris* a total of 39 types derived from the transitive closure or a total of 16 types for the *International Monetary Fund* stored in YAGO. These types “stem” from only 10 directly associated types for *Kamala Harris* and 3 directly associated types for the *International Monetary Fund*. Considering the “inflation” of types obtained when computing the transitive closure, it becomes evident that a concise type annotation

³⁵ <https://www.refinitiv.com/en/products/intelligent-tagging-text-analytics>

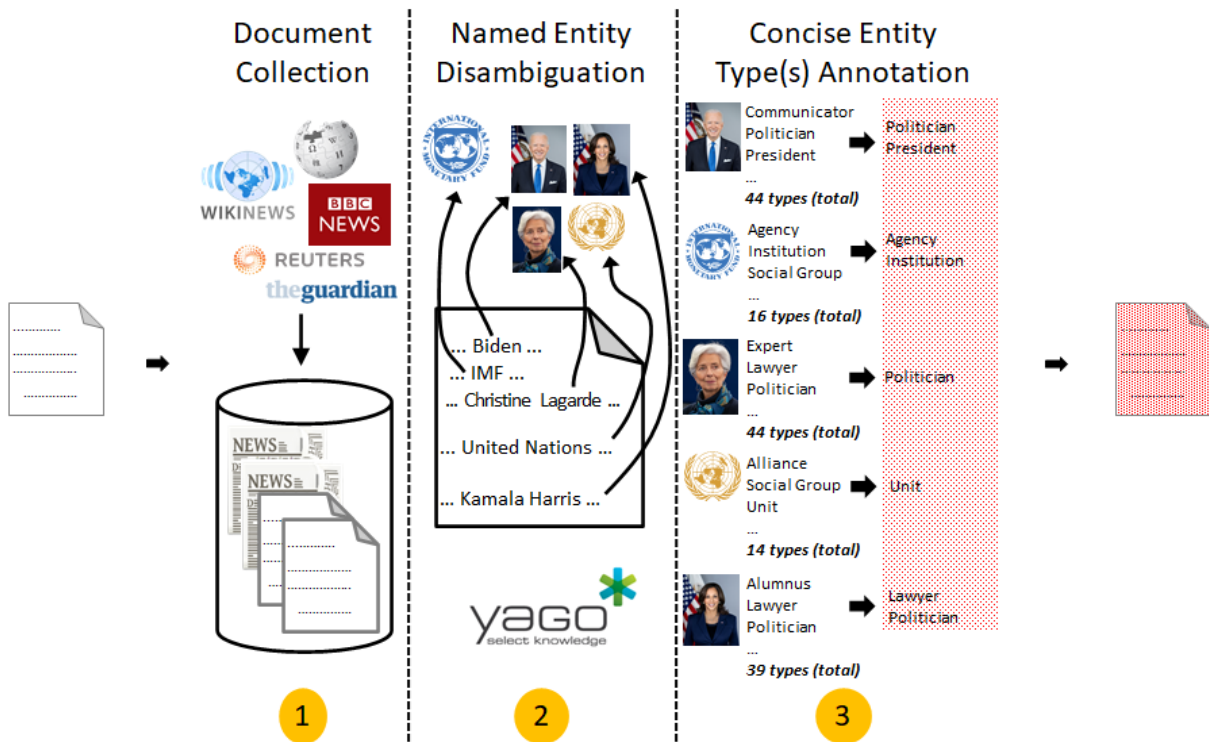


Figure 6.1: Steps in Entity-driven Semantic Tagging

is required. To this end, the most relevant types should be selected from the extensive type set contained in the transitive closure. For this purpose, we employ as a third - and optional - step the PURE (Pattern Utilization for Representative Entity type classification) framework [Kumar et al., 2020] (cf. ③ in Figure 6.1), which builds upon around 300 types structured by the 5 top-level types from the YAGO ontology. By doing so, we derive the most representative types of each named entity and a concisely annotated pseudo document as indicated by the dotted overlay of types in step ③ of Figure 6.1. An excerpt of resulting example document is shown in Figure 6.2. It consists of RDF triples (subject-predicate-object). ‘rdf-schema#member’³⁶ and ‘owl#sameas’³⁷ are used as the predicates in the annotated document. The object part of the triple represents the generated concise annotations where predicate is ‘rdf-schema#member’. These annotations are based on the AnnoTag system [Kumar and Spaniol, 2021a]. The AnnoTag system is a Web based interface which provides exploration and visualization of concise human-interpretable content annotations by simultaneously providing links with semantic concepts of the LOD cloud. The functionality of the system has been discussed in detail in Section 6.3. In order to ensure best possible interpretability of the types a reference to their instances in YAGO and (exploiting the `sameAs` link also directly to) DBpedia is provided so that they can be understood in the context of the underlying ontology.

³⁶ <https://www.w3.org/2000/01/rdf-schema#member>

³⁷ <https://www.w3.org/TR/owl-ref/#sameAs-def>

```

<My_Document_123> <http://www.w3.org/2000/01/rdf-schema#member> <company>.
<My_Document_123> <http://www.w3.org/2000/01/rdf-schema#member> <minister>.
...
<company> <http://www.w3.org/2002/07/owl#sameas> yago3:<wordnet_company_108058098>.
<company> <http://www.w3.org/2002/07/owl#sameas> <http://dbpedia.org/class/yago/Company108058098>.
<minister> <http://www.w3.org/2002/07/owl#sameas> yago3:<wordnet_minister_110320863>.
<minister> <http://www.w3.org/2002/07/owl#sameas> <http://dbpedia.org/class/yago/Minister110320863>.
<journalist> <http://www.w3.org/2002/07/owl#sameas> yago3:<wordnet_journalist_110224578>.
<journalist> <http://www.w3.org/2002/07/owl#sameas> <http://dbpedia.org/class/yago/Journalist110224578>.
...

```

Figure 6.2: Excerpt of an Annotated Example Document

A fully annotated document is available provided via the project Website³⁸. Ultimately, the resulting automatically annotated document (like the example mentioned before) can then be refined or revised by a human annotator.

6.2 Semantic Content Tagging Assessment

6.2.1 Assessment Dataset & Measures

The performance of semantic content tagging was assessed by conducting a qualitative and quantitative analysis on the goodness of the automatically generated semantic tags and report results on Precision and Mean Reciprocal Rank (MRR). To this end, we compared three variations of entity-driven semantic tagging:

- *Transitive Entity Types*: all types obtained from computing the transitive closure of a named entity
- *Direct Entity Types*: the types that are directly linked with a named entity
- *Concise Entity Types*: the concise types derived for a named entity by employing PURE [Kumar et al., 2020]

Experiments were performed by utilizing a large data set³⁹ consisting of 3,824 articles for annotation. Out of the aforementioned documents, we drew a random sample of 50 documents. The 50 documents in the evaluation data set contained on average slightly more than 500 words and 25 entities, each. Based on the entities from this evaluation data set, there were in total 811 types identified for annotation with the *Transitive Entity Types* method, 430 for the *Direct Entity Types* method and 114 for the *Concise Entity Types* method of AnnoTag. In order to provide a more detailed overview about the corresponding types related to our experimental data set, a detailed list is provided on our Website⁴⁰. The actual assessment was manually performed based on an individual

³⁸ Annotation of an example document

https://spaniol.users.greyc.fr/research/AnnoTag/Example_Annotation.txt

³⁹ Harvard Dataverse News Articles <https://doi.org/10.7910/DVN/GMFCTR>

⁴⁰ List of possible annotation types in DBpedia and YAGO

https://spaniol.users.greyc.fr/research/AnnoTag/Annotation_Types.zip

evaluation and a three-level grading scheme (2: “highly concise annotation(s)”, 1: “concise annotation(s)”, 0: “unsuitable annotation(s)”). Evaluation data are publicly available here⁴¹. Based on these evaluations, we computed the following measures:

- 1) “Hard” Precision: 2 \rightsquigarrow relevant, 1 or 0 \rightsquigarrow irrelevant
- 2) “Soft” Precision: 2 or 1 \rightsquigarrow relevant, 0 \rightsquigarrow irrelevant
- 3) “Emulated” MRR: 2 \rightsquigarrow 1st rank, score = 1
1 \rightsquigarrow 2nd rank, score = 0.5
0 \rightsquigarrow no rank, score = 0

6.2.2 Qualitative Assessment

The evaluation results of our qualitative assessment are summarized in Table 6.1. It can be observed that those methods that limit the perimeter of annotations (i.e. *Direct Entity Types* and *Concise Entity Types*) achieve the highest scores in both, Precision and emulated MRR. In particular, *Concise Entity Types* achieves 92% in “Soft” Precision. Considering the fact, that the automatically generated semantic tags are supposed to be used as an assistance in a (semi-)automatic digital curation process involving a human curator, the remaining annotation errors might be easily corrected while saving valuable human time and labor due to the high quality of the automatically generated annotations.

Measure Entity-level Analytics Method	“Hard” Precision	“Soft” Precision	“Emulated” MRR
<i>Transitive Entity Types</i>	0	0.28	0.14
<i>Direct Entity Types</i>	0.4	0.78	0.59
<i>Concise Entity Types</i>	0.72	0.92	0.82

Table 6.1: Qualitative Assessment over 50 randomly sampled Documents

6.2.3 Quantitative Assessment

In a second evaluation we now study the quantitative dimension of entity-level semantic content tagging. Not surprisingly, the number of created tags differs significantly for the various methods (cf. Table 6.2 for details). Exploiting the information by computing the transitive closure as in *Transitive Entity Types* leads to an “explosion” of tags. Inline with the observations from the qualitative assessment in Section 6.2.2 it becomes clear that this methods somewhat overshoots the target. This is due to the fact, that at the upper part of the ontology very generic types (such as **ORGANISM**, **LIVING_THING** or **ABSTRACTION**) are located. As these types are not sufficiently specific, they lead to an overall decay in Precision and MRR. In contrast, the order of tags assigned by *Direct Entity Types* and *Concise Entity Types* are about 3 to 10 times less and, thus, leading to a more concise result with higher Precision and emulated MRR scores. In particular, the

⁴¹ Semantic Tagging Assessment

https://spaniol.users.greyc.fr/research/AnnoTag/Evaluation_Data.zip

method of *Concise Entity Types* extraction by PURE [Kumar et al., 2020] shows that a few (concise) types are best suited in order to capture a content’s semantic. At the same time, an Average of 9.28 and a Median of 8 implies that the amount of tags to be verified and/or corrected by a human curator is manageable.

Entity-level Analytics Method	Annotated Type Counts		
	Total	Average	Median
<i>Transitive Entity Types</i>	5,161	103.22	99.5
<i>Direct Entity Types</i>	1,760	35.2	32
<i>Concise Entity Types</i>	464	9.28	8

Table 6.2: Quantitative Assessment over 50 randomly sampled Documents

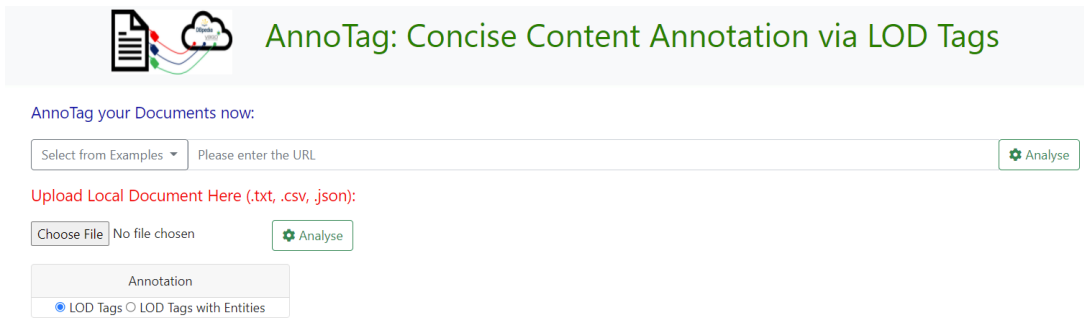
6.3 AnnoTag Interface

The AnnoTag system presents the concise annotation of documents with semantic LOD tags. Figure 6.3 depicts the steps of the AnnoTag system. Figure 6.3a shows the initial interface for document upload (cf. step ① in Figure 6.1). Here, a choice can be made between uploading a local file to the server and providing a URL. In addition, two configurations can be chosen: an annotation with LOD tags (only) or an annotation with LOD tags including the named entities. After that, the document is processed and the named entities contained are identified by employing AIDA [Hoffart et al., 2011b] (cf. step ② in Figure 6.1). The result of this process is then shown as an overview (cf. Figure 6.3b). Subsequently, the concise tags per entity are extracted, for which we employ our PURE framework [Kumar et al., 2020]. Figure 6.3c shows the obtained types in the AnnoTag user interface. Finally, the semantic LOD tags are generated and exported as RDF triples (cf. step ③ in Figure 6.1). An example excerpt of a resulting document is highlighted in Figure 6.2. It can be observed, that the RDF triples provide links of the assigned concise types to the LOD cloud, in particular, to the corresponding concepts in YAGO and (exploiting the `sameAs` link also directly to) DBpedia. The overall process including a functionality video, live interface and the assessed documents can be found at the AnnoTag Website⁴².

6.4 Findings on Content Annotation

In this chapter, we have presented several entity-level annotation approaches for concise semantic tagging of a document. In particular, we introduced the AnnoTag framework which exploits entity type information driven by the PURE framework (cf. Chapter 5) in order to assess and visualize the concise tag generation and inter-connectivity with the “Semantic LOD Tags”. The uniqueness of our methodology stems from analytics on the entity level. In contrast to the other approaches (cf. Chapter 3), our approach is applicable

⁴² AnnoTag Website <https://spaniol.users.greyc.fr/research/AnnoTag/>



AnnoTag your Documents now:

Select from Examples ▾ Please enter the URL Analyse

Upload Local Document Here (.txt, .csv, .json):

Choose File No file chosen Analyse

Annotation

LOD Tags LOD Tags with Entities

(a) AnnoTag User Interface for Document Upload

Fetches Content

Content
NEs Link
NEs Cloud

Entity	Wikipedia Link
Angela Merkel	http://en.wikipedia.org/wiki/Angela_Merkel
Bavarian	http://en.wikipedia.org/wiki/Bavaria
Belgian	http://en.wikipedia.org/wiki/Belgium
Brussels	http://en.wikipedia.org/wiki/Brussels
Charles Michel	http://en.wikipedia.org/wiki/Charles_Michel_(Belgian_politician)
Christine Lagarde	http://en.wikipedia.org/wiki/Christine_Lagarde
Commission	http://en.wikipedia.org/wiki/European_Commission
Deutsche Welle	http://en.wikipedia.org/wiki/Deutsche_Welle
Dutch	http://en.wikipedia.org/wiki/Germany
ECB	http://en.wikipedia.org/wiki/European_Central_Bank
Europe	http://en.wikipedia.org/wiki/Europe
European	http://en.wikipedia.org/wiki/European_Union
European Central Bank	http://en.wikipedia.org/wiki/European_Central_Bank
European Commission	http://en.wikipedia.org/wiki/European_Commission

(b) Listing of the Named Entities

Content Annotation

NEs LOD Tags
LOD Tags Cloud

Entity	Wikipedia Entity	LOD Tags
Angela Merkel	Angela Merkel	politician
Angela Merkel	Angela Merkel	president
Angela Merkel	Angela Merkel	chemist
Bavarian	Bavaria	region
Bavarian	Bavaria	region
Belgian	Belgium	district
Brussels	Brussels	capital
Christine Lagarde	Christine Lagarde	politician
Commission	European Commission	unit
Deutsche Welle	Deutsche Welle	station
Dutch	Germany	district
ECB	European Central Bank	company
ECB	European Central Bank	institution
European	European Union	union

(c) Concise Types per Entity

Figure 6.3: Steps of the AnnoTag Framework

in general domain and, at the same time, provides concise tags for a document. Moreover, the generated semantic tags are human and machine-interpretable (by linking generated semantic tags to LOD cloud). From our evaluation it can be observed that entity-level analytics is capable of achieving a high annotation quality based on a considerable small, but concise, amount of tags. The superior quality and comprehensibility of an annotated document confirm our hypothesis (cf. Hypothesis 3 in Chapter 1).

To this end, we believe that a method utilizing tagging such as *Concise Entity Types* might become a valuable asset in digital curation. Further, we provide end users a Web interface to explore and visualize the annotation of document along with the linkage to LOD tag cloud. In addition, the download of the annotated document with or without entity information is supported. Last but not least, we see of potential of utilizing the concisely annotated documents, e.g., in the context of semantic search (discussed in Chapter 7), concept matching of documents, etc.

Chapter 7

Semantic Search via Entity-Types

7.1	Conceptual Approach	80
7.1.1	Document Collection	80
7.1.2	Semantic Annotation	81
7.1.3	Semantic Retrieval & Exploration	81
7.2	SEMANNOREX Interface	82
7.3	Findings on Semantic Search	86

As described in the previous chapter, concise semantic annotation has an obvious application scenario in digital curation. In this chapter, we investigate an additional use-case by utilizing the concisely annotated documents for semantic search. As such, we address our fourth research question by asking - “Which retrieval method to apply in order to support ontology-driven retrieval?” (cf. RQ4 in Chapter 1). To this end, we present SEMANNOREX, which supports semantic search based on entity type information.

Collaborative tagging has been widely established as a method of content annotation and retrieval since the beginning of the Web 2.0 era [Macgregor and McCulloch, 2006]. Applications range from tagging of books⁴³, via annotations of songs⁴⁴, up to editorial contents provided in commercial platforms⁴⁵. To this end, tagging requires qualified human annotators producing a “bag of tags” content annotation. The result is a flat model that isn’t capable of exploiting the inherent semantic dependencies associated with each tag, e.g., the similarity between an **ATHLETE** and a **PLAYER**. However, the proliferation of linked open data (LOD) and knowledge bases (KBs) such as DBpedia [Auer et al., 2007] or YAGO [Suchanek et al., 2007, Hoffart et al., 2013], allows making those dependencies expressible and measurable. In order to overcome the shortcoming of relying onto high-quality manual annotations within a “bag of tags” representation, we present the SEMANNOREX (SEMantic ANNOtation, Retrieval and EXploration) framework for semantic search via entity-types.

