Knowledge Extraction from Fictional Texts

Cuong Xuan Chu

Dissertation zur Erlangung des Grades des
DOKTORS DER INGENIEURWISSENSCHAFTEN (DR.-ING.)
der Fakultät für Mathematik und Informatik
der Universität des Saarlandes

Saarbrücken, 2022

Day of Colloquium	25/04/2022
Dean of the Faculty	Univ.-Prof. Dr. Jürgen Steimle
Chair of the Committee	Prof. Dr. Dietrich Klakow
Reporters First Reviewer	Prof. Dr. Gerhard Weikum
Second Reviewer	Dr. Simon Razniewski
Third Reviewer	Prof. Dr. Martin Theobald
Academic Assistant	Dr. Frances Yung

Abstract

Knowledge extraction from text is a key task in natural language processing, which involves many sub-tasks, such as taxonomy induction, named entity recognition and typing, relation extraction, knowledge canonicalization and so on. By constructing structured knowledge from natural language text, knowledge extraction becomes a key asset for search engines, question answering and other downstream applications. However, current knowledge extraction methods mostly focus on prominent real world entities with Wikipedia and mainstream news articles as sources. The constructed knowledge bases, therefore, lack information about long-tail domains, with fiction and fantasy as archetypes. Fiction and fantasy are core parts of our human culture, spanning from literature to movies, TV series, comics and video games. With thousands of fictional universes which have been created, knowledge from fictional domains are subject of search-engine queries – by fans as well as cultural analysts. Unlike the real-world domain, knowledge extraction on such specific domains like fiction and fantasy has to tackle several key challenges:

- *Training data.* Sources for fictional domains mostly come from books and fan-built content, which is sparse and noisy, and contains difficult structures of texts, such as dialogues and quotes. Training data for key tasks such as taxonomy induction, named entity typing or relation extraction are also not available.

- *Domain characteristics and diversity.* Fictional universes can be highly sophisticated, containing entities, social structures and sometimes languages that are completely different from the real world. State-of-the-art methods for knowledge extraction make assumptions on entity-class, subclass and entity-entity relations that are often invalid for fictional domains. With different genres of fictional domains, another requirement is to transfer models across domains.

- *Long fictional texts.* While state-of-the-art models have limitations on the input sequence length, it is essential to develop methods that are able to deal with very long texts (e.g. entire books), to capture multiple contexts and leverage widely spread cues.

This dissertation addresses the above challenges, by developing new methodologies that advance the state of the art on knowledge extraction in fictional domains.

- The first contribution is a method, called TiFi, for constructing type systems (taxonomy induction) for fictional domains. By tapping noisy fan-built content from online communities such as Wikia, TiFi induces taxonomies through three main steps: category cleaning, edge cleaning and top-level construction. Exploiting a variety of features from the original input, TiFi is able to construct taxonomies for a diverse range of fictional domains with high precision.

- The second contribution is a comprehensive approach, called ENTYFI, for named entity recognition and typing in long fictional texts. Built on 205 automatically induced high-quality type systems for popular fictional domains, ENTYFI exploits the overlap and reuse of these fictional domains on unseen texts. By combining different typing modules with a consolidation stage, ENTYFI is able to do fine-grained entity typing in long fictional texts with high precision and recall.

- The third contribution is an end-to-end system, called KnowFi, for extracting relations between entities in very long texts such as entire books. KnowFi leverages background knowledge from 142 popular fictional domains to identify interesting relations and to collect distant training samples. KnowFi devises a similarity-based ranking technique to reduce false positives in training samples and to select potential text passages that contain seed pairs of entities. By training a hierarchical neural network for all relations, KnowFi is able to infer relations between entity pairs across long fictional texts, and achieves gains over the best prior methods for relation extraction.

Kurzfassung

Wissensextraktion ist ein Schlüsselaufgabe bei der Verarbeitung natürlicher Sprache, und umfasst viele Unteraufgaben, wie Taxonomiekonstruktion, Entitätserkennung und Typisierung, Relationsextraktion, Wissenskanonikalisierung, etc. Durch den Aufbau von strukturiertem Wissen (z.B. Wissensdatenbanken) aus Texten wird die Wissensextraktion zu einem Schlüsselfaktor für Suchmaschinen, Question Answering und andere Anwendungen. Aktuelle Methoden zur Wissensextraktion konzentrieren sich jedoch hauptsächlich auf den Bereich der realen Welt, wobei Wikipedia und Mainstream-Nachrichtenartikel die Hauptquellen sind. Fiktion und Fantasy sind Kernbestandteile unserer menschlichen Kultur, die sich von Literatur bis zu Filmen, Fernsehserien, Comics und Videospielen erstreckt. Für Tausende von fiktiven Universen wird Wissen aus Suchmaschinen abgefragt – von Fans ebenso wie von Kulturwissenschaftler. Im Gegensatz zur realen Welt muss die Wissensextraktion in solchen spezifischen Domänen wie Belletristik und Fantasy mehrere zentrale Herausforderungen bewältigen:

- *Trainingsdaten.* Quellen für fiktive Domänen stammen hauptsächlich aus Büchern und von Fans erstellten Inhalten, die spärlich und fehlerbehaftet sind und schwierige Textstrukturen wie Dialoge und Zitate enthalten. Trainingsdaten für Schlüsselaufgaben wie Taxonomie-Induktion, Named Entity Typing oder Relation Extraction sind ebenfalls nicht verfügbar.