⁴³ <https://blog.librarything.com/main/category/tags/>

⁴⁴ <http://www.deezer-blog.com/tags-in-search/>

⁴⁵ <https://www.bbc.co.uk/blogs/aboutthebbc/tags>

In summary, the salient contributions of this work are as following:

- incorporating the PURE framework to derive annotation and subsequently, the representation of a document
- the adaptation of semantic content similarity in order to measure the structural similarity between a query and a document
- providing a Web interface for the semantic search and the exploration of Web contents with respect to its inherent semantic

7.1 Conceptual Approach

7.1.1 Document Collection

The conceptual approach of SEMANNOREX is shown in Figure 7.1. It builds upon more than 400 types structured by the 5 top-level types from the YAGO ontology [Suchanek et al., 2007]. In our study, we utilize an English corpus of Web news contents and Wikinews⁴⁶ (cf. ① in Fig. 7.1).

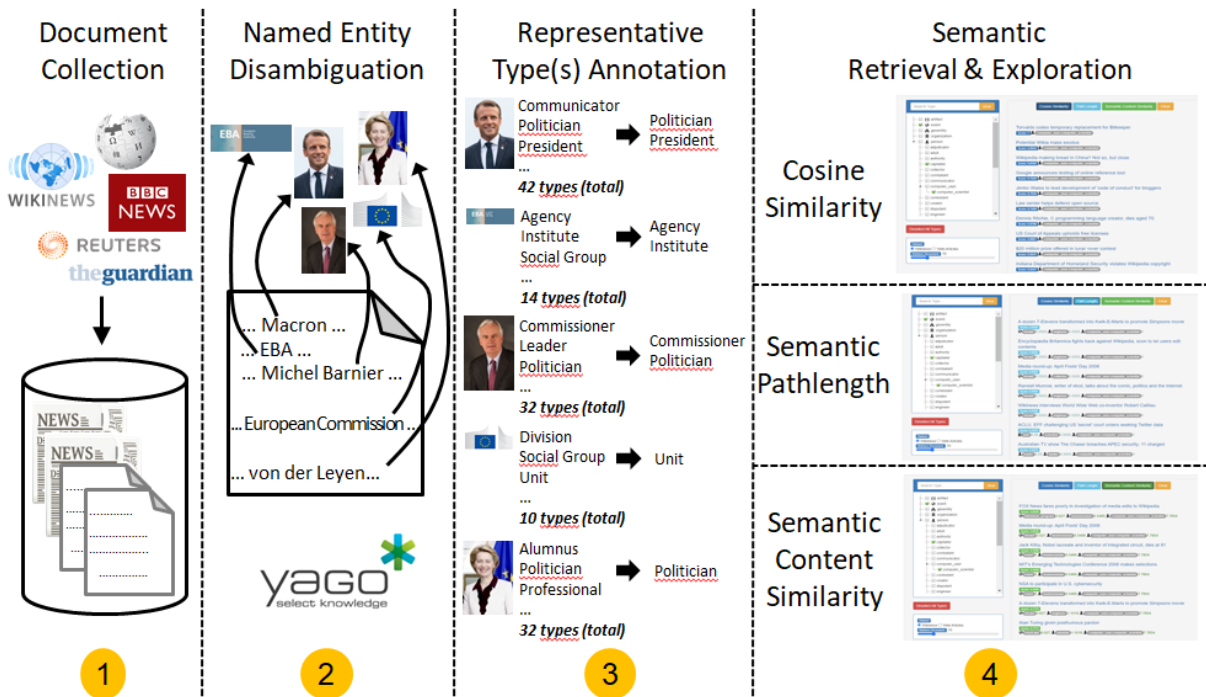


Figure 7.1: Conceptual SEMANNOREX Pipeline

⁴⁶ <https://spaniol.users.greyc.fr/research/SEMANNOREX/SEMANNOREX.zip>

7.1.2 Semantic Annotation

The semantic annotation is obtained from the named entities present in the document. These named entities in the Web contents can be identified by employing a named entity disambiguation tool [Mendes et al., 2011, Hoffart et al., 2011b, Yosef et al., 2011]. For SEMANNOREX, we employ AIDA-light [Nguyen et al., 2014] for disambiguation of Web news contents as well as mapping linked Wikipedia pages onto the canonicalized YAGO [Suchanek et al., 2007, Hoffart et al., 2013] entity for Wikinews data (cf. ② in Fig. 7.1). Since KBs capture plenitude of information about named entities via the transitive closure (e.g. in YAGO 42 types for *Emmanuel Macron* or 14 for the *European Banking Authority (EBA)*), we focus on the most “representative” type(s) by employing the PURE framework (cf. ③ in Fig. 7.1) (cf. Chapter 5, [Kumar et al., 2020]).

7.1.3 Semantic Retrieval & Exploration

For retrieval we allow three different methods (cf. ④ in Fig. 7.1). We define q as the user query types and d the types of an annotated document, where q_{τ_i} and d_{τ_j} stands for the types present in the query and the document, respectively.

$$q = \{q_{\tau_1}, q_{\tau_2} \dots q_{\tau_i}\} \quad \text{and} \quad d = \{d_{\tau_1}, d_{\tau_2} \dots d_{\tau_j}\}$$

Here, a non-zero value indicates the presence of the type. The computation is then based on the vectors for the query $\Pi(q)$ and the document $\Pi(d)$.

Cosine Similarity

The document vector entries are assigned as the number of times a type appears in the same document. The computation of cosine similarity (cf. [Manning et al., 2008]) is defined as:

$$\cos(\Pi(d), \Pi(q)) = (\Pi(d) \cdot \Pi(q)) / (\|\Pi(d)\| \|\Pi(q)\|)$$

Semantic Pathlength

In order to incorporate the structure of underlying ontology, we also utilize the Pathlength [Slimani, 2013, Jia et al., 2018] as measure of semantic similarity defined as follows:

$$\text{sempath}(q, d) = \text{avg}_{1 \leq m \leq i} \left(\max_{1 \leq n \leq j} \left(\frac{1}{1 + \text{pathlength}(q_{\tau_m}, d_{\tau_n})} \right) \right)$$

Semantic Content Similarity (SCS) of KB Types

In *SCS* we adopt the Resnik approach [Resnik, 1995] of assessing type similarity within our ontology. To this end, we treeify the directed acyclic graph (DAG) of the YAGO ontology by (recursively) duplicating child nodes having multiple parent nodes in each parent’s (sub-) branch (cf. Figure 7.2). As a consequence of treeification, content types annotations are classified in the (sub-)branch associated with the parent node of the “predominant” top-level type. This means, the “duplicated type” will be linked only to that parent node, which belongs to the top-level type where the majority of the remaining

types of this content belong to. In case, where the majority voting leads to a draw, the content will be typed to each of these duplicated types. The pseudo code of the ontology treeification process is presented in Algorithm 3.

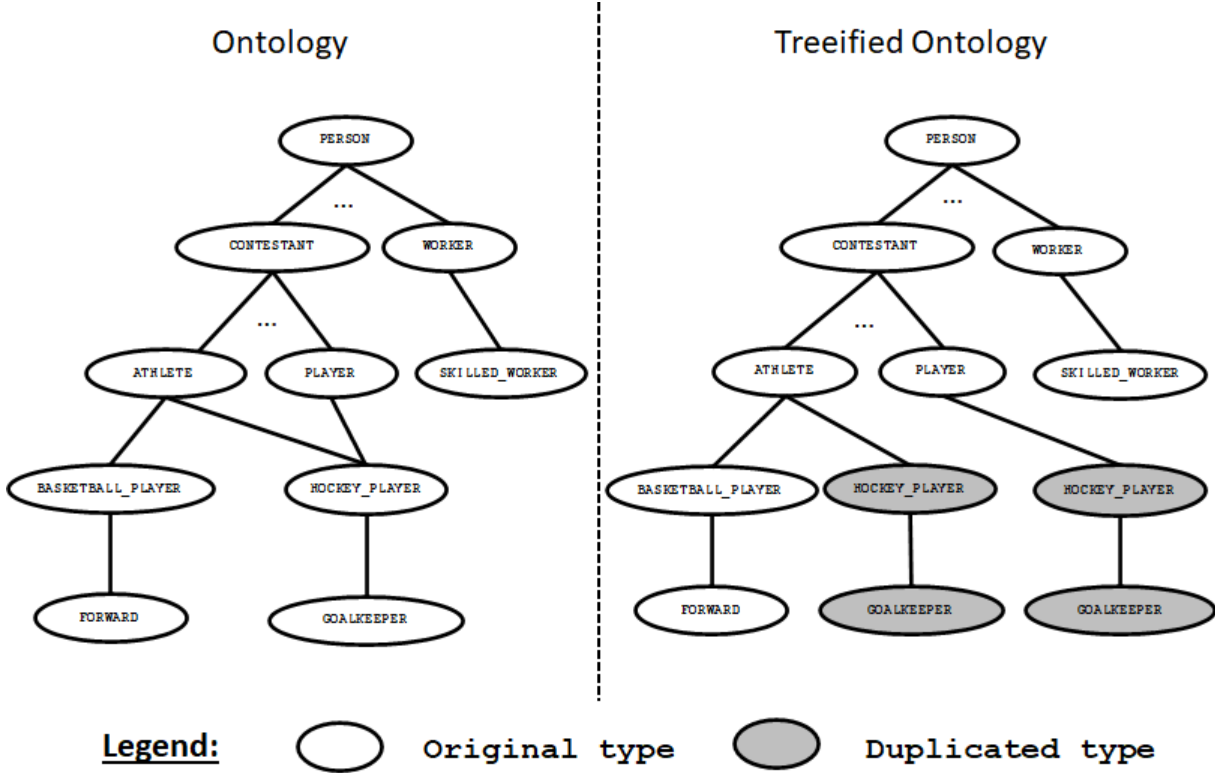


Figure 7.2: Ontology Treeification

Let $\hat{\tau}_i$ be the set of all the successor types of τ_i and itself. Then, we compute for each type τ_i its probability, defined as:

$$P(\tau_i) = \frac{\sum_{\tau \in \hat{\tau}_i} \text{count}(\tau)}{N}$$

Here, N is the frequency of total types and $\text{count}(\tau)$ is frequency of type τ . Let $LCA(\tau_x, \tau_y)$ be the lowest common ancestor of types τ_x and τ_y , then SCS is:

$$SCS(\tau_x, \tau_y) = -\log P(LCA(\tau_x, \tau_y))$$

$$SCS(q, d) = \text{avg} \left(\max_{1 \leq n \leq j} SCS(q_{\tau_m}, d_{\tau_n}) \right), 1 \leq m \leq i$$

7.2 SEMANNOREX Interface

The SEMANNOREX system showcases semantic search via entity-types based on **Cosine Similarity**, **Semantic Pathlength** as well as **Semantic Content Similarity** on a corpus of Web news and Wikinews articles. Figure 7.3 depicts the different retrieval

Algorithm 3 Ontology Treeification

Input: Original Ontology ($\mathcal{T} = t_1, t_2, \dots, t_I$);
PARENTS(t) returns parents of node t ;
len(*PARENTS*(t)) returns number of parents of node t ;
CHILDREN(t) returns all the children of node t ;
CHILDADD(t, p) sets node p as one of the children of node t ;
REMOVE(t) deletes the subtree rooted at node t

Output: Treeified Ontology

```

1: for  $t_i \in \mathcal{T}$  do
2:   if  $\text{len}(\text{PARENTS}(t_i)) > 1$  then
3:     for  $p \in \text{PARENTS}(t_i)$  do
4:        $t_{i\_new} \leftarrow p + \text{"."} + t_i$ 
5:       CHILDADD( $p, t_{i\_new}$ )
6:       for  $child \in \text{CHILDREN}(t_i)$  do
7:          $t_{i\_new\_child} \leftarrow t_{i\_new} + \text{"."} + child$ 
8:         CHILDADD( $t_{i\_new}, t_{i\_new\_child}$ )
9:       end for
10:    end for
11:    REMOVE( $t_i$ )
12:  end if
13: end for
14: return  $\mathcal{T}$ 

```

strategies, which will be presented subsequently. The functionality video and the link to a Web interface can be found on the SEMANNOREX Website⁴⁷.

Cosine Similarity (Cosine)

Cosine Similarity serves as a “baseline” retrieval method. The user might experience a somewhat “extreme” system behavior whether the selected type is present in the document, or not. This is due to the fact, that type vectors of documents tend to be sparse and semantic dependencies such as parent-child or sibling relations can not be exploited for retrieval. As a result, both sample queries in Fig. 7.3 ③ do not return the document labeled by the grey types.

Semantic Pathlength (SemPath)

Semantic Pathlength aims at overcoming the above mentioned shortcomings, through capturing parent-child or sibling relations by considering the distance between the selected type(s) and its (their) best possible match(es) in the document(s). However, the main drawback now is that types in the upper part of the ontology by definition are relatively “close” to the remaining types. Thus, example query q_1 scores higher than q_2 in Fig. 7.3 ④, although all document types are in same branch of query q_2 while q_1 belongs to a different top-level type.

⁴⁷ <https://spaniol.users.greyc.fr/research/SEMANNOREX/>

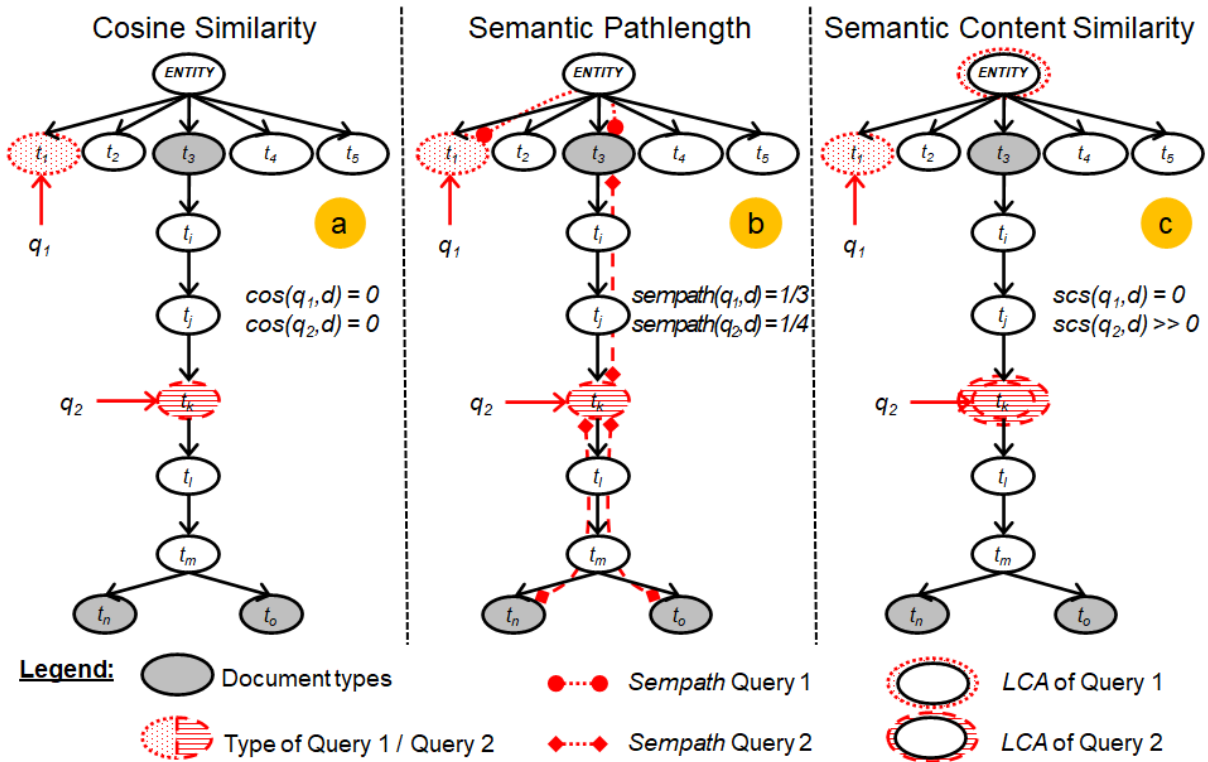


Figure 7.3: Comparison of the different Retrieval Methods

Semantic Content Similarity (SCS) of KB Types

Finally, Semantic Content Similarity (SCS) allows to exploit the semantics inherent in parent-child or sibling relations as well as putting “emphasis” on more specific types. To this end, the impact of an *LCA* type at the lower part of the ontology will be higher compared with an *LCA* type at a higher part of the ontology and, thus, leading to more concise search results. In the example of Fig. 7.3 ©, now, query q_1 does not return the document containing the grey types, because the *LCA* is the *root* node. In contrast, for q_2 the document will achieve a comparatively high score as the *LCA* of query and document is type t_k .

SEMANNOREX Search Interface

Figure 7.4 depicts the user interface of SEMANNOREX showing an example query for the types `computer_scientist`, `capitalist` and `event`. On the left hand side, the corpus (Wikinews or Web news) can be selected [at the bottom] and the treeified ontology can be explored [on top]. From the ontology representation one or more types of interest can be selected. The search results are retrieved and ranked accordingly in the main panel of the interface. In this example, the results are shown for the Semantic Content Similarity (*SCS*) method. In order to allow the user an intuition about the linked content, its title and the scores per selected type are provided. Further, the buttons on top allow the user to alter the utilized scoring method. Thus, the user is able to assess and compare the

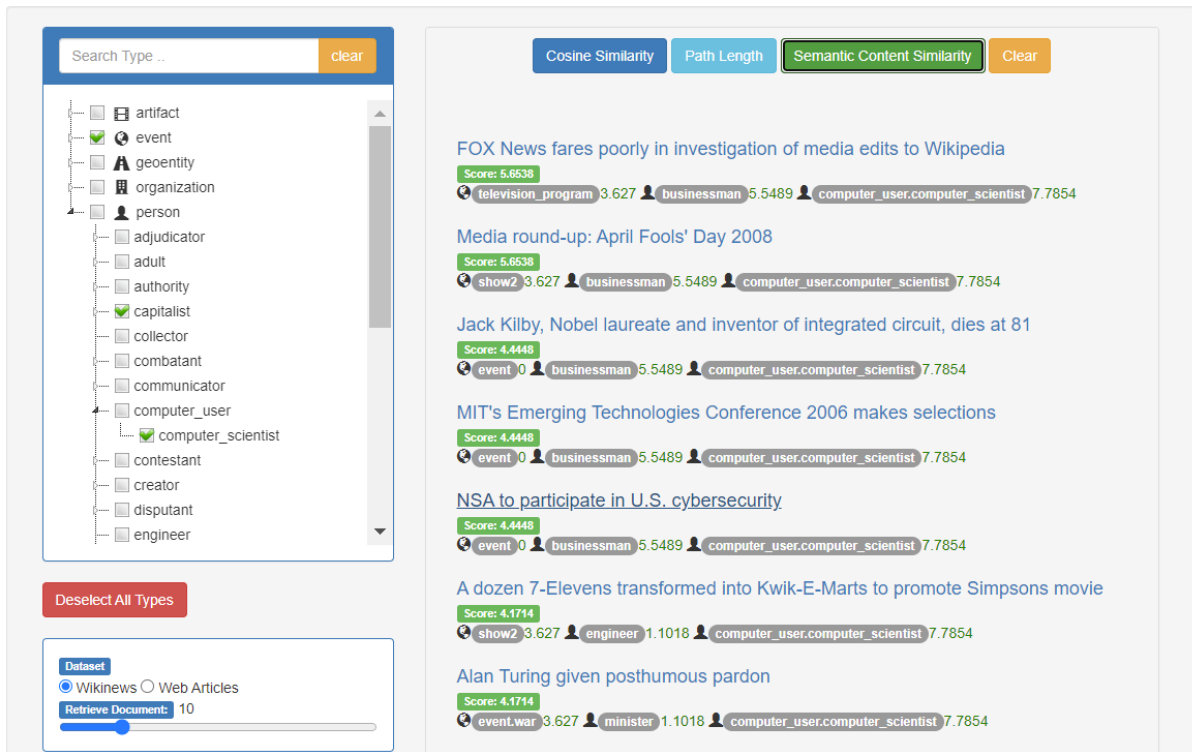


Figure 7.4: SEMANNOREX Search Interface displaying Results based on the Semantic Content Similarity (SCS) Method

relevance of the documents listed with respect to the individual types as well as based on the underlying scoring method.

Evaluation

The system corpus exists of more than 22,000 Web news and Wikinews articles. Table 7.1 summarizes the findings mentioned above conducted on 50 manually assessed queries each on Web news as well as on Wikinews articles. These queries range from 1 to 5 randomly chosen entity type(s), thus, emulating search behavior of various complexity. In order to ensure comparability, 10 queries have been constructed for each “level” (i.e. 10 queries with one type, two types, etc.). It can be observed from Table 7.1 that *SCS* ensures a balance between scarcity and information overload by simultaneously achieving the highest quality in terms of Prec@5 and MRR.

Method	Quantitative				Qualitative	
	Min	Max	Avg.	Median	Prec@5	MRR
Cosine	0	6,629	511.71	118.5	0.499	0.558
SemPath	3,662	18,929	11,295.5	11,295.5	0.590	0.711
SCS	1,417	18,903	8,653.47	5,281	0.641	0.771

Table 7.1: Quantitative and Qualitative Evaluation

In addition, we present the analysis of a sensitivity study in Table 7.2. It can be observed that the results for Cosine are somewhat extreme: queries with few entity types (one or two) lead to highly concise results (in case they exist), while a decay in quality can be observed for queries with more entity types. This observed decay can be dampened by the two other methods incorporating the underlying ontology structure (SemPath and *SCS*). Here, *SCS* is overall performing better. This is primarily caused by the fact that SemPath does establish links to all documents in the corpus (cf. quantitative analysis of Table 7.2) and, thus, also retrieves documents that are conceptually quite dissimilar. In contrast, *SCS* is more focused and retrieves only those documents that belong to the same top-level type. As a result, the number of documents retrieved is less, but they are overall more relevant.

Method \ # of Types	# of Types	Quantitative				Qualitative	
		Min	Max	Avg.	Median	Prec@5	MRR
Cosine	1	0	2,565	463.6	25	0.707	0.695
	2	0	386	84.9	43	0.75	0.589
	3	3	995	210.6	98	0.554	0.675
	4	17	6,629	1,197.75	538.5	0.437	0.618
	5	51	3,542	601.7	402	0.165	0.214
SemPath	1	3,662	18,929	11,295.5	11,295.5	0.73	0.842
	2	3,662	18,929	11,295.5	11,295.5	0.642	0.77
	3	3,662	18,929	11,295.5	11,295.5	0.482	0.607
	4	3,662	18,929	11,295.5	11,295.5	0.632	0.721
	5	3,662	18,929	11,295.5	11,295.5	0.462	0.617
SCS	1	1,417	16,644	7,256.95	5,161	0.72	0.87
	2	1,656	18,025	8,868.8	5,897.5	0.682	0.837
	3	2,959	18,747	8,592.7	7,123.5	0.627	0.731
	4	2,959	18,478	9,115.15	7,112.5	0.686	0.854
	5	2,959	18,903	9,433.75	7,124.5	0.49	0.568

Table 7.2: Sensitivity Study

7.3 Findings on Semantic Search

In this chapter, we have presented SEMANNOREX, a novel tool for the semantic annotation, retrieval, and exploration of (textual) documents. The novelty arises from exploiting concise entity-level annotations for semantic retrieval. As a proof-of-concept implementation, we applied SEMANNOREX onto a news corpus collected from Websites and Wikinews. In addition, we performed quantitative and qualitative evaluations to investigate SEMANNOREX utility and performance. We observe that our system benefits from the concise tag information derived from the named entities and is able to successfully retrieve the relevant documents with respect to a query. The most pertinent Web documents retrieved confirm our hypothesis (cf. Hypothesis 4 in Chapter 1). We also

thoroughly analyzed by varying the number of query types in order to show the vulnerability of the system with respect to a search query. In particular, our experiments show that an increasing number of entity types in a query can cause a decay in the quality of search results. This issue arises due to the disjunctive nature of the underlying implementation. While standard approaches suffer heavily in such a scenario, *SCS* still shows promising results. It is worth mentioning here that the proposed approach is independent of a domain and, thus, can be applied to any specific type of hierarchy.

In general we have observed, that the inaccuracy of the NED system and the PURE framework might harm the overall quality of the retrieved results. However, the performance of the state-of-the-art NED system utilized [Nguyen et al., 2014] and the PURE framework (cf. Chapter 5, [Kumar et al., 2020]) are very well and, thus, negligible.

Chapter 8

Semantic User Interest Tracing via Entity-level Analytics

8.1	Approach and Contribution	90
8.2	CONCEPTUAL APPROACH	91
8.3	Sub-user Representation	92
8.4	USER INTEREST TRACING MODELS	93
8.4.1	Random Forest based Models	94
8.4.2	Graph Convolutional Networks Models	95
8.4.3	SUIT	96
8.5	EXPERIMENTAL EVALUATION	96
8.5.1	Experimental Setup	96
8.5.2	Model Configurations	98
8.5.3	Experimental Results	99
8.5.4	Sensitivity Analysis	101
8.5.5	Findings on User Interest Tracing	105

As mentioned in the previous chapters, not each and every piece of information is relevant to a user. In general, an average user is interested in a certain set of Web documents. To this end, we “turn-around” our investigations by addressing our research question “How to predict user behavior by exploiting the semantics of contents they are interested in?” (cf. RQ5 in Chapter 1). In this chapter, we address the task of user interest tracing based on individual’s semantic publishing/editing behavior. To this end, we introduce an approach towards Semantic User Interest Tracing called SUIT.

Even after three decades of World Wide Web, one can still realize the tremendous amount of proliferation of Web data being generated and, subsequently, being accessible to Web users. In particular, the Web 2.0 and its social networking services such as Twitter, Facebook, online discussion forums, or Wikipedia have created the so-called “prosumer” [Toffler, 1980]: a (Web) user that actively produces and consumes. As a

result, millions of new Web contents are being generated on a daily basis. However, not each and every piece of information is equally relevant to a user. In general, an average user is interested in a certain set of Web documents, only. To this end we postulate, that a user can be characterized by the concepts s/he is interested in. To be concise, we claim that the (semantic) concepts [categories] inherent in documents published and/or modified by a user can be utilized in order to allow the tracing of his/her interests (cf. Hypothesis 5 in Chapter 1). As such, we raise user tracing to the entity-level and offer a novel, purely semantic, and language-agnostic approach. While we consider the personalization of a user’s Web experience (in a general) as a positive thing, we also want to raise aware of the inherent privacy problems. This is due to the fact, that our approach is capable of effectively [through (semantic) concepts] and efficiently [via a relatively small amount of training data] identifying user interest traces.

8.1 Approach and Contribution

Nowadays, identification and tracing of users’ interests from social media platforms texts has become a significant research topic [Han and Lee, 2014]. Although, it is incredibly challenging to capture user interests without categorical information. Moreover, identification of an author for a given document has several applications in various domains such as information retrieval, bibliometrics, and plagiarism detection [Rexha et al., 2015, Rexha et al., 2016, Rexha et al., 2018]. The objective shared by mentioned research topics is the identification if a Web content can be associated with the publishing/editing behavior of a specific user. In this study, we therefore introduce Semantic User Interest Tracing (SUIT in short), which aims at exploiting the (semantic) concepts [categories] inherent in documents in order to identify the user “behind” the content. To this end, SUIT identifies the concepts associated with a user in order to trace and - ultimately - reveal the publishing/editing pattern. Therefore, SUIT utilizes the inherent structure and relationships among the (semantic) concepts derived from a knowledge graph in order to identify and reveal the respective/individual user interests.

To this end, we investigate the concerned concepts based on editing behavior of newly generated or published Web documents from Web users. We employ a novel graph convolutional network architecture (GCN) to capture the inherent characteristics among the concepts extracted from a knowledge base (KB) in order to distill the user publishing/editing patterns. We perform our experiments in multiple languages, *i.e.*, English, German, and French. In particular, we utilize the Wikipedia articles published/edited by the Wikipedia user community. Extensive experiments on a multi-language dataset demonstrate the viability of our proposed approach. Furthermore, enhanced performance over all the mentioned languages confirms our hypothesis that our purely semantic approach can be accommodated for any of the languages.

In summary, the salient contributions of this study are:

- a language agnostic semantic user interest tracing and prediction model;
- the creation of a user interest tracing dataset based on the publishing behavior of Wikipedia editors;

- the adaptation of a GCN architecture in order to identify the structural patterns present among the (semantic) concepts linked with the different users;
- a comprehensive experimental study in multiple languages (English, German, and French languages) on semantic user interest tracing demonstrating the superior quality of our approach over state-of-the-art implementations and revealing a potential privacy intrusion.

8.2 CONCEPTUAL APPROACH

For the user’s interests tracing task, we propose a methodology that receives a set of documents for different users as input and predicts those documents’ potential candidate users/authors as output based on identified user publishing/editing patterns. As such, we provide a prediction module whether a document is likely to be edited by a specific user entirely based on concepts [categories] of the document. Let u be the set of users (cf. Equation 8.1) and d be the set of documents (cf. Equation 8.2). u_{i_d} represents the set of documents associated with user u_i as shown in Equation 8.3.

$$u = \{u_1, u_2, \dots, u_I\} \quad (8.1)$$

$$d = \{d_1, d_2, \dots, d_P\} \quad (8.2)$$

$$u_{i_d} = \{d_{i_1}, d_{i_2}, \dots, d_{i_N}\}, i \in [1, I] \quad (8.3)$$

With the emergence of Linked Open Data (LOD), many documents have already been interlinked/classified via an underlying ontology (e.g., Wikipedia category structure). In contrast to the previously mentioned approaches (cf. Chapter 3), we exploit such an underlying ontology which has been extracted from the YAGO knowledge base [Mahdisoltani et al., 2013]. In particular, we utilize the WordNet category system underlying YAGO. Each unit of the category system is termed a “concept”. It is worth to mention here that the concepts within the WordNet category system form a directed acyclic graph (DAG). It entails that a concept might be associated with more than one super concept. The semantic user interest tracing task consists of two building blocks: sub-user representation (cf. Section 8.3) and user interest tracing model (cf. Section 8.4). The sub-user representation is needed in order to serve as a ground truth for our experiments later on in order to connect documents with users. Since an individual document’s semantic representation is comparatively sparse (around 5-10 concepts compared with around 70,000 concepts of the entire ontology), we construct an aggregated sub-user representation. As a result, we obtain a sub-user representation graph, which is a DAG that forms the backbone of our GCN based approaches. This representation, will subsequently serve as input for the different user tracing models. Let U_i denote the set of sub-users corresponding to user u_i and $u_{i_d}^j$ representing the set of documents associated with sub-user u_i^j as shown in Equations 8.4 and 8.5, respectively.

$$U_i = \{u_i^1, u_i^2, \dots, u_i^J\}, i \in [1, I] \quad (8.4)$$

$$u_{i_d}^j = \{d_{i_1}^j, d_{i_2}^j, \dots, d_{i_Q}^j\}, i \in [1, I], j \in [1, J] \quad (8.5)$$

Further, we define a function ϕ , which predicts if u_i^j is Doppelgänger of user u_i or not. A Doppelgänger represents a double or an apparition of an alive person in fiction or folklore. Equation 8.6 describes the formulation.

$$\phi(u_i^j, u_i) = \begin{cases} 1, & \text{if } u_i^j \text{ is Doppelgänger w.r.t. } u_i \\ 0, & \text{if } u_i^j \text{ is not Doppelgänger w.r.t. } u_i \end{cases} \quad (8.6)$$

$$\forall u_i^j \in U_i, i \in [1, I], j \in [1, J]$$

Given a document d_p , predict the potential candidate users/authors $\sigma(d_p)$, *i.e.*, the set of users/authors that are likely to publish/edit the corresponding document d_p . Formally, it can be defined as:

$$\sigma(d_p) = \{u_i, u_i \in u \mid \phi(u_i^j, u_i) = 1, d_p \in u_{i_d}^j\} \quad (8.7)$$

8.3 Sub-user Representation

In this section, we present the methodology for sub-user representation. Figure 8.1 depicts the conceptual approach of the sub-user representation. It implicates five generic steps for sub-user representation discussed as follows:

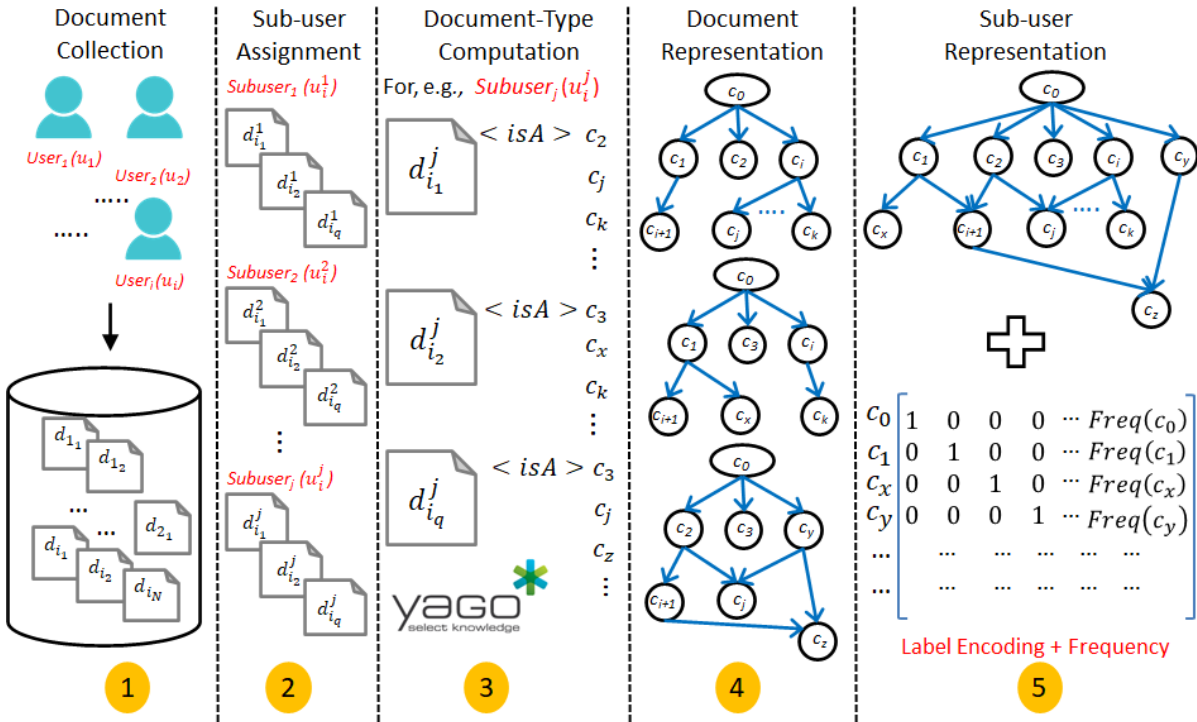


Figure 8.1: Conceptual Approach for Sub-user Representation

1) Document Collection

In a first step, we extract the documents published/edited from the Wikipedia revision history and create a separate list of documents associated with each user.

2) Sub-user Assignment

The second step involves the partition of a user into J sub-users. It means that the documents associated with a user are randomly split into J equal parts, and each part is assigned to a sub-user associated with the corresponding user. Thus, each of the sub-users is randomly assigned an equal number of distinguished documents from the retrieved documents associated with the respective users.

3) Document-Type Computation

For each document $d_{i_q}^j$ associated with sub-user u_i^j , we compute its associate concepts from the underlying ontology in the YAGO knowledge base. These concepts are called the direct concepts (③ of Figure 8.1).

4) Document Representation

In order to get the document representation, we first derive all the transitive concepts associated with the concepts computed in the previous step via the KB. Then, with the help of inherent hierarchical relationships present among the concepts (direct as well as transitive), we construct a graph. This graph consists of concepts as nodes and relationships among these concepts as edges. Again, this graph is directed and acyclic in nature. A separate graph is constructed for each of the documents as shown in ④ of Figure 8.1.

5) Sub-user Representation

In the fifth and last step, all the derived concepts in the previous steps (*i.e.*, concepts as well as their transitives) along with their semantic relationships are combined. They form a larger directed acyclic graph. In addition to the graph, we also define concept label encoding (cf. Section 8.4.2) and frequency of the respective concept (cf. Section 8.4.3) as illustrated in ⑤ of Figure 8.1. We construct an individual graph for each of the sub-users and call it sub-user representation graph. This way, each sub-user is represented by a directed acyclic graph.

Once the sub-user representation graph is constructed, it is provided as input to the next building block, *i.e.*, to the user interest tracing model. The model then aims at identifying the user publishing/editing patterns based on the sub-user representation graph. The Web documents utilized in the construction of the sub-user representation are regarded as the documents likely to be published/edited by the user who predicts the sub-user as a Doppelgänger.

8.4 USER INTEREST TRACING MODELS

In this section, we introduce and explain various user interest tracing models. We develop two machine learning models based on random forests. In order to learn the semantic characteristics, we also develop two graph convolutional network based models.

8.4.1 Random Forest based Models

In a first step, we employ a random forest as a learner to predict the user publishing/editing patterns. Random forest (*RF*) is a classifier based on ensemble learning techniques, which utilizes a collection of several decision trees to reduce the training error [Breiman, 2001]. The model attempts to enhance generalization by employing bootstrap aggregation over the training data and random subspace over the features.