- *Domain-Eigenschaften und Diversität.* Fiktive Universen können sehr anspruchsvoll sein und Entitäten, soziale Strukturen und manchmal auch Sprachen enthalten, die sich von der realen Welt völlig unterscheiden. Moderne Methoden zur Wissensextraktion machen Annahmen über Entity-Class-, Entity-Subclass- und Entity-Entity-Relationen, die für fiktive Domänen oft ungültig sind. Bei verschiedenen Genres fiktiver Domänen müssen Modelle auch über fiktive Domänen hinweg transferierbar sein.

- *Lange fiktive Texte.* Während moderne Modelle Einschränkungen hinsichtlich der Länge der Eingabesequenz haben, ist es wichtig, Methoden zu entwickeln, die in

der Lage sind, mit sehr langen Texten (z.B. ganzen Büchern) umzugehen, und mehrere Kontexte und verteilte Hinweise zu erfassen.

Diese Dissertation befasst sich mit den oben genannten Herausforderungen, und entwickelt Methoden, die den Stand der Kunst zur Wissensextraktion in fiktionalen Domänen voranbringen.

- Der erste Beitrag ist eine Methode, genannt TiFi, zur Konstruktion von Typsystemen (Taxonomie induktion) für fiktive Domänen. Aus von Fans erstellten Inhalten in Online-Communities wie Wikia induziert TiFi Taxonomien in drei wesentlichen Schritten: Kategoriereinigung, Kantenreinigung und Top-Level-Konstruktion. TiFi nutzt eine Vielzahl von Informationen aus den ursprünglichen Quellen und ist in der Lage, Taxonomien für eine Vielzahl von fiktiven Domänen mit hoher Präzision zu erstellen.

- Der zweite Beitrag ist ein umfassender Ansatz, genannt ENTYFI, zur Erkennung von Entitäten, und deren Typen, in langen fiktiven Texten. Aufbauend auf 205 automatisch induzierten hochwertigen Typsystemen für populäre fiktive Domänen nutzt ENTYFI die Überlappung und Wiederverwendung dieser fiktiven Domänen zur Bearbeitung neuer Texte. Durch die Zusammenstellung verschiedener Typisierungsmodule mit einer Konsolidierungsphase ist ENTYFI in der Lage, in langen fiktionalen Texten eine feinkörnige Entitätstypisierung mit hoher Präzision und Abdeckung durchzuführen.

- Der dritte Beitrag ist ein End-to-End-System, genannt KnowFi, um Relationen zwischen Entitäten aus sehr langen Texten wie ganzen Büchern zu extrahieren. KnowFi nutzt Hintergrundwissen aus 142 beliebten fiktiven Domänen, um interessante Beziehungen zu identifizieren und Trainingsdaten zu sammeln. KnowFi umfasst eine ähnlichkeitsbasierte Ranking-Technik, um falsch positive Einträge in Trainingsdaten zu reduzieren und potenzielle Textpassagen auszuwählen, die Paare von Kandidats-Entitäten enthalten. Durch das Trainieren eines hierarchischen neuronalen Netzwerkes für alle Relationen ist KnowFi in der Lage, Relationen zwischen Entitätspaaren aus langen fiktiven Texten abzuleiten, und übertrifft die besten früheren Methoden zur Relationsextraktion.

Acknowledgments

First and foremost, I would like to thank my supervisor, Prof. Dr. Gerhard Weikum, for being my mentor since I started my Master program, giving me the opportunity to carry out this research and providing invaluable guidance throughout my doctoral studies. From him, I have learned about simplicity, vision and enthusiasm that broaden my attitude towards research. This seven years is definitely an invaluable period of time in my life.

I would like to thank my co-supervisor, Dr. Simon Razniewski, for being a great friend and an excellent collaborator. Without his insightful comments and guidance, I would have not done this work. Working with Simon, I have gained a lot of valuable experience on the research and advices for my future career.

I would like to thank the additional reviewer and examiner of my dissertation, Prof. Dr. Martin Theobald, and thanks to Prof. Dr. Dietrich Klakow and Dr. Frances Yung for being the chair and the academic assistant of my Ph.D. committee.

I also would like to thank my colleagues and staff at D5 group for making the workplace an exciting and friendly atmosphere. A special note of thanks to Petra, Alena, Daniela, Jenny and Steffi, for their great support. Many thanks to my officemates, Dat Ba Nguyen, Dhruv Gupta, Mohamed H. Gad-Elrab, to my lunchmates, Vinh-Thinh Ho, Tuan-Phong Nguyen, Thong Nguyen and Hai-Dang Tran, for the relax and helpful discussions with them.

I am also very grateful to Uncle Duy Ta, Aunt Hong Le, and my friends in Saarbruecken for their great help to my life.

Last but not least, I would like to thank my family for their constant support throughout the years.

Contents

Chapter 1

Introduction

1.1 Motivation and Scope

Motivation With the tremendous expansion of the internet, there is a huge amount of data that is put online every day. This information is stored and shared in different forms such as text, audio or visual. Among them, text is the most popular form that is presented in variety sources such as books, news articles, web pages and more. With the rapid development of artificial intelligence, the need to develop intelligent applications requires computers or machines to be able to learn "knowledge". That is the time when the term *knowledge harvesting* (or *knowledge extraction*) appeared. Knowledge harvesting is the task of extracting structured knowledge (or machine-readable knowledge) from noisy Internet content and storing them into *knowledge bases*. A knowledge base (KB) is a collection of facts, usually presented in a form of triples SPO: subject-predicate-object, about the real world. Consider the following example:

> "In 1895, Marie Curie married the French physicist Pierre Curie, and she shared the 1903 Nobel Prize in Physics with him and with the physicist Henri Becquerel for their pioneering work developing the theory of "radioactivity" – a term she coined." [1]

From this text, the goal of knowledge harvesting is extracting a list of facts, such as:

- `<Marie_Curie, married_to, Pierre_Curie, 1895>`
- `<Marie_Curie, win, Nobel_Prize_in_Physics, 1903>`
- `<Pierre_Curie, win, Nobel_Prize_in_Physics, 1903>`
- `<Henri_Becquerel, win, Nobel_Prize_in_Physics, 1903>`
- `<Marie_Curie, work_on, Radioactivity>`

[1] https://en.wikipedia.org/wiki/Marie_Curie

- `<Pierre_Curie, is_a, Physicist>`
- ...

In the last decade, computer scientists have put a lot of effort into automatically extracting and organizing these structured knowledge. Large KBs have been built like YAGO [Hoffart et al., 2013, Suchanek et al., 2007], DBpedia [Auer et al., 2007], Wikidata [Vrandečić and Krötzsch, 2014], etc., and become a key asset on search engine and question answering systems. For example, when a user searches for "Nobel prizes of Marie Curie" on search engine systems like Google or Bing, a direct answer, which includes a list of two Nobel prizes, `Physics, 1903` and `Chemistry, 1911`, is returned. Apparently, these systems have knowledge about `Marie Curie` and knowledge about the concept *Nobel prizes*, hence, are able to provide answers for the user. In fact, Google has Google Knowledge Graph and Bing has Microsoft Satori in their backend data.

However, current KBs are mostly constructed for our real world domain, where Wikipedia and main stream news are primary sources. These KBs, hence, lack knowledge about long-tail domains, where fiction and fantasy are the most prominent. Fiction and fantasy are core parts of our human culture, spanning from traditional literature into modern stories, movies, TV series and games. People have created a huge collection of fictional universes such as Greek and Roman Mythology (myths), Marvel and DC comics (comics), Harry Potter and Lord of the Rings (high fantasy novels), World of Warcraft and League of Legends (games), and so on. These universes are well-structured, with thousands of entities and types that are usually completely different from our real world. Served as entertainment, people spend a lot of time on fiction and fantasy. As a statistic in 2020, a U.S consumer spent 213 minutes (3h33min) daily watching TV on average [2]. With such high attention, the information from fiction and fantasy are usually subjects for search-engine queries by fans and topics for culture analysis.

Consider more examples. As a fan of the popular TV series *Game of Thrones*, a user wants to retrieve a list of "enemies of Jon Snow" – a main character in the series and looks for the answer from search engine systems. Instead of providing a list of enemies, these systems, however, only return a list of web pages where the user can access and find the answer by themselves. This scenario also happens when a user is looking for a list of "muggles in Harry Potter" – another popular TV series (and novels as well). Apparently, KBs in the backend data of search engine systems lack information about these fictional domains. Research shows that in popular recommendation systems, the dataset in DBpedia only contains less than 85% number of movies, 63% number of music artists and 31% for books [Hertling and Paulheim, 2018, Noia et al., 2016] and

[2] https://www.statista.com/statistics/186833/average-television-use-per-person-in-the-us-since-2002/

the numbers for entities and facts about them in these domains are much more lower. Therefore, knowledge extraction from fictional domains becomes an essential task. Not only using the output to enhance existing KBs, techniques used in these domains can be also adapted for other specific domains such as professional domains, companies or even in new languages.

Scope Working on knowledge extraction involves three main sub-tasks: building type systems for entities (e.g. taxonomy induction), named entity recognition and typing, and relation extraction.

Taxonomy induction is the task of constructing type systems or class subsumption hierarchies. For example, electric guitar players are rock musicians, and muggle-born wizards are magic creatures. Taxonomies are an essential part of KBs, and important resources for a variety of tasks such as entity search, question answering and relation extraction. As statistics, YAGO includes over 350,000 entity types [Suchanek et al., 2007], and DBPedia includes over one million type labels and concepts that are retrieved from Wikipedia and also linked to other KBs such as Yago, UMBEL and schema.org.

Named entity recognition and typing is the task of identifying entity mentions in text and classifying them into semantic classes such as person, location, etc. as in coarse-grained level, or musicians, muggle-born wizards, etc. as in finer-grained level. For the example about Marie Curie, state-of-the-art NER systems annotate `Marie Curie`, `Pierre Curie` and `Henri Becquerel` as *person* and *physicist*, and `1903 Nobel Prize in Physics` as *award*.

Relation extraction is the task of identifying and classifying semantic relations between entities, and thus can extract facts from natural language texts. For example, the relation *spouse* between `Marie Curie` and `Pierre Curie` can be inferred based on the context around these two entities.

Along with the above sub-tasks, a variety of other sub-tasks are also tackled to improve the quality of extracted knowledge, such as co-reference resolution, name entity disambiguation and discourse parsing. Although those problems have been investigated for a long time, knowledge about fiction and fantasy has been not explored yet. The issues come from sparse sources that are used to extract the knowledge and suitable methodologies for natural language processing and knowledge extraction for these specific domains.

1.2 Challenges

Challenge C1: Input Sources and Training Data Knowledge extraction mainly takes the Internet content as resources. While Wikipedia, a premium source with rich and high-quality content, is the main input for knowledge extraction in the real-world domain, sources for fictional domains come from books or fan-built content, which is noisy and contains difficult structures of text such as dialogues and quotes. In addition, with recent advances in deep learning, it is essential to prepare training data for each specific NLP task, which are mostly not available when working on new domains, like fiction and fantasy. For example, taxonomy induction in the real-world domain can leverage the existing Wikipedia category system as the starting point [Gupta et al., 2016c, Hoffart et al., 2013, Ponzetto and Strube, 2011], but this category network is not suitable for fiction and fantasy due to poor coverage. The taxonomies (or type systems) also needs to be pre-defined and constructed before working on named entity recognition and typing task, especially when the target types are fine-grained. In the case of relation extraction, output relations and their training data are also not available for fictional domains.