Since a Web document can be edited by multiple users, we employ a multi-label classification technique. To this end, we convert the user interest tracing task into a set of sub-tasks and exploit the one-against-all scheme to solve the sub-tasks. Following the scheme, we train an individual classifier for each of the $|u|$ users. We follow the bag-of-words technique for the feature set construction and call it “bag-of-concepts”. The size of a feature vector for a test instance depends on all the possible concepts in that particular approach. In order to encode a sub-user representation graph, we insert a “1” in the feature vector at the corresponding locations of concepts present in the graph and rest of the entries are set to “0”. Once the feature vectors for each of the sub-users are encoded, a separate *RF* classifier is trained for each of the individual users using the one-against-all scheme. Here each classifier decides if the test instance sub-user is a Doppelgänger or not. for the tested sub-user. In the end, the decisions of all the classifiers are combined to conclude the decision about the tested sub-user.

Direct Concepts (σ_{Dir})

In our first approach, we attempt to solve the task by considering the directly connected concepts for all the Web documents associated with a sub-user as the features (cf. ③ in Section 8.3). Leaf concepts, are called the direct concepts. All the direct concepts from all documents associated with the train sub-users are defined as all the possible concepts in this approach. The size of a feature vector is the count of all the possible direct concepts. For example - $c_2, c_{i+1}, c_j, c_k, c_3, c_x, c_z$ are the direct concepts of the example in Figure 8.1 ③ and ④. We derive the feature vector for an instance using direct concepts and as discussed in the above section. The key idea behind this approach is that a Web user can be individualized by the concepts they are interested in. So, documents related to those concepts are more likely to be published/edited by that user.

Transitive Concepts (σ_{Trans})

The previously described *RF* based model (σ_{Dir}) considers only the direct concepts for the Web documents associated with a sub-user. It does not consider the other concepts which are related to those direct concepts and do not appear in the list of direct concepts for the respective sub-users. For instance, if some user is interested in topics related to (sub-)concept VICE_PRESIDENT then it is most likely that the user will also be interested in topics related to the more generic concept PRESIDENT.

In order to address the scenario described before, we compute all the transitive closures associated with the direct concepts for all the documents corresponding to some sub-user and utilize them as the feature set. All the transitive closures concepts and their respective direct concepts for all the sub-users in the training set are defined as all the possible concepts. Again, we derive the feature vector for a sub-user using all the direct

as well as transitive closure concepts associated with all the documents corresponding to that sub-user.

The fundamental limitation of the random forest based models is that they are not efficient enough to learn the inherent semantic patterns present among the concepts through the hierarchical relationships. Furthermore, the representative feature vector for the sub-user is not very informative in nature due to its sparsity.

8.4.2 Graph Convolutional Networks Models

In order to overcome the constraints of *RF* based models as pointed out in the previous section, we introduce graph convolutional network (GCN) as models for the user interest tracing task. At first, we adapt the GCN architecture for user publishing/editing pattern prediction and propose the underlying framework. The basic prediction model is represented by σ_{GCN} . We also propose an increment over the basic GCN framework represented by σ_{SUIT} in the following section by incorporating frequency information. The GCN models aim at learning the patterns by employing the sub-user representation graph and its associated concepts. More specifically, these models exploit the inherent semantics relationship among the concepts derived from the YAGO KB.

Graph Convolutional Networks

A GCN is a multi-layered neural network architecture that precisely operates on a graph designed dataset and generates embedding vector associated with each node of the graph [Kipf and Welling, 2017]. These embedding vectors are based on the attributes of the direct neighbors of the nodes.

In order to accommodate GCN for the user interest tracing task, we exploit the direct and all the transitive concepts used in a sub-user representation graph (cf. ⑤ in Fig. 8.1). We utilize the concept label encoding strategy in order to define the initial feature vector of each concept c as discussed in the following subsection. Further, we design a two-layer GCN architecture which is followed by a linear and a sigmoid layer. The architecture is nourished with the sub-user representation graph along with the concept label encoding associated with the concept present in the graph. This network shares the information among the nodes which are maximum two hops far from each other. We utilize the same aggregate function as advised in [Kipf and Welling, 2017]. Multiple kinds of readout operations have been discussed in [Wu et al., 2019] to get the graph level representation. We utilize the arithmetic mean of all the nodes as the readout operation to get the sub-user representation. Our observations based on the initial experiments show that a two-layer GCN accomplishes better results in comparison with a single-layer GCN. Further, incorporating more layers did not help in improving the prediction performance.

Concept Label Encoding

The user interest tracing model based on GCN receives the sub-user representation graph as input in order to identify the Doppelgänger. This graph is a directed acyclic graph where each node represents one of the concepts. In order to get a better understanding of the similarity among the sub-users, providing a GCN only with the graph structure is not sufficient. Thus, we define a label encoding for each distinguished node, *i.e.*, concept. To

this end, we utilize a one-hot encoding scheme. For example, while conducting experiments for the English language, we create a list of all the concepts present in the training set. Let $|c|$ be the total number of different concepts identified in the previous step. Then, we define a vector of dimension $|c|$ for each of the concepts where each position of the vector corresponds to one of the concepts. The entries of the concept vector are initialized with value “0” except only one position set to “1” (one-hot vector). Consequently, each concept is encoded with a one-hot vector with a value of “1” at the respective position in the defined vector.

We provide the sub-user representation graph and the concept vector for each node of the graph as an input. This matrix acts as the initial feature matrix for the basic GCN model. As it can be observed, the basic GCN model provides equal weight to each concept published/edited by a sub-user. However, it may happen that a sub-user has published/edited some concepts more than once, which is lost in the current configuration.

8.4.3 SUIT

The σ_{SUIT} model attempts to address the shortcomings of the above-mentioned graph convolutional network model (σ_{GCN}) by incorporating the frequency information. As mentioned, σ_{GCN} does not grant any kind of weight to the concept. We derive the weight of a concept through the appearance frequency of the respective concept in the sub-user representation. We incorporate the frequency information in the one-hot concept vector by integrating an additional dimension (cf. ⑤ in Figure 8.1). The last column of the feature matrix now represents the frequency of the respective concepts. Thus, σ_{SUIT} provides the label as well as the weight information for each of the concepts within the sub-user representation graph.

8.5 EXPERIMENTAL EVALUATION

We now explain the experimental settings. To this end, we introduce the experimental setup before presenting the experimental data set and results. We also present a sensitivity study and our findings.

8.5.1 Experimental Setup

The task of user interest tracing aims at identifying the same users based on their semantic representation of publishing/editing behavior towards the Web documents. In particular, we focus on the exploitation of the concepts of the Web documents to derive the Doppelgänger users. We develop various models in English, German, and French. For conducting the experiments, we need two pieces of information: a set of users along with their published/edited Web documents. Due to the availability of an ample amount of concepts associated with every document, we settle for a realistic setting to make it more impactful since the other category structure like Wikipedia Category System is noisy and not handled systematically. Therefore, we exploit the WordNet concepts for the documents as mentioned in YAGO [Mahdisoltani et al., 2013], totaling 68,423.

Data Set Extraction

In order to perform the experiments, we aim at inspecting the set of users along with their published/edited Web documents. One, if not the most paramount source for this sort of information is Wikipedia. For our experiments, we utilize a subset of the Wikipedia encyclopedia and its associated user community. More precisely, we extracted all the Wikipedians (users) from the European Union⁴⁸ using the available category members identifier Wikipedia API⁴⁹. Then, we extracted each user’s contributions in English Wikipedia by exploiting the revision history of the users. To this end, we utilize the user contribution Wikipedia API⁵⁰ for the retrieval of user revision history. It is worth mentioning that we focus on the main Wikipedia articles edited by a user for the experiments. For the same users, we extracted their revision history for French and German versions of Wikipedia, as well. In the current experiments, we utilize revision history as of March 23, 2021, for English and June 1, 2021, for German and French versions of Wikipedia. Not surprisingly, it is observable that the European users interested in the English version of Wikipedia aren’t necessarily interested in other versions of Wikipedia. It is because English is more globalized and contains a massive amount of documents compared to the other languages. Table 8.1 represents the statistics about different users along with their contributions in the Wikipedia articles, which validates our observation.

Language	#Users	#Average Articles Edited	#Median Articles Edited
English	5400	307.35	18
German	2097	253.59	5
French	1125	242.37	4

Table 8.1: Statistics of Edited Articles

Evaluation Dataset

Since there is no proper annotated dataset available for this task, we follow the approach mentioned in [Johansson et al., 2015, Chatzakou et al., 2020] and adapt it. For that purpose, we split a user into J sub-users and randomly assign an equal number of distinguished documents to each sub-user (cf. Section 8.3). For example - for $|u|$ number of users, we have $(J * |u|)$ sub-users in the dataset. The value of J is set to 100 in the current scenario. Each sub-user, along with its direct, transitive, and/or inherent semantic relationships among the concepts, are given as input to the respective models. We randomly split each user in (80 : 20) out of the 100 assigned sub-users. We utilize 80% and 20% of the datasets for training and testing purposes. We repeat the same steps for all the languages. In order to predict user publishing/editing patterns, we performed extensive experiments with different thresholds of documents and settled ourselves to the users who have published text in at least 100 different documents, since less number of

⁴⁸ https://en.wikipedia.org/wiki/Category:Wikipedians_in_the_European_Union

⁴⁹ Category Member: <https://www.mediawiki.org/wiki/API:Categorymembers>

⁵⁰ User Contribution: <https://www.mediawiki.org/wiki/API:Usercontribs>

English			
Threshold	#Users	Train Set	Test Set
100	1,345	805,919	267,431
200	901	791,179	259,720
300	695	778,299	253,259
400	568	766,606	247,621
500	467	753,451	241,518
600	412	744,367	237,377
700	351	730,506	231,577
800	322	723,419	228,457
900	297	716,804	225,449
1000	269	707,555	221,609

Table 8.2: English - Statistics of Users, Train, and Test Set

German			
Threshold	#Users	Train Set	Test Set
100	281	309,203	93,583
200	215	305,687	92,181
300	170	301,253	90,389
400	147	297,664	89,062
500	126	293,341	87,513
600	121	292,163	87,097
700	111	289,493	86,068
800	102	286,454	84,954
900	95	283,720	83,961
1000	83	278,841	82,083

Table 8.3: German - Statistics of Users, Train, and Test Set

edited documents are not enough to generate the patterns. We report the detailed results for document thresholds of 500 and 1000 in Tables 8.5 - 8.8 (cf. Section 8.5.3) and left the other thresholds for the sensitivity study (cf. Section 8.5.4). The statistics for different document thresholds, along with the number of users and documents associated with the train and the test set, are shown in Tables 8.2 - 8.4 for English, German, and French languages, respectively. For the sake of reproducibility, the dataset has been made publicly accessible via the project page⁵¹.

8.5.2 Model Configurations

For the random forest based models, we utilize the Scikit-learn library⁵². The bootstrap sample aggregation of the training data and the gini impurity measure to quantify the

⁵¹ SUI Dataset: <https://spaniol.users.greyc.fr/research/SUIT/SUIT.zip>

⁵² <https://scikit-learn.org/>

Threshold	French		
	#Users	Train Set	Test Set
100	133	171,773	49,461
200	101	169,475	48,714
300	80	166,647	47,778
400	71	165,127	47,262
500	65	163,721	46,796
600	60	162,581	46,362
700	56	161,433	45,910
800	51	159,568	45,266
900	50	159,041	45,110
1000	48	158,120	44,784

Table 8.4: French - Statistics of Users, Train, and Test Set

quality of a split have been exploited for the training of random forest based models. The number of decision trees in the forest is 100. We implement the GCN based models by exploiting PyTorch⁵³ and DGL⁵⁴ libraries. The pre-sigmoid logits are attained by operating the two hidden layers of convolution followed by a linear layer. A geometric pyramid rule [Masters, 1993] assigns the number of neurons in the respective convolutional layers. Both the GCN based models are trained by employing Adam optimization technique. The learning rate and the number of epochs are 0.001 and 100, respectively. After performing several experiments with different settings, we identified the aforementioned configurations as best performing due to their better generalization.

8.5.3 Experimental Results

We conduct a wide range of experiments based on the previously discussed experimental settings. We develop several models for all three languages based on approaches described in Section 8.4. In order to show the difficulty of the task, we develop a “naive” baseline and call it random (σ_{Rand}). This method randomly selects a user from a set of candidate users and assigns it to a document. Additionally, we also develop a baseline model (σ_{Base}). More specifically, (σ_{Base}) is based on “supervised authorship attribution” problem as in [Afroz et al., 2014]. It solves the problem by employing stylometric features and a Support Vector Machine as a classifier. We include all the features except the “leetspeak” since our experiments are based on Wikipedia articles, and leetspeak (or Internet slang) is very uncommon in Wikipedia.

We summarize the macro-averaged and the micro-averaged scores for document thresholds of 500 and 1000 in Tables 8.5 - 8.8. One can observe that both the GCN based approaches dominate the random forest as well as the baseline models with a larger margin. In particular, σ_{SUIT} outperforms the competitor models by a margin of around 7% to 16% in all the three languages for macro-averaged F-measure score. σ_{SUIT} beats σ_{Base}

⁵³ <https://pytorch.org/>

⁵⁴ <https://www.dgl.ai/>

with a margin of at least 36% in macro-averaged F-score across all the languages. It can also be observed that baseline approaches have a very high precision value compared to recall value. This gap between measures is because these models are able to capture the patterns for the highly active users. On the contrary, they fail to do for the less active ones. This difference can be observable both in macro and micro average scores. The excelling performance of the GCN based models is accredited to their capability of better encoding of the structured inherent semantic among the concepts. In addition, σ_{SUIT} performs significantly high, because of its conceptual adaptation of giving more weight to the more significant concept. Following the same line of observation, the σ_{SUIT} model also outperforms the other competitor models in micro-averaged F-measure score with a margin of 3% to 7% across all the three languages. Here, the baseline model σ_{Base} is beaten by at least 25% in F-measure score. The micro-averaged score is relatively high compared to the macro-averaged one since some instances (more active users) are performing better, and the micro-averaged score is influenced by the documents associated with those active users. In contrast, the macro-averaged score treats each instance equally.

Dataset	Metrics	σ_{Rand}	σ_{Base}	σ_{Dir}	σ_{Trans}	σ_{GCN}	σ_{UPPER}
English	Precision	0.0027	0.5427	0.3317	0.5223	0.6762	0.8119
	Recall	0.0032	0.3328	0.1595	0.2834	0.6195	0.7898
	F-measure	0.0029	0.4126	0.2154	0.3674	0.6466	0.8007
German	Precision	0.0061	0.525	0.4916	0.68	0.7672	0.8552
	Recall	0.0086	0.3702	0.2416	0.4105	0.6637	0.8158
	F-measure	0.0071	0.4342	0.324	0.512	0.7117	0.835
French	Precision	0.0208	0.6071	0.7067	0.9538	0.8462	0.9118
	Recall	0.0322	0.4509	0.3571	0.5617	0.7601	0.8792
	F-measure	0.0253	0.5175	0.4745	0.7071	0.8008	0.8952

Table 8.5: Macro-average Scores for Document Threshold of 500

Dataset	Metrics	σ_{Rand}	σ_{Base}	σ_{Dir}	σ_{Trans}	σ_{GCN}	σ_{UPPER}
English	Precision	0.0028	0.5387	0.9986	0.9993	0.8756	0.9407
	Recall	0.0028	0.538	0.4902	0.6473	0.8517	0.9342
	F-measure	0.0028	0.5383	0.6576	0.7857	0.8635	0.9374
German	Precision	0.0055	0.6138	0.9998	0.9997	0.9225	0.9522
	Recall	0.0055	0.6113	0.5882	0.7479	0.8668	0.9395
	F-measure	0.0055	0.6125	0.7407	0.8557	0.8937	0.9458
French	Precision	0.0192	0.6842	0.999	1	0.9483	0.9682
	Recall	0.0192	0.6828	0.5832	0.8236	0.8996	0.9591
	F-measure	0.0192	0.6835	0.7365	0.9033	0.9233	0.9636

Table 8.6: Micro-average Scores for Document Threshold of 500

Dataset	Metrics	σ_{Rand}	σ_{Base}	σ_{Dir}	σ_{Trans}	σ_{GCN}	σ_{UPPER}
English	Precision	0.0021	0.5846	0.451	0.689	0.8082	0.919
	Recall	0.0033	0.3832	0.226	0.4044	0.7802	0.9049
	F-measure	0.0025	0.463	0.3011	0.5096	0.794	0.9119
German	Precision	0.0068	0.5535	0.6145	0.8313	0.8502	0.9184
	Recall	0.0116	0.4387	0.3341	0.5232	0.7859	0.8911
	F-measure	0.0085	0.4895	0.4328	0.6422	0.8168	0.9045
French	Precision	0.0179	0.6496	0.8112	0.9792	0.9182	0.9529
	Recall	0.025	0.52	0.4689	0.7124	0.84	0.9374
	F-measure	0.0209	0.5776	0.5943	0.8247	0.8773	0.9451

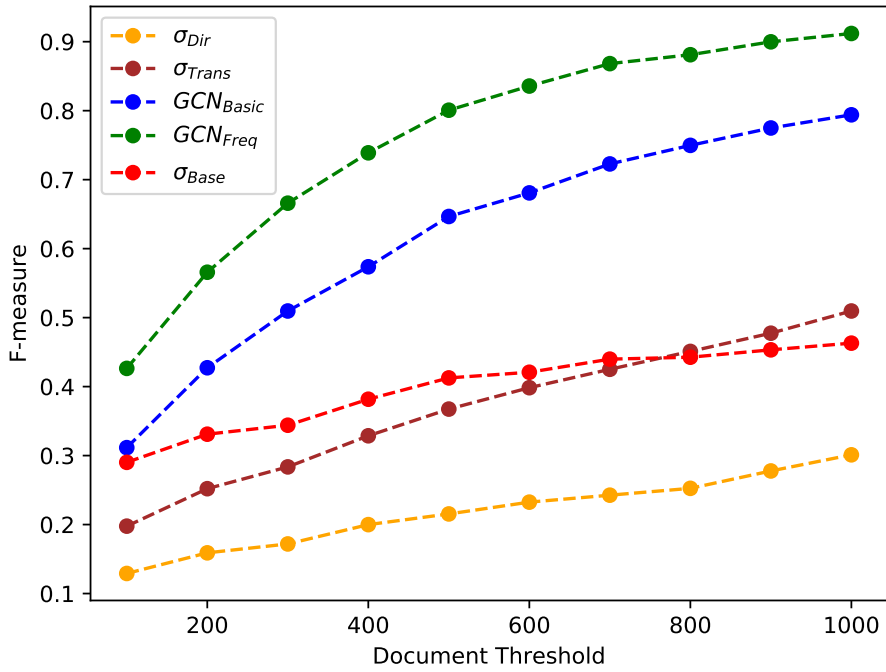
Table 8.7: Macro-average Scores for Document Threshold of 1000