Challenge C2: Domain-specific Taxonomy Entity classes and subclass relations are different from the real-world domain. State-of-the-art methods for taxonomy induction make assumptions about the surface forms of entity names and entity classes which do not apply in fictional domains. For example, they assume typical phrases for classes (e.g. noun phrases in plural form) and named entities (e.g. proper names) which do not always hold in fictional domains. Also the assumption that certain classes are disjoint is also invalid (e.g., living beings and abstract entities, the oracle of Delphi being a counterexample).

Challenge C3: Contextual Typing in Long Fictional Texts State-of-the-art methods for entity typing on news and other real-world texts leverage types from Wikipedia categories or WordNet concepts and focus on typing a single entity mention, based on its surrounding context (e.g. usually in a single sentence) [Choi et al., 2018, Dong et al., 2015, Shimaoka et al., 2017]. Entity typing in fictional domains, on the other hand, requires the model to predict types for entity mentions in long texts (e.g. `Potter` in the whole book *Harry Potter*). Since one entity could be mentioned in multiple sentences, it is essential to design a model that is able to leverage different contexts and consolidate the outputs.

Challenge C4: Relation Extraction in Long Fictional Texts Similar to the entity typing task, relation extraction in fictional domains also has to tackle the same challenge when working on long texts. State-of-the-art methods for relation extraction mostly work on single sentences or short documents. They focus on general encyclopedic knowledge about prominent people, places, etc., and basic relations of wide interest such as birthplace, birthdate, spouses, etc. [Carlson et al., 2010b, Shi and Lin, 2019, Soares et al., 2019, Zhou et al., 2021]. For knowledge on fictional domains, people are more interested in relations that capture traits of characters and key elements of the narrations where training data for them is not available, such as allies, enemies, skills, etc. To extract these relations, it requires the model to handle multiple contexts between each entity pair, across the whole input text (e.g. books). For example, what is the relation between `Harry Potter` and `Severus Snape` in *Harry Potter*? *enemy* or *ally*?

1.3 Contributions

This work addresses the above challenges by developing methods to advance the state of the art:

TiFi We present TiFi [Chu et al., 2019], the first method to construct taxonomies for fictional domains (**Challenge C2**). TiFi uses noisy category systems from fan wikis or text extraction as input and building the taxonomies through three main steps: (i) category cleaning, by identifying candidate categories that truly represent classes in the domain of interest, (ii) edge cleaning, by selecting subcategory relationships that correspond to class subsumption, and (iii) top-level construction, by mapping classes onto a subset of high-level WordNet categories. A comprehensive evaluation shows that TiFi is able to construct taxonomies for a diverse range of fictional domains such as Lord of the Rings, The Simpsons, or Greek Mythology with very high precision and that it outperforms state-of-the-art baselines for taxonomy induction by a substantial margin.

ENTYFI We present ENTYFI [Chu et al., 2020a,b], the first method for typing entities in fictional texts coming from books, fan communities or amateur writers (**Challenge C3**). ENTYFI builds on 205 automatically induced high-quality type systems for popular fictional domains, and exploits the overlap and reuse of these fictional domains for fine-grained typing in previously unseen texts. ENTYFI comprises five steps: type system induction, domain relatedness ranking, mention detection, mention typing, and type consolidation. The recall-oriented typing module combines a supervised neural model,

unsupervised Hearst-style and dependency patterns, and knowledge base lookups. The precision-oriented consolidation stage utilizes co-occurrence statistics in order to remove noise and to identify the most relevant types. Extensive experiments on newly seen fictional texts demonstrate the quality of ENTYFI.

KnowFi We present KnowFi [Chu et al., 2021], for extracting relations between entities coming from very long texts such as books, novels or fan-built wikis (**Challenge C4**). KnowFi leverages semi-structured content in wikis of fan communities on fandom.com (aka wikia.com) to extract initial KBs of background knowledge for 142 popular domains (TV series, movies, games). This serves to identify interesting relations and to collect distant supervision samples. Yet for many relations, this results in very few samples. To overcome this sparseness challenge and to generalize the training across a wide variety of relations, a similarity-based ranking technique is devised for matching seeds in text passages. Given a long input text, KnowFi judiciously selects a number of context passages containing seed pairs of entities. To infer if a certain relation holds between two entities, KnowFi's neural network is trained jointly for all relations as a multi-label classifier. Experiments with several fictional domains demonstrate the gains that KnowFi achieves over the best prior methods for neural relation extraction.

The **challenge C1** is addressed along with other challenges when working on above tasks.

1.4 Publications

Specific results of this work have been published:

- **KnowFi: Knowledge Extraction in Long Fictional Texts**.
  Cuong Xuan Chu, Simon Razniewski, Gerhard Weikum. Proceedings of the 3rd Conference on Automated Knowledge Base Construction, AKBC 2021.

- **ENTYFI: Entity Typing in Fictional Texts**.
  Cuong Xuan Chu, Simon Razniewski, Gerhard Weikum. Proceedings of the 13th ACM International Conference on Web Search and Data Mining, WSDM 2020.

- **ENTYFI: A System for Fine-grained Entity Typing in Fictional Texts**.
  Cuong Xuan Chu, Simon Razniewski, Gerhard Weikum. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020.

- **TiFi: Taxonomy Induction for Fictional Domains**.
  Cuong Xuan Chu, Simon Razniewski, Gerhard Weikum. Proceedings of the Web Conference (the 28th International Conference on World Wide Web), WWW 2019.

  The author of this dissertation is the main author of all these publications. Demonstration, code and data are also published and available at https://www.mpi-inf.mpg.de/yago-naga/fiction-fantasy to accelerate further research in fictional domains.

1.5 Organization

The remainder of this dissertation is organized as follows. Chapter 2 introduces background about knowledge bases and methodologies for sub-tasks in KB construction, which include taxonomy induction, named entity recognition and typing and relation extraction. Three following chapters describe our methods for solving these tasks in fictional domains. Chapter 3 presents a method for taxonomy induction. Chapter 4 presents an end-to-end system for named entity recognition and typing from long fictional texts. Chapter 5 presents a model for relation extraction that overcomes sparsity in training data when working on long texts in fictional domains and Chapter 6 concludes the dissertation with some discussions on open problems for knowledge extraction in fictional domains.

Chapter 2

Background

2.1 Knowledge Bases

2.1.1 Encyclopedic Knowledge Bases

Encyclopedic knowledge represents facts about notable real-world entities such as person, location, organization, etc. A knowledge base that contains this kind of knowledge is called encyclopedic KB or entity-centric KB.

In general, an encyclopedic KB contains three primary pieces of information:

- entities, like people, events, products, organizations such as `Albert Einstein, Joe Biden, Iphone XS Max, WHO`, etc.
- entity types or entity classes to which entities belong, for example, `person, location` at coarse-grained level, or `musicians, left-wing politician` at finer-grained level.
- statements about entities (e.g. relations between entities), for example (`Max Planck` *isFatherOf* `Erwin Planck`), or (`Max Planck` *bornIn* `Kiel`).

Additional information like temporal or spatial information is also presented in several KBs such as YAGO 2 [Hoffart et al., 2013].

Large-scale encyclopedic KBs are YAGO, DBpedia and Wikidata. These KBs have become major assets for enriching search engine and question answering systems. The above KBs are mostly extracted from Wikipedia and enhanced by adding more extracted knowledge from news articles.

2.1.2 Other Knowledge Bases

Along with encyclopedic knowledge, other kinds of knowledge have been also investigated, such as commonsense knowledge, product knowledge, and long-tail domain

knowledge.

Commonsense knowledge embodies facts about classes and concepts, such as properties of concepts (gold *hasProperty* conductivity), relations between concepts (keyboard *partOf* computer), and interaction between concepts (musician *create* song). Popular commonsense KBs are Cyc [Lenat, 1995], ConceptNet [Liu and Singh, 2004], BabelNet [Navigli and Ponzetto, 2010], Webchild [Tandon et al., 2014], Quasimodo [Romero et al., 2019], and ASCENT [Nguyen et al., 2021]. Most of them are extracted from Web content, either manually or automatically, with hundred thousands of concepts and millions of statements.

Product knowledge contains knowledge about products, product types, services, etc. from commercial enterprises. This knowledge has been constructed to help companies manage their internal data, improve customer service and marketing. Some examples are Amazon product graph [Dong et al., 2020], Alibaba E-commerce graph [Luo et al., 2020], or Bloomberg Knowledge Graph [Meij, 2019].

Long-tail domain knowledge contains knowledge about entities from long-tail domains. For example, medical knowledge contains knowledge about medicine, disease, symptoms, etc. Food knowledge presents knowledge about food, dishes, ingredients or receipts. Or cultural knowledge describes information about customs and practices in different countries. Not like other knowledge mentioned above, the sources for extracting long-tail domain knowledge are very sparse and it usually requires domain experts to be involved.

Fiction and fantasy are also archetypes of long-tail domains. Although there is a huge potential, knowledge in fiction and fantasy has not yet received sufficient attention from computer scientists. Section 2.4 describes related work on these domains in detail.

2.1.3 Applications

With structured knowledge extracted from noisy Internet content, knowledge bases have been used in a wide variety of applications and downstream tasks.

Semantic search and question answering: Many commercial search engines incorporate data from KBs to improve their search results. For example, Google uses Google Knowledge Graph, while Bing uses Microsoft Satosi and Facebook uses Graph Search. By taking advantage of these knowledge bases, search-engine systems are able to provide direct answers for queries from users. For instance, an answer "France national football team" is directly given for the query "which team won world cup 2018?" by Google. Since the KBs are entity-centric, a major use case is entity-oriented search, which utilizes

large-scale KBs to improve representations of queries, documents (i.e. web pages), as well as ranking results. In particular, entities from queries are disambiguated by recognizing and linking to existing KBs. In the example, "world cup 2018" is more likely to be linked to `2018 FIFA World Cup`, instead of other events such as `2018 ITTF Team World Cup` or `2018 Athletics World Cup`, hence, a football team is returned. On the other hand, document representation can be enriched by annotating entities (i.e. semantic web) and adding the information into the vector space model [Ensan and Bagheri, 2017, Liu and Fang, 2015, Raviv et al., 2016].

Question answering also leverages data from KBs. IBM Waston used knowledge bases like YAGO and DBpedia in the Jeopardy game show [Ferrucci et al., 2010]. In recent years, many methods of question answering over knowledge bases have been developed. The goals of these tasks are to understand question semantics, reduce the search space and retrieve accurate answers efficiently [Christmann et al., 2019, Wu et al., 2019].