Dataset	Metrics	σ_{Rand}	σ_{Base}	σ_{Dir}	σ_{Trans}	σ_{GCN}	σ_{UPPER}
English	Precision	0.002	0.5718	0.9989	0.9995	0.9161	0.9653
	Recall	0.002	0.5686	0.555	0.7104	0.9064	0.9617
	F-measure	0.002	0.5702	0.7135	0.8305	0.9112	0.9635
German	Precision	0.0083	0.6413	1	1	0.9415	0.9684
	Recall	0.0083	0.6389	0.6533	0.7956	0.9049	0.9606
	F-measure	0.0083	0.6401	0.7903	0.8862	0.9229	0.9645
French	Precision	0.0171	0.7128	0.999	1	0.9626	0.9703
	Recall	0.0171	0.7117	0.6373	0.8629	0.9109	0.9632
	F-measure	0.0171	0.7123	0.7782	0.9264	0.936	0.9668

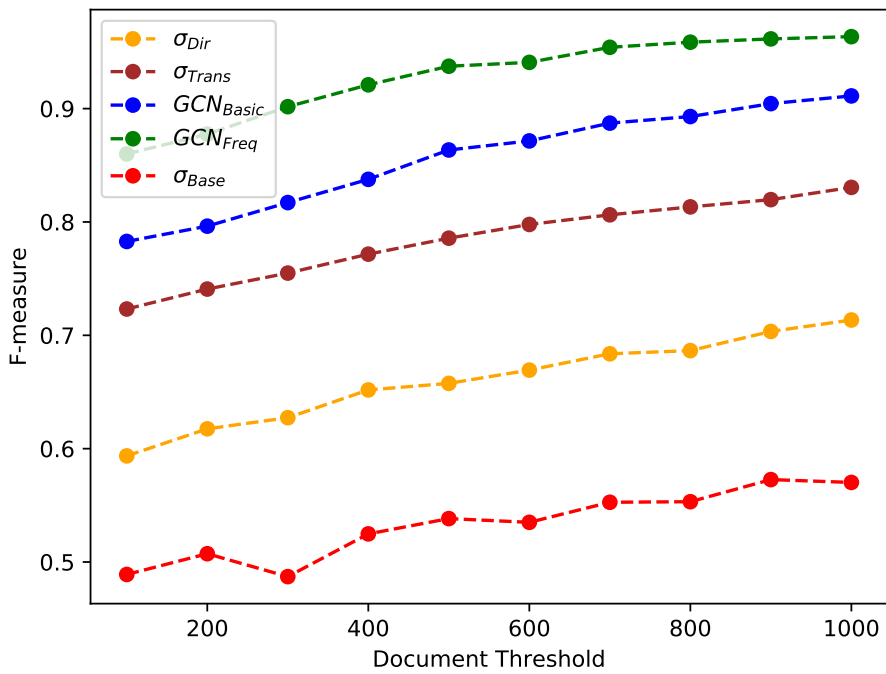
Table 8.8: Micro-average Scores for Document Threshold of 1000

8.5.4 Sensitivity Analysis

In addition to the previously reported results, we also present a sensitivity study based on the different document thresholds. In particular, we analyze the performance of different approaches by altering the document thresholds value from 100 to 1000. A document threshold of p means that we develop models only for those users who have published texts in at least p different documents. The statistics of users with different threshold are reported Tables 8.2 - 8.4. We illustrate the macro and micro averaged F-measure score with different document threshold across all the three languages in Figures 8.2 - 8.4. As we can observe, the performance of all the models is generally increasing with the increment of the threshold value. GCN based approaches (green and blue dotted lines) dominate the other approaches across all languages. This supports our hypothesis that GCN is capable of representing the better encoding of the inherent semantic among the concepts. The green dotted line at the top of each graph claims the superiority of σ_{SUIT} among all the models and supports the hypothesis that significant concepts deserve more priority. The same pattern can also be observed for the document thresholds of 500 and 1000 which are reported in Tables 8.5 - 8.8 (cf. Section 8.5.3). The increasing performance

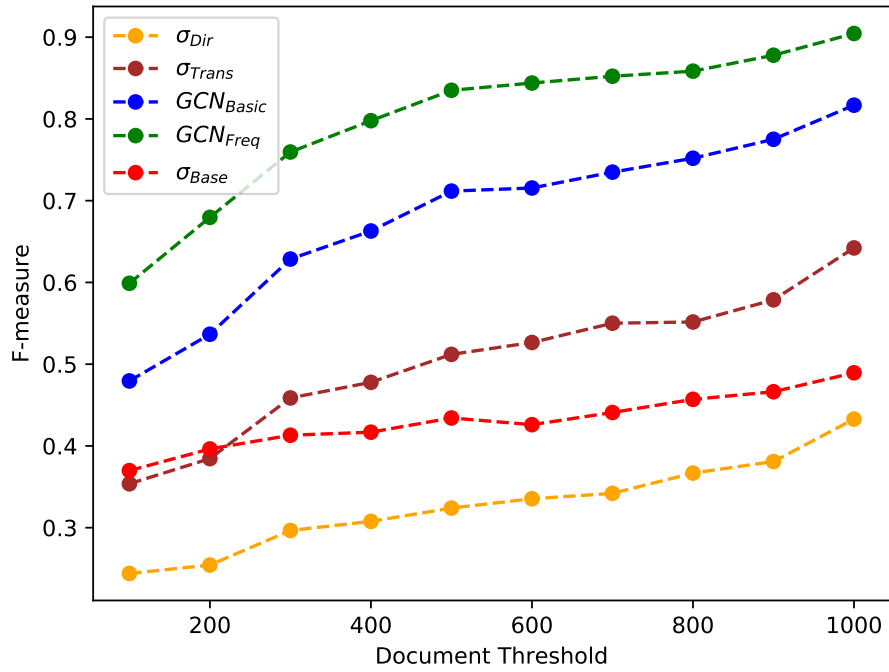


(a) Macro-score

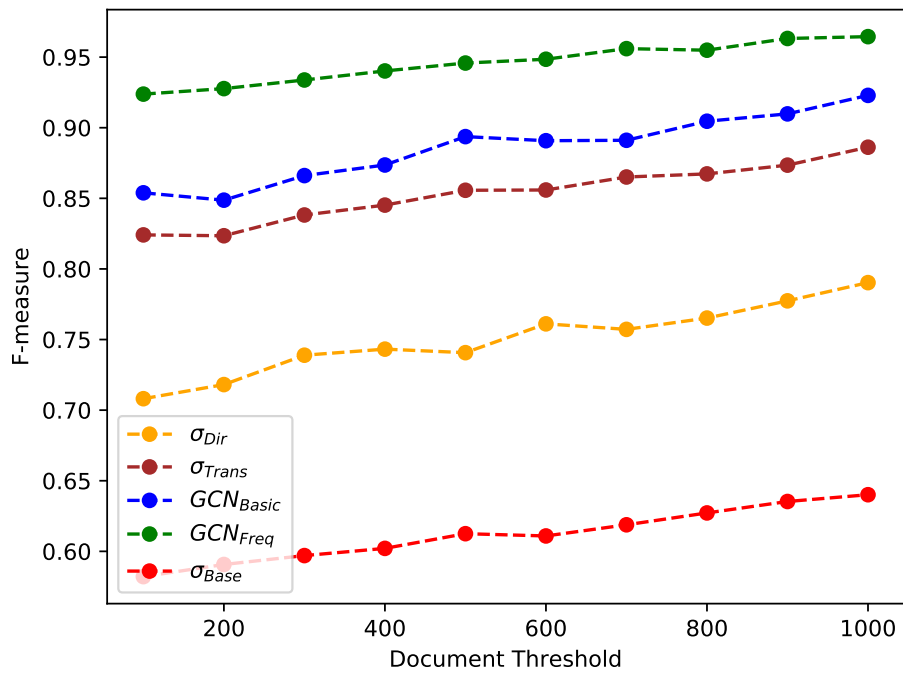


(b) Micro-score

Figure 8.2: Illustration of F-Measure Score in English Language

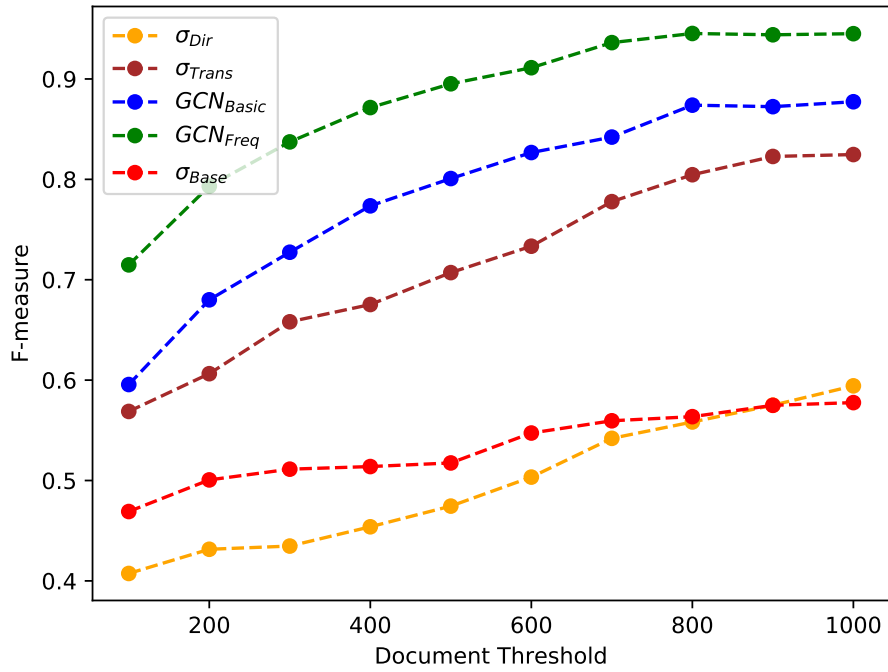


(a) Macro-score

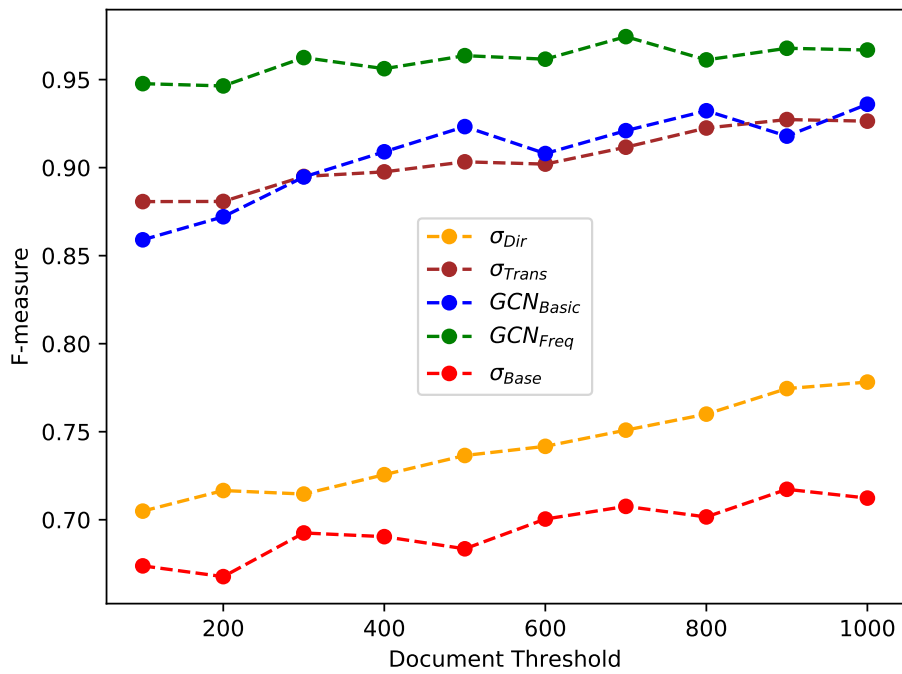


(b) Micro-score

Figure 8.3: Illustration of F-Measure Score in German Language



(a) Macro-score



(b) Micro-score

Figure 8.4: Illustration of F-Measure Score in French Language

of the models with the increment in the threshold value highlights that the patterns for the most active users (*i.e.*, users who are publishing texts frequently) are relatively easy to predict in comparison with the less active ones. This also enlightens that more coverage of concepts provides a better representation of the sub-user, and its versatility is more capable of identifying the patterns among the sub-users.

8.5.5 Findings on User Interest Tracing

In this chapter, we have presented SUIT, a novel methodology towards Semantic User Interest Tracing. To the best of our knowledge, this is the first ever approach aiming at predicting user publishing/editing behavior patterns purely via semantic tracing of edited Web documents derived from relationship of an entity’s KB concept graph. In order to address this problem, we conceptually adapted a GCN by defining concept label encodings and the node weight information within the concept relationship graph for a sub-user. We performed several experiments in multiple languages. In our comprehensive study, we have highlighted that SUIT significantly outperforms baseline approaches including a basic GCN implementation.

Our hypothesis that concepts are an excellent indicator to predict a user pattern was confirmed through extensive experiments in English as well as non-English languages (cf. Hypothesis 5 in Chapter 1). As discussed in Sections 8.2 - 8.4, the entire methodology is totally based on the derived concepts associated with a Web document. The outstanding results show the language-agnostic nature of the proposed approach. It entails that this methodology should be adapted for any language. In particular, the enhanced performance of the SUIT model for German and French demonstrate the viability of our method also for languages with less ample resources. As a result, user interest traces can be effectively and efficiently revealed. This raises serious risks of a potential privacy intrusion, particularly, because of the language-agnostic nature of our approach.

As our approach is purely semantic, it does not require linguistic features to predict user behavior. So, this semantic approach can further be exploited to predict user behavior based on visited Web contents. Moreover, we also noticed that incorporating the transitive concept information in prediction leads to a further improvement of the prediction model as reported in Tables 8.5 - 8.8. The reason for this behavior can be attributed to the fact that the direct concepts only provide very focused information, whereas integration of transitive concepts allows the model to learn more facets as well as is able to generalize concept dependencies.

Chapter 9

Conclusion and Outlook

9.1	Findings on Semantic Contextualization of Web content . . .	108
9.1.1	Findings on Entity-driven Content Analysis	108
9.1.2	Findings on Semantic Annotation and Retrieval	109
9.1.3	Findings on Semantic User Tracing	110
9.2	Future Research Directions	110
9.2.1	Entity Type Appearance in Events	111
9.2.2	(Dis)-information Spread Detection	111
9.2.3	Semantic-aware Privacy Protection	112

In this chapter, we summarize and conclude our findings from the research studies performed in the scope of this thesis. In addition, we also provide an outlook on the future research directions.

As mentioned in the previous chapters, our society realizes a tremendous amount of growth in Web data because of the advancement of the Web and the broad access to the Internet. The contextualization of information and its relevance is getting more and more challenging for an ordinary user. The key point is simple: “the more the merrier” is not always a good choice. This dissertation aims at getting a deeper insight into Web data in order to obtain a better semantic contextualization. In particular, we focus on entity-level analytics in order to enrich Web documents’ semantics via LOD. A noteworthy facet of all the research done as a part of this thesis is that the here presented methods are purely semantic, and thus, language-agnostic in nature. Our experiments show that entity-level analytics is an efficient technique in getting a deeper comprehensibility of Web documents and can be incorporated in various tasks (e.g., semantic search). The promising results of our extensive experiments in several studies have confirmed our formulated hypothesis (cf. Chapter 1).

9.1 Findings on Semantic Contextualization of Web content

This section highlights the significant findings of the research studies performed as part of this dissertation. In particular, we provide a critical reflection on the initial research questions (cf. Chapter 1).

9.1.1 Findings on Entity-driven Content Analysis

In order to get an initial contextualization of Web documents, we have started our study with semantic analysis and digestion of Web contents. In Chapter 4, we asked our first research question:

RQ1 How to explore a Web content with respect to named entities mentioned in it?

In order to answer this question, we hypothesized that named entities present in a Web document reveal its characteristics (cf. Hypothesis 1 in Chapter 1). To this end, we introduced a novel entity-based analytics framework CALVADOS, for the semantic distillation of Web contents. The system is an extension of “semantic fingerprinting”, which exploits entity-level analytics to distill a Web content and represents the overall semantics by a compact vector with information collected from LOD. We observed that the system is successfully able to capture the semantics of a Web content. Additionally, the system tackles the Web contents noise by employing aggregation over the named entity types. Further, we identified that the type information related to named entities mentioned in Web content conveys a substantial understanding of the document, confirming our hypothesis. However, named entities are predominantly annotated with multiple types without any order of importance associated. This creates turmoil in getting concise knowledge about document core semantics.

Thus, we realized that concise type(s) information about a named entity could reveal the document core semantics and as a consequence, might result in a better contextualization. This led to our next research question:

RQ2 Which facet(s) are the most expressive and representative for a named entity?

We addressed this issue in Chapter 5 and postulated the hypothesis: “named entities of certain types share a multitude of common and (at the same time) characteristic facets” (cf. Hypothesis 2 in Chapter 1). Therefore, we aimed at finding the most representative type(s) for a named entity and introduced the PURE framework. One of the notable contributions of our methodology is to rely solely on the structural information obtained from LOD. To this end, we performed a comprehensive study on named entities in Wikipedia. We conceptually adapted GCN and solely exploited the structured patterns present among the types derived from a knowledge graph in order to distill the most representative type(s) for a named entity. The accomplished experiments demonstrate very promising results. In particular, PURE outperforms competitors and achieves very good

results for the PERSON entity type, which is inherently highly complex and diverse. The PURE framework relies solely on the inherent structural patterns within an entity type graph without incorporating any additional information, such as the textual context of the entity, information about other related entities, etc. The resulting GCN based model has proven to efficiently and effectively capture the shared multitude of characteristic types among the entities by providing the representative type(s), confirming our hypothesis.

9.1.2 Findings on Semantic Annotation and Retrieval

After having been able to identify the most representative type(s) of a named entity, the most generic step is to utilize this information to annotate a document accordingly. So, we asked the question:

RQ3 How to concisely annotate a document and interlink it with concepts of LOD?