Recommender systems and chatbots: With the advance of artificial intelligence, digital assistants, such as recommender systems and chatbots, have become more and more popular. For example, a user can interact with a recommender system to find a good movie, or communicate with a chatbot to find out what services a store is providing. Using only users' data, such as user-item interactions, is not enough for these systems to be able to work properly. To overcome the issue, recent systems and studies start to consider KBs as a source for background information. Large-scale KBs such as Wikidata have become good choices [Gao et al., 2021, Jannach et al., 2020]. Social chatbots and digital assistants such as Cortana, Siri or Alexa use KBs as key assets. Many e-commerce companies also construct their own KBs to improve customer services, such as Amazon and Alibaba.

Text and visual understanding: With a lot of ambiguities on texts, downstream tasks and applications need to understand the meaning of the input text. For example, a user asks a digital assistant to "play some songs of Monkees member David Jones". In this case, the assistant knows that the mention "David Jones" should be linked to `David Jones` (aka `Davy Jones`), a member of Monkees. However, if the user only asks to "play some songs of David Jones", how does the assistant know which entity the mention "David Jones" should be linked to, `Davy Jones` (member of Monkees) or singer `David Jones` (aka `David Bowie`)? KBs are the key assets to distinguish the meanings of the input words. For instance, WordNet, a lexical database for English, contains synsets of hundred thousands of English concepts with their descriptions. Commonsense KBs such as ConceptNet, Webchild, contain millions of concepts, along with their properties.

Entity-centric KBs such as YAGO, DBpedia, contain millions of unique entities. Word and entity disambiguation is not only useful for digital assistants, but also for other downstream tasks such as search, question answering or machine translation [Shen et al., 2014].

Although recent works on visual understanding, such as object detection, have achieved impressive results with the advances of deep learning, leveraging external knowledge can further improve the performance of deep learning models. For example, with common-sense knowledge, the model should be able to learn that a tennis racket usually appears along with a tennis ball, and not along with other similar objects like a lemon or an orange [Chowdhury et al., 2019, Nag Chowdhury et al., 2021].

2.2 Knowledge Base Construction

2.2.1 Manual Construction

The idea of constructing a knowledge base was first pursued in the 1980s, with Cyc [Lenat, 1995] being a seminal project. By manually construction, Cyc contained hundred thousands of concepts and millions of facts.

WordNet [Fellbaum and Miller, 1998] is a lexical database for English. WordNet describes the relations between concept synsets, which include synonymy, hypo-hypernymy, and mero-holonymy. The most recent WordNet database contains more than 155k words which belong to more than 117k synsets and the number of word-sense pairs is over 200k. WordNet is carefully handcrafted and has high accuracy, but low coverage of concepts and statements. VerbNet [Schuler, 2005] is also a lexical database for English, which focuses on English verbs and is compatible with WordNet .

With the advances of the internet, people are able to collaborate with others on such projects. Wikidata [Vrandečić and Krötzsch, 2014] is a project that was established based on this idea. By providing a free open API, Wikidata can be read and edited by both humans and machines. Wikidata contains more than 95M data items with almost 10k predicates and millions of facts. Wikidata can be considered as the largest project on constructing KBs with over 25k active users and over 1.5B edits that have been made since the project launched.

By manually constructing, the advantages of these systems are having high quality and easily maintained. However, due to high cost and much time consuming, they are not scalable and have low coverage.

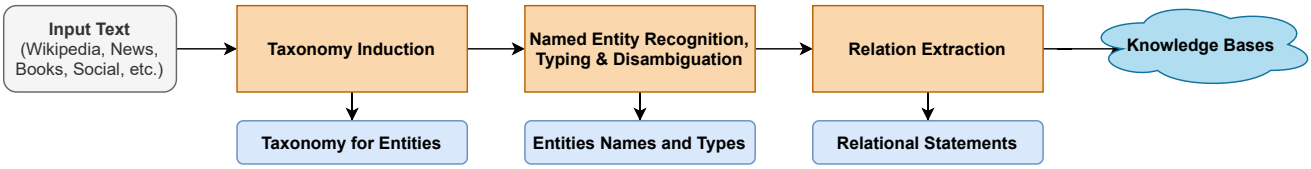


Figure 2.1: A general framework for automated knowledge extraction.

2.2.2 Automated KB Construction

Since the late 2000s, there is a variety of knowledge bases that have been built automatically, such as YAGO, DBpedia, Freebase, ConceptNet, BabelNet, NELL, WebIsALOD, etc. Compared to handcrafted KBs, these KBs are much larger, with millions of entities, hundred thousands of entity types and hundred millions to billions of assertions.

The output of automated extraction methods is usually represented as in one of the following two types: schema-free and schema-based. Since concepts and relations in a schema-free KB do not follow any ontology, it is hard to infer new knowledge from existing knowledge. Most of KBs, especially encyclopedic KBs, therefore, are schema-based, where components follow a specific ontology (e.g. relations between entities are pre-defined, the entity types are pre-defined, etc.). Figure 2.1 shows a basic framework to construct schema-based knowledge. The following subsections give an overview on state-of-the-art methods for each task in the framework in detail.

Taxonomy Induction

Taxonomies, also known as type systems or class subsumption hierarchies, are an important resource for a variety of tasks and a core piece in knowledge graphs. Taxonomy induction, hence, is a common problem that has been explored in many works [de Melo and Weikum, 2010, Flati et al., 2014, Gupta et al., 2016b, 2017b, Ponzetto and Strube, 2007], which can be classified based on two dimensions: input source and model. Figure 2.2 shows design space for the taxonomy induction task.

In the timeline of taxonomy induction, using Hearst patterns [Hearst, 1992] seems to be the earliest method. With the simple patterns such as *"X is a Y"*, *"X such as Y and Z"*, the method is able to achieve very high precision when working on unstructured texts and still part of other advanced approaches.

With the rapid expansion of Wikipedia, there is a variety of methods that use Wikipedia as the input for taxonomy induction. Along with encyclopedic information about entities, Wikipedia also provides categories, which groups Wikipedia pages and other related categories as well. The categories can be organized as a directed graph, and are often

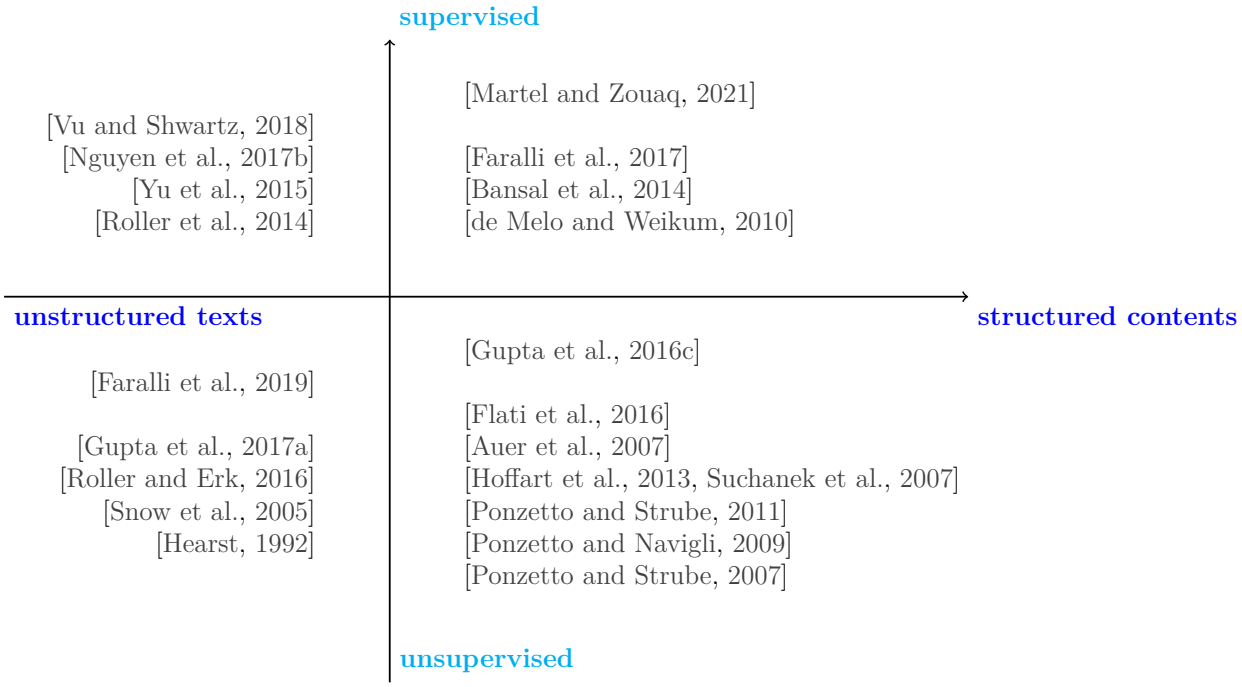


Figure 2.2: Design space for taxonomy induction.

referred to as Wikipedia category network (WCN). By leveraging the information from WCN and other existing ontologies like WordNet, these methods are able to construct large-scale full-fledged ontologies with high accuracy. Some notable works are WikiTaxonomy [Ponzetto and Navigli, 2009, Ponzetto and Strube, 2007, 2011], WikiNet [Nastase et al., 2010], YAGO, DBpedia, MENTA [de Melo and Weikum, 2010], MultiWibi [Flati et al., 2014] and HEAD [Gupta et al., 2016c]. Among these works, MENTA [de Melo and Weikum, 2010] was one of the largest multilingual lexical knowledge bases with over 5.4 million entities in more than 270 languages. In the case of English only, ProBase [Wu et al., 2012a] contains over 20 million *isA* pairs between over 2.6 million concepts.

With advanced deep neural models, many recent approaches utilize distributional representations of entity types [Nguyen et al., 2017b, Roller et al., 2014, Vu and Shwartz, 2018, Yu et al., 2015], and classify hypernym relations between the entity type pairs using supervised techniques. Some of the methods leverage existing knowledge graphs, like YAGO, DBpedia and learn their embeddings to automatically extract the taxonomies [Martel and Zouaq, 2021].

Named Entity Recognition and Typing (NER)

Named entity recognition is the task of identifying named entities in natural language texts and classifying them into coarse-grained semantic types such as person, location,