We focused on the above research question in Chapter 6. In particular, we utilized the obtained representative type(s) from the previous study for the concise semantic annotation of a document. We hypothesized that the PURE framework can help in getting the semantic characterization of a document (cf. Hypothesis 3 in Chapter 1). For that purpose, we have proven the viability of our proposed hypothesis by introducing AnnoTag, which has been demonstrated to be beneficial in concise document annotation. Moreover, the system has interlinked the generated semantic tags with LOD tags. One of the salient contributions of our approach is that it solely utilizes the entity type information to generate the tags. Based on the quantitative and the qualitative evaluations, we observed that entity-level analytics provides a high quality annotated document. This is due to the fact that PURE utilizes a relatively small number of concise tags for the document annotation. Thus, we suggest that a concise entity type document annotation strategy might become a valuable asset in digital curation. It is worth mentioning here, that the generated tags are machine (linked to the LOD cloud) as well as human interpretable. Moreover, the AnnoTag system allows an end-user to download the annotated document with or without entity information, which can be further utilized in other applications, such as semantic search, digital libraries, etc.

After addressing the above task, we moved a step further and investigated a use-case scenario of semantically annotated documents. In the subsequent study, we addressed the task of semantic search by asking the following question:

RQ4 Which retrieval method to apply in order to support ontology-driven retrieval?

To answer the above question, we hypothesized that entity-level analytics can be utilized to reveal the semantic characteristics of a query and a document (cf. Hypothesis 4 in Chapter 1). To this end, we introduced and compared systematically several similarity measure techniques in Chapter 7. For this purpose, we introduced the SEMANNOREX framework, which provides the end-user with an innovative semantically-driven access paradigm in order to explore a news corpus from Websites and Wikinews articles. In order to retrieve contents, SEMANNOREX supports the semantic search based on underlying

ontology. The uniqueness of our approach stems from entity-level analytics via PURE. By doing so, we observed that with a very compact representation, a highly concise contextualization of a Web document can be obtained. We noticed that SEMANNOREX is able to successfully retrieve the relevant documents with the help of concise tags derived from PURE. We have shown the SEMANNOREX system's effectiveness by performing quantitative and qualitative evaluations and a sensitivity study with respect to a varying number of types in a search query. Moreover, the proposed approach is scalable and can be adapted for any use-case based on a specific type hierarchy. We noticed that the inaccurate results of the NERD system and the PURE framework might influence the quality of search results. However, the encouraging results of the state-of-the-art NERD system utilized and the PURE framework have shown these concerns to be irrelevant in the given application scenario.

9.1.3 Findings on Semantic User Tracing

Finally, in our last study, we turned around our investigations. We focused on modeling user interests and asked the following research question:

RQ5 How to predict user behavior by exploiting the semantics of contents they are interested in?

In Chapter 8, we addressed the above research question. We hypothesized that user interests can be modeled by the concepts they are interested in (cf. Hypothesis 5 in Chapter 1). For this purpose, we introduced a novel semantic user interest tracing model, SUIT. The novel contribution of this work is that the SUIT framework is purely deriving its semantic representation from LOD. We noticed various benefits by employing semantic user tracings, such as a reduction in the number of features and a better understanding of user characteristics. In order to generate user patterns, the SUIT system solely exploits the concepts information of edited Web documents derived from a knowledge graph. We have shown the effectiveness of SUIT by conducting experiments on the Wikipedia user community. By performing extensive experiments in multiple languages (English, German, and French), we have shown the language-agnostic nature of the proposed methodology. The SUIT method outperforms state-of-the-art competitors as it is able to capture a semantic representation of a user successfully. The excellent performance in all the languages indicates that concepts convey significant information about a user character and that this pattern is totally independent of a language. Moreover, we noticed that aggregation of transitive concept information achieves further improvement. The reason is simple and instinctive: the direct concepts always show the fine-grained characteristics, while the transitive concepts help the model to learn more facets of the user character.

9.2 Future Research Directions

We believe that enriching semantics for the contextualization of Web contents via Linked Open data will remain an important research area. The research studies pursued in this

dissertation might encourage other researchers to perform studies in diverse directions. In this section, we provide an overview of several future directions.

9.2.1 Entity Type Appearance in Events

Our study observed that a Web document depicting an event contains explicitly recurring patterns of entity types appearing together. Further, if we got a deeper insight into the event, we noticed that the entity types appear sequentially for a particular event. For example - in the case of events of the kind “natural disaster” entities like `countries` or `organizations` appear first, while `politicians`, `presidents`, `agencies` or `banks` entities appear at later point of time. In the context of this thesis, we pursued the task of finding the most representative type(s) for a named entity (PURE, cf. Chapter 5). To this end, at a first step, entity type sequencing and then a novel entity-type emergence in the near future for an event can be predicted by employing entity-level analytics, in particular PURE. The subsequent investigation would be the “turn around” of the before-mentioned task. For some `person` entity, it would be interesting to find the list of the event types in which they can appear. At the same time, the temporal sequencing of events in which they appear is also important. By doing so, we can get the future aspects of an event and a person entity, which can be utilized to find the impact of an event in the different sectors, such as tourism, economics, international relations, etc., to name a few.

9.2.2 (Dis)-information Spread Detection

As discussed in the previous chapters, our society recently realizes an enormous amount of data being generated because of the development of the World Wide Web and wide access to the Internet. Thus, the dependency on the Web is growing very rapidly for the average user. As a consequence, the Web is an easy target for illegitimate or wrongful deception for personal and/or financial benefits. The proliferation of (dis)-information on the Web can be attributed to multiple prominent factors, such as online advertising revenue, tortious political influence, defame some people/organizations, etc. Recent global events (e.g., the COVID-19 outbreak) have witnessed loathsome influences due to the (dis)-information spread. Thus, the eminent important question is “how much one should rely on the integrity of the Web contents?”. As a subsequent generic step to the research studies conducted in this thesis, entity-level analytics and inter-connection with Linked Open Data might be explored in order to expose and - in a standard-setting - assist in analyzing/argue (dis)-information. In particular, the study based on semantic user tracing (cf. Chapter 8) can be further extended to identify the malign users who are creating the disinformation. Moreover, it might be anticipated that in a general setting, the user who is creating disinformation in one language could also spread disinformation in other languages. As the here presented approach (SUIT) is purely semantic and, thus, language-agnostic (in contrast to the previous approaches, cf. Chapter 3), it might serve in finding malign users in cross-language and cross-community settings (which is very common on the Web).

9.2.3 Semantic-aware Privacy Protection

The Web receives an ample amount of societal activities from everyday happenings. The central theme of the research conducted in the scope of this thesis is “to get a better contextualization of Web contents”. Our study on semantic user tracing (cf. Chapter 8) has shown that user preferences can be anticipated through semantic concepts based on the publish/edit history. This study has an obvious application scenario in the recommendation of Web contents. However, the study based on semantic concepts might be further exploited to trace also users across different communities in a negative (privacy intruding) way. “Even worse”, since our approach is conceptually language-agnostic, users could even be traced across different languages. As a result, this scenario creates a kind of “Big Brother” setting that heavily endangers the privacy of a user. In particular, a user might be traced solely based on his/her preferred concepts across communities and/or languages. Further, this information might be exploited against the very same user for some criminal activity, such as security threats, blackmailing, personal/organizational gain, etc. In order to avoid problems like the one mentioned before, studies on semantic-aware privacy protection should be undertaken in order to alert the user if his/her identity might become detectable solely through the observation of semantic concepts he/she is interested in.

List of Figures

2.1	Example of Named Entity Disambiguation for <code>Turing</code> and <code>computer science</code>	16
2.2	A Sample Text Snippet of Mention-entity Graph with Ambiguous Entities Mentioned from AIDA [Hoffart et al., 2011b]	18
2.3	Binary Classification Problem SVM Hyperplane with Margin	22
2.4	A Schematic Representation of Multi-layer GCN with First-order Filters	25
2.5	Schematic Depiction of GCN Batch-wise Graph Classification	25
2.6	Confusion Matrix: Layout for Performance Visualization	26
4.1	Conceptual Overview of the CALVADOS Pipeline	50
4.2	Semantic Digest of a Web Page	51
4.3	Semantic Comparison of Web Pages	52
5.1	Entity-type Graph Excerpt for <i>Donald Trump</i>	56
5.2	Conceptual Approach for Representative Type(s) Prediction of <i>Entities</i>	57
5.3	Illustration of the Entity Representative Type(s) Classification Pipeline for GCN based Approaches	60
6.1	Steps in Entity-driven Semantic Tagging	73
6.2	Excerpt of an Annotated Example Document	74
6.3	Steps of the AnnoTag Framework	77
7.1	Conceptual SEMANNOREX Pipeline	80
7.2	Ontology Treeification	82
7.3	Comparison of the different Retrieval Methods	84
7.4	SEMANNOREX Search Interface displaying Results based on the Semantic Content Similarity (SCS) Method	85
8.1	Conceptual Approach for Sub-user Representation	92
8.2	Illustration of F-Measure Score in English Language	102
8.3	Illustration of F-Measure Score in German Language	103
8.4	Illustration of F-Measure Score in French Language	104

List of Tables

2.1	Excerpt of an RDF Graph for Resource <code>Alan_Turing</code> in DBpedia; <code>dbr</code> and <code>dbo</code> , symbolize a DBpedia Resource and the DBpedia Ontology Schema, respectively	13
2.2	RDFS Class Definition - Excerpt of RDF Triples in DBpedia Ontology Schema	13
3.1	Comparative Study of Fine-grained Entity Type Classification	32
3.2	Summarization of Few Applications of GCN	37
3.3	A Comparative Illustration of Various Doppelgänger Detection Studies	43
3.4	A Comparative Illustration of Various Authorship Attribution Studies	45
5.1	Gold Standard (GS) Statistics	63
5.2	Regular Expressions for Candidate Text String Extraction (‘ ’ Represents ‘or’)	64
5.3	Experimental Dataset Statistics	64
5.4	Macro-average Scores for <i>Entity-centric</i> Evaluation	66
5.5	Micro-average Scores for <i>Entity-centric</i> Evaluation	67
5.6	Macro-average Scores for <i>Type-centric</i> Evaluation	67
5.7	Micro-average Scores for <i>Type-centric</i> Evaluation	68
6.1	Qualitative Assessment over 50 randomly sampled Documents	75
6.2	Quantitative Assessment over 50 randomly sampled Documents	76
7.1	Quantitative and Qualitative Evaluation	85
7.2	Sensitivity Study	86
8.1	Statistics of Edited Articles	97
8.2	English - Statistics of Users, Train, and Test Set	98
8.3	German - Statistics of Users, Train, and Test Set	98
8.4	French - Statistics of Users, Train, and Test Set	99
8.5	Macro-average Scores for Document Threshold of 500	100
8.6	Micro-average Scores for Document Threshold of 500	100
8.7	Macro-average Scores for Document Threshold of 1000	101
8.8	Micro-average Scores for Document Threshold of 1000	101

Bibliography

- [Abbasi and Chen, 2008] Abbasi, A. and Chen, H. (2008). Writeprints: A Stylometric Approach to Identity-Level Identification and Similarity Detection in Cyberspace. *ACM Trans. Inf. Syst.*, 26(2).
- [Afiontzi et al., 2013] Afiontzi, E., Kazadeis, G., Papachristopoulos, L., Sfakakis, M., Tsakonas, G., and Papatheodorou, C. (2013). Charting the Digital Library Evaluation Domain with a Semantically Enhanced Mining Methodology. In *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '13, pages 125–134.
- [Afroz et al., 2014] Afroz, S., Caliskan-Islam, A., Stolerman, A., Greenstadt, R., and McCoy, D. (2014). Doppelgänger Finder: Taking Stylometry to the Underground. *Proceedings - IEEE Symposium on Security and Privacy*, pages 212–226.
- [Auer et al., 2007] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). DBpedia: A Nucleus for a Web of Open Data. *The Semantic Web*, pages 722–735.
- [Ayadi et al., 2019] Ayadi, A., Zanni-Merk, C., de Bertrand de Beuvron, F., Thompson, J., and Krichen, S. (2019). BNO - An Ontology for Understanding the Transittability of Complex Biomolecular Networks. *Journal of Web Semantics*, 57:100495.
- [Balog and Neumayer, 2012] Balog, K. and Neumayer, R. (2012). Hierarchical Target Type Identification for Entity-Oriented Queries. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 2391–2394.
- [Banko et al., 2007] Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., and Etzioni, O. (2007). Open Information Extraction from the Web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, IJCAI'07, pages 2670–2676.
- [Bast et al., 2016] Bast, H., Buchhold, B., and Haussmann, E. (2016). Semantic Search on Text and Knowledge Bases. *Foundations and Trends® in Information Retrieval*, 10:119–271.
- [Bastings et al., 2017] Bastings, J., Titov, I., Aziz, W., Marcheggiani, D., and Sima'an, K. (2017). Graph Convolutional Encoders for Syntax-aware Neural Machine Translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1957–1967.

- [Benaouicha et al., 2015] Benaouicha, M., Taieb, M. A. H., and Hamadou, A. B. (2015). Taxonomy-based information content and wordnet-wiktionary-wikipedia glosses for semantic relatedness. *Applied Intelligence*, 45:475–511.
- [Berant et al., 2013] Berant, J., Chou, A., Frostig, R., and Liang, P. (2013). Semantic Parsing on Freebase from Question-Answer Pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544.
- [Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic Web. *Scientific american*, 284(5):34–43.
- [Bian et al., 2020] Bian, T., Xiao, X., Xu, T., Zhao, P., Huang, W., Rong, Y., and Huang, J. (2020). Rumor Detection on Social Media with Bi-Directional Graph Convolutional Networks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 549–556.
- [Bikakis et al., 2010] Bikakis, N., Giannopoulos, G., Dalamagas, T., and Sellis, T. (2010). Integrating Keywords and Semantics on Document Annotation and Search. In *Proceedings of the 2010 International Conference on On the Move to Meaningful Internet Systems: Part II, OTM’10*, pages 921–938.
- [Bird et al., 2009] Bird, S., Klein, E., and Loper, E. (2009). Natural Language Processing with Python. *O’Reilly Media, Inc.*, 1st edition.
- [Bizer et al., 2009a] Bizer, C., Heath, T., and Berners-Lee, T. (2009a). Linked Data - The Story So Far. *Int. J. Semantic Web Inf. Syst.*, 5:1–22.
- [Bizer et al., 2009b] Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., and Hellmann, S. (2009b). DBpedia - A Crystallization Point for the Web of Data. *Web Semant.*, 7(3):154–165.
- [Boenninghoff et al., 2019] Boenninghoff, B. T., Hessler, S., Kolossa, D., and Nickel, R. M. (2019). Explainable Authorship Verification in Social Media via Attention-based Similarity Learning. *2019 IEEE International Conference on Big Data (Big Data)*, pages 36–45.
- [Bollacker et al., 2008] Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD ’08*, pages 1247–1250.
- [Borthwick et al., 1998] Borthwick, A., Sterling, J., Agichtein, E., and Grishman, R. (1998). Exploiting Diverse Knowledge Sources via Maximum Entropy in Named Entity Recognition. In *Sixth Workshop on Very Large Corpora*.
- [Breiman, 2001] Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

-
- [Cano et al., 2013] Cano, A. E., Varga, A., Rowe, M., Ciravegna, F., and He, Y. (2013). Harnessing Linked Knowledge Sources for Topic Classification in Social Media. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, HT '13, pages 41–50.
- [Carlson et al., 2010] Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E. R., and Mitchell, T. M. (2010). Toward an Architecture for Never-Ending Language Learning. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, page 1306–1313. AAAI Press.
- [Chatzakou et al., 2017] Chatzakou, D., Kourtellis, N., Blackburn, J., De Cristofaro, E., Stringhini, G., and Vakali, A. (2017). Measuring #GamerGate: A Tale of Hate, Sexism, and Bullying. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, pages 1285–1290.
- [Chatzakou et al., 2020] Chatzakou, D., Soler-Company, J., Tsikrika, T., Wanner, L., Vrochidis, S., and Kompatsiaris, I. (2020). User Identity Linkage in Social Media Using Linguistic and Social Interaction Features. In *12th ACM Conference on Web Science*, WebSci '20, pages 295–304.
- [Chen et al., 2010] Chen, J., Nairn, R., Nelson, L., Bernstein, M., and Chi, E. (2010). Short and Tweet: Experiments on Recommending Content from Information Streams. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1185–1194.
- [Cheng et al., 2016] Cheng, D., Gong, Y., Zhou, S., Wang, J., and Zheng, N. (2016). Person Re-identification by Multi-Channel Parts-Based CNN with Improved Triplet Loss Function. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1335–1344.
- [Choi et al., 2018] Choi, E., Levy, O., Choi, Y., and Zettlemoyer, L. (2018). Ultra-Fine Entity Typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 87–96.
- [Ciampaglia et al., 2015] Ciampaglia, G. L., Shiralkar, P., Rocha, L. M., Bollen, J., Menczer, F., and Flammini, A. (2015). Computational Fact Checking from Knowledge Networks. *PLOS ONE*, 10(6):1–13.
- [Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 20(3):273–297.
- [Cristani et al., 2012] Cristani, M., Roffo, G., Segalin, C., Bazzani, L., Vinciarelli, A., and Murino, V. (2012). Conversationally-Inspired Stylometric Features for Authorship Attribution in Instant Messaging. In *Proceedings of the 20th ACM International Conference on Multimedia*, MM '12, pages 1121–1124.

- [Cucerzan, 2007] Cucerzan, S. (2007). Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 708–716.
- [Cui et al., 2020] Cui, Z., Henrickson, K. C., Ke, R., and Wang, Y. (2020). Traffic Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 21:4883–4894.
- [Dalton et al., 2014] Dalton, J., Dietz, L., and Allan, J. (2014). Entity Query Feature Expansion Using Knowledge Base Links. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR '14*, pages 365–374.
- [Defferrard et al., 2016] Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16*, pages 3844–3852.
- [Dong et al., 2021] Dong, S., Wang, P., and Abbas, K. (2021). A Survey on Deep Learning and its Applications. *Computer Science Review*, 40:100379.
- [Dong et al., 2014] Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., and Zhang, W. (2014). Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14*, pages 601–610.
- [Duan et al., 2017] Duan, Y., Jatowt, A., and Tanaka, K. (2017). Discovering Typical Histories of Entities by Multi-Timeline Summarization. In *Proceedings of the 28th ACM Conference on Hypertext and Social Media, HT '17*, pages 105–114.
- [Duvenaud et al., 2015] Duvenaud, D., Maclaurin, D., Aguilera-Iparraguirre, J., Gómez-Bombarelli, R., Hirzel, T., Aspuru-Guzik, A., and Adams, R. P. (2015). Convolutional Networks on Graphs for Learning Molecular Fingerprints. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2, NIPS'15*, pages 2224–2232.
- [Eagle and (Sandy) Pentland, 2006] Eagle, N. and (Sandy) Pentland, A. (2006). Reality Mining: Sensing Complex Social Systems. *Personal Ubiquitous Comput.*, 10(4):255–268.
- [Eckart de Castilho et al., 2016] Eckart de Castilho, R., Mújdricza-Maydt, É., Yimam, S. M., Hartmann, S., Gurevych, I., Frank, A., and Biemann, C. (2016). A Web-based Tool for the Integrated Annotation of Semantic and Syntactic Structures. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 76–84.

-
- [Eke et al., 2019] Eke, C. I., Norman, A. A., Shuib, L., and Nweke, H. F. (2019). A Survey of User Profiling: State-of-the-Art, Challenges, and Solutions. *IEEE Access*, 7:144907–144924.
- [El Bolock et al., 2020] El Bolock, A., Herbert, C., and Abdennadher, S. (2020). CCOnto: Towards an Ontology-Based Model for Character Computing. In *Research Challenges in Information Science*, pages 529–535.
- [El Ghosh et al., 2020] El Ghosh, M., Zanni-Merk, C., Delestre, N., Kotowicz, J.-P., and Abdulrab, H. (2020). Topic-OPA: A Topic Ontology for Modeling Topics of Old Press Articles. In *KEOD*, pages 275–282.
- [Elberrichi et al., 2008] Elberrichi, Z., Rahmoun, A., and Bentaallah, M. A. (2008). Using WordNet for Text Categorization. In *International Arab Journal of Information Technology*, vol. 5(1), pages 16–24.
- [Ernst et al., 2016] Ernst, P., Siu, A., Milchevski, D., Hoffart, J., and Weikum, G. (2016). DeepLife: An Entity-aware Search, Analytics and Exploration Platform for Health and Life Sciences. In *Proceedings of ACL-2016 System Demonstrations*, pages 19–24.
- [Fernquist et al., 2017] Fernquist, J., Fångström, T., and Kaati, L. (2017). IoT Data Profiles: The Routines of Your Life Reveals Who You Are. In *2017 European Intelligence and Security Informatics Conference (EISIC)*, pages 61–67.
- [Ferragina and Scaiella, 2010] Ferragina, P. and Scaiella, U. (2010). TAGME: On-the-fly Annotation of Short Text Fragments (by Wikipedia Entities). In *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010*, pages 1625–1628.
- [Finkel et al., 2005] Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370.
- [Flekova and Gurevych, 2016] Flekova, L. and Gurevych, I. (2016). Supersense Embeddings: A Unified Model for Supersense Interpretation, Prediction, and Utilization. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2029–2041.
- [Garigliotti et al., 2019] Garigliotti, D., Hasibi, F., and Balog, K. (2019). Identifying and Exploiting Target Entity Type Information for Ad Hoc Entity Retrieval. *Inf. Retr.*, 22(3–4):285–323.
- [Ghaddar and Langlais, 2018] Ghaddar, A. and Langlais, P. (2018). Transforming Wikipedia into a Large-Scale Fine-Grained Entity Type Corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, European Language Resources Association (ELRA).

- [Giannopoulos et al., 2010] Giannopoulos, G., Bikakis, N., Dalamagas, T., and Sellis, T. (2010). GoNTogle: A Tool for Semantic Annotation and Search. In *The Semantic Web: Research and Applications*, pages 376–380.
- [Godoy and Amandi, 2006] Godoy, D. and Amandi, A. (2006). Modeling User Interests by Conceptual Clustering. newblock In *Information Systems, Elsevier Science Ltd.*, vol. 31, pages 247–265.
- [Govind et al., 2018a] Govind, Alec, C., and Spaniol, M. (2018a). ELEVATE-Live: Assessment and visualization of online news virality via Entity-Level Analytics. In *Proceedings of the 18th International Conference on Web Engineering, ICWE 2018*, pages 482–486.
- [Govind et al., 2018b] Govind, Alec, C., and Spaniol, M. (2018b). Semantic Fingerprinting: A Novel Method for Entity-Level Content Classification. In *Proceedings of the 18th International Conference on Web Engineering, ICWE 2018*, pages 279–287.
- [Govind et al., 2019a] Govind, Alec, C., and Spaniol, M. (2019a). Fine-grained Web Content Classification via Entity-level Analytics: The Case of Semantic Fingerprinting. *Journal of Web Engineering*, 17:449–482.
- [Govind et al., 2019b] Govind, Kumar, A., Alec, C., and Spaniol, M. (2019b). CALVADOS: A Tool for the Semantic Analysis and Digestion of Web Contents. In *The Semantic Web: ESWC 2019 Satellite Events*, pages 84–89.
- [Govind and Spaniol, 2017] Govind and Spaniol, M. (2017). ELEVATE: A Framework for Entity-Level Event Diffusion Prediction into Foreign Language Communities. In *Proceedings of the 2017 ACM on Web Science Conference, WebSci '17*, pages 111–120.
- [Guha et al., 2003] Guha, R., McCool, R., and Miller, E. (2003). Semantic Search. In *Proceedings of the 12th International Conference on World Wide Web, WWW '03*, pages 700–709.
- [Gunaratna et al., 2016] Gunaratna, K., Thirunarayan, K., Sheth, A., and Cheng, G. (2016). Gleaning Types for Literals in RDF Triples with Application to Entity Summarization. In *The Semantic Web. Latest Advances and New Domains*, pages 85–100.
- [Gupta and Berberich, 2019] Gupta, D. and Berberich, K. (2019). Structured Search in Annotated Document Collections. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM '19*, pages 794–797.
- [Hamdan et al., 2013] Hamdan, H., Béchet, F., and Bellot, P. (2013). Experiments with DBpedia, WordNet and SentiWordNet as resources for sentiment analysis in microblogging. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 455–459.

-
- [Han and Lee, 2014] Han, J. and Lee, H. (2014). Characterizing User Interest Using Heterogeneous Media. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 Companion, pages 289–290.
- [Han and Lee, 2016] Han, J. and Lee, H. (2016). Characterizing the Interests of Social Media Users. *Inf. Sci.*, 358(C):112–128.
- [Han et al., 2019] Han, P., Yang, P., Zhao, P., Shang, S., Liu, Y., Zhou, J., Gao, X., and Kalnis, P. (2019). GCN-MF: Disease-Gene Association Identification By Graph Convolutional Networks and Matrix Factorization. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 705–713.
- [Han et al., 2011] Han, X., Sun, L., and Zhao, J. (2011). Collective Entity Linking in Web Text: A Graph-Based Method. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 765–774.
- [Hao et al., 2018] Hao, Y., Liu, H., He, S., Liu, K., and Zhao, J. (2018). Pattern-revising Enhanced Simple Question Answering over Knowledge Bases. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3272–3282.
- [Hasibi et al., 2017] Hasibi, F., Balog, K., Garigliotti, D., and Zhang, S. (2017). Nordlys: A Toolkit for Entity-Oriented and Semantic Search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, pages 1289–1292.
- [He et al., 2016] He, Q., Chen, B.-C., and Agarwal, D. (2016). Building The LinkedIn Knowledge Graph. <https://engineering.linkedin.com/blog/2016/10/building-the-linkedin-knowledge-graph>.
- [Ho, 1998] Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 20(8):832–844.
- [Ho et al., 2020] Ho, V. T., Pal, K., Kleer, N., Berberich, K., and Weikum, G. (2020). Entities with Quantities: Extraction, Search, and Ranking. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, pages 833–836.
- [Hoffart et al., 2014] Hoffart, J., Milchevski, D., and Weikum, G. (2014). STICS: Searching with strings, things, and cats. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*.
- [Hoffart et al., 2011a] Hoffart, J., Suchanek, F. M., Berberich, K., Lewis-Kelham, E., de Melo, G., and Weikum, G. (2011a). YAGO2: Exploring and Querying World Knowledge in Time, Space, Context, and Many Languages. In *Proceedings of the 20th International Conference Companion on World Wide Web*, WWW '11, pages 229–232.

- [Hoffart et al., 2013] Hoffart, J., Suchanek, F. M., Berberich, K., and Weikum, G. (2013). YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence*, 194:28–61. Artificial Intelligence, Wikipedia and Semi-Structured Resources.
- [Hoffart et al., 2011b] Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Thater, S., and Weikum, G. (2011b). Robust Disambiguation of Named Entities in Text. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 782–792.
- [Hong et al., 2011] Hong, Y., Zhang, J., Ma, B., Yao, J., Zhou, G., and Zhu, Q. (2011). Using Cross-entity Inference to Improve Event Extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, HLT '11*, pages 1127–1136.
- [Houvardas and Stamatatos, 2006] Houvardas, J. and Stamatatos, E. (2006). N-Gram Feature Selection for Authorship Identification. In *AIMSA*.
- [Hu et al., 2020] Hu, Z., Lee, R. K.-W., Wang, L., Lim, E.-P., and Dai, B. (2020). Deep-Style: User Style Embedding for Authorship Attribution of Short Texts. In *APWeb-WAIM 2020: Web and Big Data*, pages 221–229.
- [Huang et al., 2020] Huang, D., Chen, P., Zeng, R., Du, Q., Tan, M., and Gan, C. (2020). Location-Aware Graph Convolutional Networks for Video Question Answering. *ArXiv*, abs/2008.09105.
- [Jia et al., 2018] Jia, B., Huang, X., and Jiao, S. (2018). Application of Semantic Similarity Calculation Based on Knowledge Graph for Personalized Study Recommendation Service. *Educational Sciences: Theory & Practice*, 18(6):2958–2966.
- [Johansson et al., 2014] Johansson, F., Kaati, L., and Shrestha, A. (2014). Time Profiles for Identifying Users in Online Environments. In *2014 IEEE Joint Intelligence and Security Informatics Conference*, pages 83–90.
- [Johansson et al., 2015] Johansson, F., Kaati, L., and Shrestha, A. (2015). Timeprints for identifying social media users with multiple aliases. *Security Informatics*, 4:1–11.
- [Kalgutkar et al., 2019] Kalgutkar, V., Kaur, R., Gonzalez, H., Stakhanova, N., and Matyukhina, A. (2019). Code Authorship Attribution: Methods and Challenges. *ACM Computing Surveys*, 52(1).
- [Kang et al., 2018] Kang, J., Choi, H., and Lee, H. (2018). Deep recurrent convolutional networks for inferring user interests from social media. *Journal of Intelligent Information Systems*, 52:191–209.
- [Kipf and Welling, 2017] Kipf, T. N. and Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017*.

-
- [Kolbe et al., 2020] Kolbe, N., Vandenbussche, P.-Y., Kubler, S., and Le Traon, Y. (2020). LOVBench: Ontology Ranking Benchmark. In *Proceedings of The Web Conference 2020*, WWW '20, pages 1750–1760.
- [Koppel et al., 2007] Koppel, M., Schler, J., and Bonchek-Dokow, E. (2007). Measuring Differentiability: Unmasking Pseudonymous Authors. In *Journal of Machine Learning Research*, vol. 8, pages 1261–1276.
- [Kulkarni et al., 2009] Kulkarni, S., Singh, A., Ramakrishnan, G., and Chakrabarti, S. (2009). Collective Annotation of Wikipedia Entities in Web Text. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 457–466.
- [Kumar, 2020] Kumar, A. (2020). Towards a Better Contextualization of Web Contents via Entity-Level Analytics. In *Advances in Information Retrieval*, pages 613–618.
- [Kumar et al., 2020] Kumar, A., Govind, Alec, C., and Spaniol, M. (2020). Blogger or President? Exploitation of Patterns in Entity Type Graphs for Representative Entity Type Classification. In *12th ACM Conference on Web Science*, WebSci '20, pages 59–68.
- [Kumar et al., 2021] Kumar, A., Govind, and Spaniol, M. (2021). Semantic Search via Entity-Types: The SEMANNOREX Framework. In *Companion Proceedings of the Web Conference 2021*, WWW '21, pages 690–694.
- [Kumar and Spaniol, 2021a] Kumar, A. and Spaniol, M. (2021a). AnnoTag: Concise Content Annotation via LOD Tags derived from Entity-Level Analytics. In *Linking Theory and Practice of Digital Libraries*, pages 175–180.
- [Kumar and Spaniol, 2021b] Kumar, A. and Spaniol, M. (2021b). Semantic Tagging via Entity-Level Analytics: Assessment of Concise Content Tagging. In *Linking Theory and Practice of Digital Libraries*, pages 97–105.
- [Lehmann et al., 2014] Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., and Bizer, C. (2014). DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal*, 6.
- [Leo Breiman, 1984] Leo Breiman, Jerome Friedman, C. J. S. R. O. (1984). Classification and Regression Trees. *Chapman and Hall/CRC*.
- [Li et al., 2008] Li, D., Savova, G., and Kipper-Schuler, K. (2008). Conditional Random Fields and Support Vector Machines for Disorder Named Entity Recognition in Clinical Texts. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 94–95.
- [Li et al., 2019] Li, Y., Jin, R., and Luo, Y. (2019). Classifying relations in clinical narratives using segment graph convolutional and recurrent neural networks (Seg-GCRNs). *Journal of the American Medical Informatics Association : JAMIA*, 26:262–268.

- [Li et al., 2020] Li, Y., Su, Z., Yang, J., and Gao, C. (2020). Exploiting Similarities of User Friendship Networks across Social Networks for User Identification. *Information Sciences*, 506:78–98.
- [Li et al., 2016] Li, Y., Tarlow, D., Brockschmidt, M., and Zemel, R. S. (2016). Gated Graph Sequence Neural Networks. In Bengio, Y. and LeCun, Y., editors, *4th International Conference on Learning Representations, ICLR 2016*.
- [Liao and Zhao, 2019] Liao, X. and Zhao, Z. (2019). Unsupervised Approaches for Textual Semantic Annotation, A Survey. *ACM Comput. Surv.*, 52(4).
- [Ling et al., 2015] Ling, X., Singh, S., and Weld, D. S. (2015). Design Challenges for Entity Linking. *Transactions of the Association for Computational Linguistics (TACL)*, 3:315–328.
- [Ling and Weld, 2012] Ling, X. and Weld, D. S. (2012). Fine-Grained Entity Recognition. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, AAAI’12*, pages 94–100.
- [Lytvyn et al., 2017] Lytvyn, V., Vysotska, V., Veres, O., Rishnyak, I., and Rishnyak, H. (2017). Classification Methods of Text Documents Using Ontology Based Approach. In *Advances in Intelligent Systems and Computing*, vol. 512, pages 229–240.
- [Macgregor and McCulloch, 2006] Macgregor, G. and McCulloch, E. (2006). Collaborative Tagging as a Knowledge Organisation and Resource Discovery Tool. *Library Review*, 55:291–300.
- [Mahdisoltani et al., 2013] Mahdisoltani, F., Biega, J., and Suchanek, F. M. (2013). YAGO3: A Knowledge Base from Multilingual Wikipedias. In *CIDR, Asilomar*.
- [Maity et al., 2017] Maity, S. K., Chakraborty, A., Goyal, P., and Mukherjee, A. (2017). Detection of Sockpuppets in Social Media. In *Companion of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing, CSCW ’17 Companion*, pages 243–246.
- [Manning et al., 2008] Manning, C. D., Raghavan, P., and Schütze, H. (2008). Introduction to Information Retrieval. *Cambridge University Press, USA*.
- [Marouf and Hossian, 2019] Marouf, A. A. and Hossian, R. (2019). Lyricist identification using stylometric features utilizing banglamusicstylo dataset. In *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–4.
- [Masters, 1993] Masters, T. (1993). Practical Neural Network Recipes in C++. *Morgan Kaufmann*.
- [McKenna et al., 2019] McKenna, L., Debruyne, C., and O’Sullivan, D. (2019). NAISC: An Authoritative Linked Data Interlinking Approach for the Library Domain. In *2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 11–20.

-
- [Medeiros et al., 2018] Medeiros, J. F., Pereira Nunes, B., Siqueira, S. W. M., and Portes Paes Leme, L. A. (2018). TagTheWeb: Using Wikipedia Categories to Automatically Categorize Resources on the Web. In *The Semantic Web: ESWC 2018 Satellite Events*, pages 153–157.
- [Mendes et al., 2011] Mendes, P. N., Jakob, M., García-Silva, A., and Bizer, C. (2011). DBpedia Spotlight: Shedding Light on the Web of Documents. In *Proceedings of the 7th International Conference on Semantic Systems, I-Semantics '11*, pages 1–8.
- [Mihalcea and Csomai, 2007] Mihalcea, R. and Csomai, A. (2007). Wikify! Linking Documents to Encyclopedic Knowledge. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, CIKM '07*, pages 233–242.
- [Mihaylov et al., 2015] Mihaylov, T., Koychev, I., Georgiev, G., and Nakov, P. (2015). Exposing Paid Opinion Manipulation Trolls. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 443–450.
- [Miller, 1995] Miller, G. A. (1995). WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41.
- [Milne and Witten, 2008] Milne, D. and Witten, I. H. (2008). Learning to Link with Wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM '08*, pages 509–518.
- [Montoya et al., 2018] Montoya, D., Tanon, T. P., Abiteboul, S., Senellart, P., and Suchanek, F. M. (2018). A Knowledge Base for Personal Information Management. In *LDOW@WWW*.
- [Moreno et al., 2017] Moreno, J. G., Besançon, R., Beaumont, R., D’hondt, E., Ligozat, A.-L., Rosset, S., Tannier, X., and Grau, B. (2017). Combining Word and Entity Embeddings for Entity Linking. In *The Semantic Web*, pages 337–352.
- [Musen, 2015] Musen, M. A. (2015). The Protégé Project: A Look Back and a Look Forward. *AI Matters*, 1(4):4–12.
- [Nakov et al., 2017] Nakov, P., Mihaylova, T., Màrquez, L., Shiroya, Y., and Koychev, I. (2017). Do Not Trust the Trolls: Predicting Credibility in Community Question Answering Forums. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 551–560.
- [Nguyen et al., 2014] Nguyen, D. B., Hoffart, J., Theobald, M., and Weikum, G. (2014). AIDA-light: High-Throughput Named-Entity Disambiguation. In *Proceedings of the Workshop on Linked Data on the Web co-located with the 23rd International World Wide Web Conference (WWW 2014), CEUR Workshop Proceedings*.
- [Obeidat et al., 2019] Obeidat, R., Fern, X., Shahbazi, H., and Tadepalli, P. (2019). Description-Based Zero-shot Fine-Grained Entity Typing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, pages 807–814.

- [Ottoni et al., 2014] Ottoni, R., Casas, D., Pesce, J. P., Meira Jr, W., Wilson, C., Mislove, A., and Almeida, V. (2014). Of Pins and Tweets: Investigating how users behave across image- and text-based social networks. *Proceedings of the 8th International Conference on Weblogs and Social Media (ICWSM 2014)*, pages 386–395.
- [Panigutti et al., 2020] Panigutti, C., Perotti, A., and Pedreschi, D. (2020). Doctor XAI: An Ontology-Based Approach to Black-Box Sequential Data Classification Explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, FAT* '20*, pages 629–639.
- [Parisot et al., 2018] Parisot, S., Ktena, S. I., Ferrante, E., Lee, M. J., Guerrero, R., Glocker, B., and Rueckert, D. (2018). Disease Prediction using Graph Convolutional Networks: Application to Autism Spectrum Disorder and Alzheimer’s Disease. *Medical Image Analysis*, 48:117–130.
- [Pellissier Tanon et al., 2020] Pellissier Tanon, T., Weikum, G., and Suchanek, F. (2020). YAGO 4: A Reason-able Knowledge Base. In *The Semantic Web*, pages 583–596.
- [Pennekamp et al., 2019] Pennekamp, J., Henze, M., Hohlfeld, O., and Panchenko, A. (2019). Hi Doppelgänger: Towards Detecting Manipulation in News Comments. In *Companion Proceedings of The 2019 World Wide Web Conference, WWW '19*, pages 197–205.
- [Perifanos et al., 2018] Perifanos, K., Florou, E., and Goutsos, D. (2018). Neural Embeddings for Idiolect Identification. In *2018 9th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pages 1–3.
- [Peters et al., 2019] Peters, M. E., Neumann, M., Logan, R., Schwartz, R., Joshi, V., Singh, S., and Smith, N. A. (2019). Knowledge Enhanced Contextual Word Representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54.
- [Pfahler and Morik, 2020] Pfahler, L. and Morik, K. (2020). Semantic Search in Millions of Equations. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '20*, pages 135–143.
- [Piao et al., 2018] Piao, S., Rayson, P., Knight, D., and Watkins, G. (2018). Towards a Welsh Semantic Annotation System. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- [Prokofyev et al., 2014] Prokofyev, R., Demartini, G., and Cudré-Mauroux, P. (2014). Effective Named Entity Recognition for Idiosyncratic Web Collections. In *Proceedings of the 23rd International Conference on World Wide Web, WWW '14*, pages 397–408.
- [Qian, 2013] Qian, R. (2013). Understand your world with bing. <https://blogs.bing.com/search/2013/03/21/understand-your-world-with-bing/>.

-
- [Quinlan, 1993] Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. *Morgan Kaufmann Publishers Inc.*, San Francisco, CA, USA.
- [Quinlan, 2004] Quinlan, J. R. (2004). Induction of Decision Trees. *Machine Learning*, 1:81–106.
- [Ramanathan and Kapoor, 2009] Ramanathan, K. and Kapoor, K. (2009). Creating User Profiles Using Wikipedia. In *Conceptual Modeling - ER 2009*, pages 415–427.
- [Resnik, 1995] Resnik, P. (1995). Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1, IJCAI '95*, pages 448–453.
- [Rexha et al., 2015] Rexha, A., Klampfl, S., Kröll, M., and Kern, R. (2015). Towards Authorship Attribution for Bibliometrics using Stylometric Features. In *CLBib@ISSI*.
- [Rexha et al., 2016] Rexha, A., Klampfl, S., Kröll, M., and Kern, R. (2016). Towards a More Fine Grained Analysis of Scientific Authorship: Predicting the Number of Authors Using Stylometric Features. In *BIR@ECIR*.
- [Rexha et al., 2018] Rexha, A., Kröll, M., Ziak, H., and Kern, R. (2018). Authorship identification of documents with high content similarity. *Scientometrics*, 115:223 – 237.
- [Rifkin and Klautau, 2004] Rifkin, R. and Klautau, A. (2004). In Defense of One-Vs-All Classification. *J. Mach. Learn. Res.*, 5:101–141.
- [Sandhaus, 2008] Sandhaus, E. (2008). The New York Times Annotated Corpus LDC2008T19. In *Philadelphia: Linguistic Data Consortium*.
- [Sarwar et al., 2020] Sarwar, R., Porthaveepong, T., Rutherford, A., Rakthanmanon, T., and Nutanong, S. (2020). StyloThai: A Scalable Framework for Stylometric Authorship Identification of Thai Documents. In *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 19(3), pages 1–15.
- [Schwartz et al., 2013] Schwartz, R., Tsur, O., Rappoport, A., and Koppel, M. (2013). Authorship Attribution of Micro-Messages. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1880–1891.
- [Sen, 2012] Sen, P. (2012). Collective Context-Aware Topic Models for Entity Disambiguation. In *Proceedings of the 21st International Conference on World Wide Web, WWW '12*, pages 729–738.
- [Sheth and Deshpande, 2015] Sheth, N. S. and Deshpande, A. R. (2015). A Review of Splitting Criteria for Decision Tree Induction. *Fuzzy Systems*, 7(1).
- [Shimaoka et al., 2017] Shimaoka, S., Stenetorp, P., Inui, K., and Riedel, S. (2017). Neural Architectures for Fine-grained Entity Type Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1271–1280.

- [Shrestha et al., 2017] Shrestha, P., Sierra, S., González, F., Montes, M., Rosso, P., and Solorio, T. (2017). Convolutional Neural Networks for Authorship Attribution of Short Texts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 669–674.
- [Slimani, 2013] Slimani, T. (2013). Description and Evaluation of Semantic Similarity Measures Approaches. *arXiv preprint arXiv:1310.8059*.
- [Song et al., 2015] Song, M., Yu, H., and Han, W.-S. (2015). Developing a Hybrid Dictionary-based Bio-entity Recognition Technique. *BMC Medical Informatics and Decision Making*, 15:S9 – S9.
- [Sousa Silva et al., 2011] Sousa Silva, R., Laboreiro, G., Sarmiento, L., Grant, T., Oliveira, E., and Maia, B. (2011). ‘twazn me!!! ;(’ Automatic Authorship Analysis of Micro-Blogging Messages. In *Natural Language Processing and Information Systems*, pages 161–168.
- [Spitz et al., 2017] Spitz, A., Almasian, S., and Gertz, M. (2017). EVELIN: Exploration of Event and Entity Links in Implicit Networks. In *Proceedings of the 26th International Conference on World Wide Web Companion, WWW ’17 Companion*, pages 273–277.
- [Spitz and Gertz, 2016] Spitz, A. and Gertz, M. (2016). Terms over LOAD: Leveraging Named Entities for Cross-Document Extraction and Summarization of Events. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’16*, pages 503–512.
- [Stamatatos, 2009] Stamatatos, E. (2009). A Survey of Modern Authorship Attribution Methods. In *Journal of the American Society for Information Science and Technology*, vol. 60, pages 538–556.
- [Stork et al., 2019] Stork, L., Weber, A., Gassó Miracle, E., Verbeek, F., Plaat, A., van den Herik, J., and Wolstencroft, K. (2019). Semantic annotation of natural history collections. *Journal of Web Semantics*, 59:100462.
- [Suchanek et al., 2007] Suchanek, F., Kasneci, G. M., and Weikum, G. M. (2007). YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia. In *Proceedings of 16th International World Wide Web Conference (WWW) 2007*, pages 697–706.
- [Suchanek et al., 2008] Suchanek, F. M., Kasneci, G., and Weikum, G. (2008). YAGO: A Large Ontology from Wikipedia and WordNet. In *Journal of Web Semantics*, vol. 6(3), pages 203–217.
- [Sweeney and Padmanabhan, 2017] Sweeney, C. and Padmanabhan, D. (2017). Multi-entity sentiment analysis using entity-level feature extraction and word embeddings approach. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 733–740.

-
- [Tang et al., 2009] Tang, L., Rajan, S., and Narayanan, V. K. (2009). Large Scale Multi-label Classification via Metalabeler. In *Proceedings of the 18th International Conference on World Wide Web, WWW '09*, pages 211–220.
- [Tao et al., 2013] Tao, C., Song, D., Sharma, D., and Chute, C. G. (2013). Semantator: Semantic annotator for converting biomedical text to linked data. *Journal of Biomedical Informatics*, 46(5):882–893.
- [Toffler, 1980] Toffler, A. (1980) The third wave / by Alvin Toffler 1st ed. ed.. *Morrow New York*, 544 p.
- [Tonon et al., 2013] Tonon, A., Catasta, M., Demartini, G., Cudré-Mauroux, P., and Aberer, K. (2013). TRank: Ranking Entity Types Using the Web of Data. In *The Semantic Web – ISWC 2013*, pages 640–656.
- [Tonon et al., 2016] Tonon, A., Catasta, M., Prokofyev, R., Demartini, G., Aberer, K., and Cudré-Mauroux, P. (2016). Contextualized Ranking of Entity Types Based on Knowledge Graphs. *Web Semant.*, 37(C):170–183.
- [Torfi et al., 2020] Torfi, A., Shirvani, R. A., Keneshloo, Y., Tavaf, N., and Fox, E. A. (2020). Natural Language Processing Advancements By Deep Learning: A Survey. *CoRR*, abs/2003.01200.
- [Tsoumakas and Katakis, 2007] Tsoumakas, G. and Katakis, I. M. (2007). Multi-Label Classification: An Overview. *Int. J. Data Warehous. Min.*, 3:1–13.
- [Usbeck et al., 2014] Usbeck, R., Ngonga Ngomo, A.-C., Röder, M., Gerber, D., Coelho, S. A., Auer, S., and Both, A. (2014). AGDISTIS - Graph-Based Disambiguation of Named Entities Using Linked Data. *The Semantic Web – ISWC 2014*, pages 457–471.
- [Veličković et al., 2018] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph Attention Networks. *International Conference on Learning Representations, ICLR 2018*.
- [Villar-Rodriguez et al., 2016] Villar-Rodriguez, E., Ser, J. D., Bilbao, M. N., and Salcedo-Sanz, S. (2016). A feature selection method for author identification in interactive communications based on supervised learning and language typicality. *Engineering Applications of Artificial Intelligence*, 56:175–184.
- [Vrandečić and Krötzsch, 2014] Vrandečić, D. and Krötzsch, M. (2014). Wikidata: A Free Collaborative Knowledge base. *Commun. ACM*, 57(10):78–85.
- [W3C et al., 2014] W3C et al. (2014). RDF 1.1 Concepts and Abstract Syntax.
- [Wang et al., 2018] Wang, Z., Lv, Q., Lan, X., and Zhang, Y. (2018). Cross-lingual Knowledge Graph Alignment via Graph Convolutional Networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 349–357.

- [Wang et al., 2020] Wang, Z., Wu, C.-H., Li, Q.-B., Yan, B., and Zheng, K.-F. (2020). Encoding Text Information with Graph Convolutional Networks for Personality Recognition. *Applied Sciences*, 10(12).
- [Weikum et al., 2011] Weikum, G., Ntarmos, N., Spaniol, M., Triantafillou, P., Benczúr, A. A., Kirkpatrick, S., Rigaux, P., and Williamson, M. (2011). Longitudinal Analytics on Web Archive Data: It’s About Time! In *CIDR*, pages 199–202.
- [Weng et al., 2010] Weng, J., Lim, E.-P., Jiang, J., and He, Q. (2010). TwitterRank: Finding Topic-Sensitive Influential Twitterers. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining, WSDM ’10*, pages 261–270.
- [Wu et al., 2019] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. (2019). A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32:4–24.
- [Xiong et al., 2017] Xiong, C., Power, R., and Callan, J. (2017). Explicit Semantic Ranking for Academic Search via Knowledge Graph Embedding. In *Proceedings of the 26th International Conference on World Wide Web, WWW ’17*, pages 12710–12719.
- [Yahya et al., 2012] Yahya, M., Berberich, K., Elbassuoni, S., Ramanath, M., Tresp, V., and Weikum, G. (2012). Natural Language Questions for the Web of Data. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 379–390.
- [Yao et al., 2018] Yao, L., Mao, C., and Luo, Y. (2018). Graph Convolutional Networks for Text Classification. In *CoRR*, abs/1809.05679.
- [Ying et al., 2018a] Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., and Leskovec, J. (2018a). Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD’18*, pages 974–983.
- [Ying et al., 2018b] Ying, R., You, J., Morris, C., Ren, X., Hamilton, W. L., and Leskovec, J. (2018b). Hierarchical Graph Representation Learning with Differentiable Pooling. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS’18*, pages 4805–4815.
- [Yosef et al., 2012] Yosef, M. A., Bauer, S., Hoffart, J., Spaniol, M., and Weikum, G. (2012). HYENA: Hierarchical Type Classification for Entity Names. In *Proceedings of COLING 2012: Posters*, pages 1361–1370.
- [Yosef et al., 2011] Yosef, M. A., Hoffart, J., Bordino, I., Spaniol, M., and Weikum, G. (2011). AIDA: An Online Tool for Accurate Disambiguation of Named Entities in Text and Tables. *Proc. VLDB Endow.*, 4(12):1450–1453.
- [Zhang et al., 2019] Zhang, C., Li, Q., and Song, D. (2019). Aspect-based Sentiment Classification with Aspect-specific Graph Convolutional Networks. In *Proceedings of*

the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 4568–4578.

- [Zhang et al., 2014] Zhang, C., Wu, X., Niu, Z., and Ding, W. (2014). Authorship identification from unstructured texts. *Knowledge-Based Systems*, 66:99–111.
- [Zhang et al., 2018] Zhang, M., Cui, Z., Neumann, M., and Chen, Y. (2018). An End-to-End Deep Learning Architecture for Graph Classification. In *AAAI*.
- [Zhang et al., 2020] Zhang, S., Yin, H., Chen, T., Hung, Q. V. N., Huang, Z., and Cui, L. (2020). GCN-Based User Representation Learning for Unifying Robust Recommendation and Fraudster Detection. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20*, pages 689–698.
- [Zhang et al., 2017] Zhang, Y., Jatowt, A., and Tanaka, K. (2017). Is Tofu the Cheese of Asia? Searching for Corresponding Objects across Geographical Areas. In *Proceedings of the 26th International Conference on World Wide Web Companion, WWW '17 Companion*, pages 1033–1042.
- [Zhao, 2017] Zhao, Y. (2017). Research on Entity Recognition in Traditional Chinese Medicine Diet. In *2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, volume 1, pages 284–287.
- [Zheng et al., 2011] Zheng, X., Lai, Y. M., Chow, K., Hui, L. C., and Yiu, S. (2011). Sockpuppet Detection in Online Discussion Forums. In *2011 Seventh International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 374–377.
- [Zhou et al., 2019] Zhou, W., Wang, J., Lin, J., Li, J., Han, J., and Hu, S. (2019). A Time-Series Sockpuppet Detection Method for Dynamic Social Relationships. In *Database Systems for Advanced Applications*, pages 36–51.

Contextualization of Web Contents through Semantic Enrichment from Linked Open Data

Abstract: Thirty years of the Web have led to a tremendous amount of contents and the enormous growth is still ongoing, even accelerating. Thus, Web users are confronted with an abundance of information. While this is clearly beneficial, there is a risk of “information overload” and it is very hard for a Web user to access, contextualize and digest Web contents. Thus, there is an increasing need of categorizing, summarizing, and/or interpretability of Web contents in order to get a proper contextualization. While contents of the early years have been predominantly “simple” HTML documents, more recent ones have become more and more “machine-interpretable” and contribute to the ever growing Linked Open Data (LOD) cloud. LOD provides us a multitude of research opportunities for investigating and harvesting insights about Web contents.

In this thesis, we investigate a variety of tasks related to semantic contextualization of Web contents. Specifically, we address three facets in the context of distillation of the Web contents, namely, entity-driven content analysis, semantic annotation & retrieval, and semantic user tracing. We hypothesize that named entities and their types present in a Web document convey substantial semantic information. We have displayed by employing multiple studies that projecting Web contents to the entity-level captures their fundamental semantics. Thus, it provides significant knowledge about the Web contents and, subsequently, comprehensibility. We report novel findings over diverse tasks in an attempt to accomplish our overall goal of a better contextualization of Web contents.

Keywords: Entity-level Analytics, Entity-type Classification, Semantic Document Representation, Multilingual Web Data, Semantic User Representation, Web Semantics

Contextualisation des contenus Web par l'enrichissement sémantique à partir de données

Résumé: Les trente années d'existence du Web ont donné lieu à une quantité phénoménale de contenus et cette croissance énorme se poursuit, voire s'accélère. Les utilisateurs du Web sont donc confrontés à une abondance d'informations. Bien que cela soit clairement bénéfique, il existe un risque de “surcharge d'informations” et il est très difficile pour un utilisateur du Web d'accéder, de contextualiser et de digérer les contenus du Web. Il est donc de plus en plus nécessaire pour catégoriser, de résumer et/ou d'interpréter les contenus du Web afin d'obtenir une contextualisation adéquate. Alors que les contenus des premières années étaient principalement de “simples” documents HTML, les plus récents sont devenus de plus en plus “interprétables par les machines” et contribuent au nuage de données ouvertes liées (LOD) en constante expansion. Le LOD nous offre une multitude de possibilités de recherche pour étudier et récolter des informations sur les contenus du Web.

Dans cette thèse, nous étudions une variété de tâches liées à la contextualisation sémantique des contenus Web. Plus précisément, nous abordons trois facettes dans le contexte de la distillation des contenus Web, à savoir, l'analyse de contenu axée sur les entités, l'annotation et la recherche sémantiques, et le traçage sémantique des utilisateurs. Nous supposons que les entités nommées et leurs types présents dans un document Web véhiculent des informations sémantiques substantielles. Nous avons démontré, à l'aide de multiples études, que la projection des contenus Web au niveau des entités permet de capturer leur sémantique fondamentale. Ainsi, elle fournit des connaissances significatives sur le contenu du Web et, par conséquent, une meilleure compréhension. Nous présentons de nouveaux résultats sur diverses tâches dans le but d'atteindre notre objectif global d'une meilleure contextualisation des contenus Web.

Mots-clés: Analyse au niveau des entités, Classification des types d'entités, Représentation sémantique des documents, Données Web multilingues, Représentation sémantique des utilisateurs, Sémantique du Web.