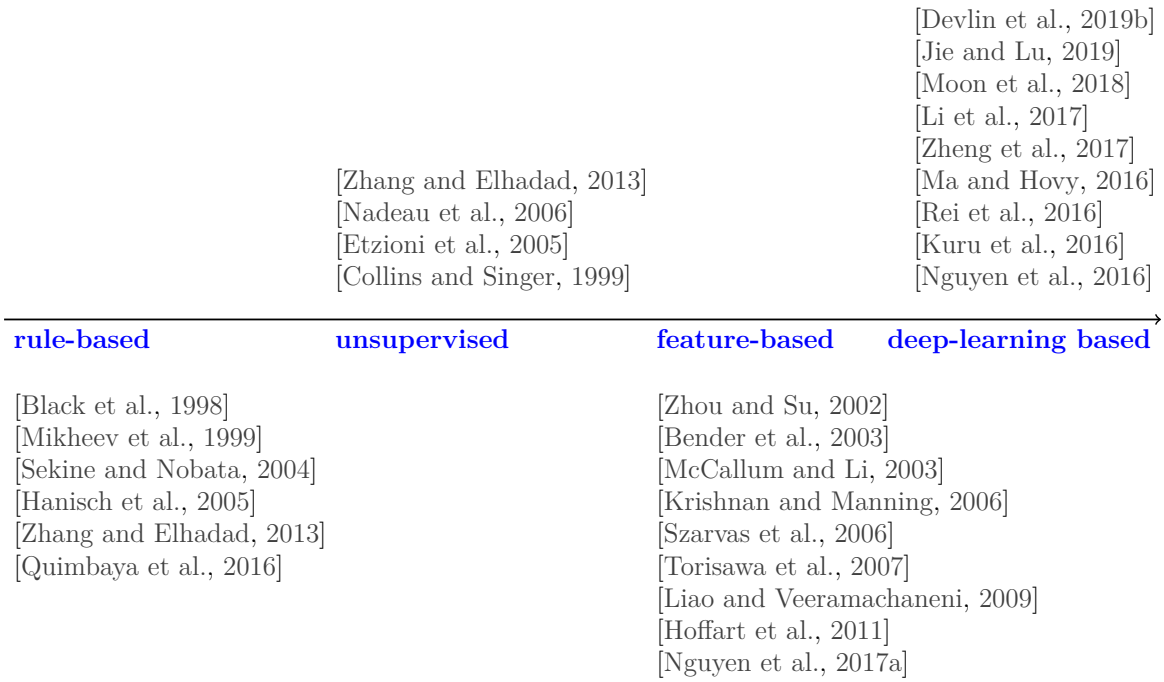


Figure 2.3: Design space for named entity recognition.

organization and misc [Collins and Singer, 1999, Grishman and Sundheim, 1996, Li et al., 2020a, Zhang and Elhadad, 2013, Zheng et al., 2017]. NER has been investigated since the 90s and achieved remarkable results, along with the development of machine learning. Figure 2.3 shows the design space for NER, which can be classified into four main streams: 1) rule-based approaches, 2) unsupervised learning approaches, 3) feature-based supervised learning approaches, and 4) deep-learning based approaches.

Rule-based approaches usually design hand-crafted semantic and syntactic rules to recognize entities [Black et al., 1998, Hanisch et al., 2005, Zhang and Elhadad, 2013]. These rules are based on domain-specific dictionaries and syntactic-lexical patterns. However, due to insufficiency in dictionaries, these methods often achieve low recall and are hardly transferred to other domains.

Unsupervised learning approaches, like clustering, recognize named entities by computing their context similarity with other "seed" entities. The similarity score is usually based on lexical form (the noun phrase and its surrounding context) and statistics (e.g. frequency, context vectors) from a large corpus [Collins and Singer, 1999, Nadeau et al., 2006, Zhang and Elhadad, 2013].

Feature-based approaches, on the other hand, exploit different features, such as morphology, part-of-speech tags, dependency relations, and use machine learning algorithms such as Hidden Markov Models (HMM), Support Vector Machines (SVM) or Conditional

Random Fields (CRF), to cast NER into a sequence tagging task or a multi-class classification problem [Hoffart et al., 2011, McCallum and Li, 2003, Torisawa et al., 2007, Zhou and Su, 2002]. Among these models, CRF-based NER has been widely applied, not only in mainstream texts, but also in domain-specific texts, such as medical texts [Funk et al., 2014], chemical texts [Rocktäschel et al., 2012] or product-related texts [Shang et al., 2018].

Similar to other NLP tasks, deep-learning based approaches have gained much attention recently, and also achieve state-of-the-art results on NER [Devlin et al., 2019b, Nguyen et al., 2016, Zheng et al., 2017]. With sufficient training data, deep-learning models are able to exploit hidden features without engineering. Input of deep-learning models are usually distributional representations of texts, such as word-level representation [Nguyen et al., 2016, Zheng et al., 2017], character-level representation [Kuru et al., 2016] or contextualized language-model embeddings [Devlin et al., 2019b]. Models for NER vary from convolutional neural networks (CNN) [Strubell et al., 2017, Yao et al., 2015], to recurrent neural networks (e.g. gated recurrent unit – GRU, long-short term memory – LSTM) [Ju et al., 2018, Katiyar and Cardie, 2018, Ma and Hovy, 2016] and deep transformers (e.g. transformer, BERT) [Devlin et al., 2019b, Vaswani et al., 2017].

Named entity typing is the task of identifying semantic classes for named entities in textual contexts. While NER focuses on recognition of the entities and distinguishes them into a few coarse-grained types such as `person, organization, location`, named entity typing usually works on a fine-grained level, where entity mentions are classified into hundreds to thousands of types [Choi et al., 2018, Lee et al., 2006, Ling and Weld, 2012]. Figure 2.4 shows the design space for named entity typing.

As other extraction tasks, pattern-based approaches design specific patterns that describe relations between entity mentions and classes in texts. For example, a text snippet like "hobbits such as Frodo and Sam" suggests that Frodo and Sam belong to the class `hobbit`. This pattern and other similar patterns are well known as Hearst patterns and are widely used, especially when the type system is not available [Hearst, 1992, Seitner et al., 2016]. On the other hand, supervised typing has gained more attention when the taxonomies (e.g. type systems) are pre-defined. These methods leverage information about surrounding contexts of entity mentions as features to classify entity mentions. The features consist of lexical, syntactic and semantic features [Corro et al., 2015, Ling and Weld, 2012, Yogatama et al., 2015, Yosef et al., 2012]. Recently, more neural methods are being investigated on entity typing and able to classify entity mentions into hundreds to thousands of types [Choi et al., 2018, Shimaoka et al., 2017, Xiong et al.,

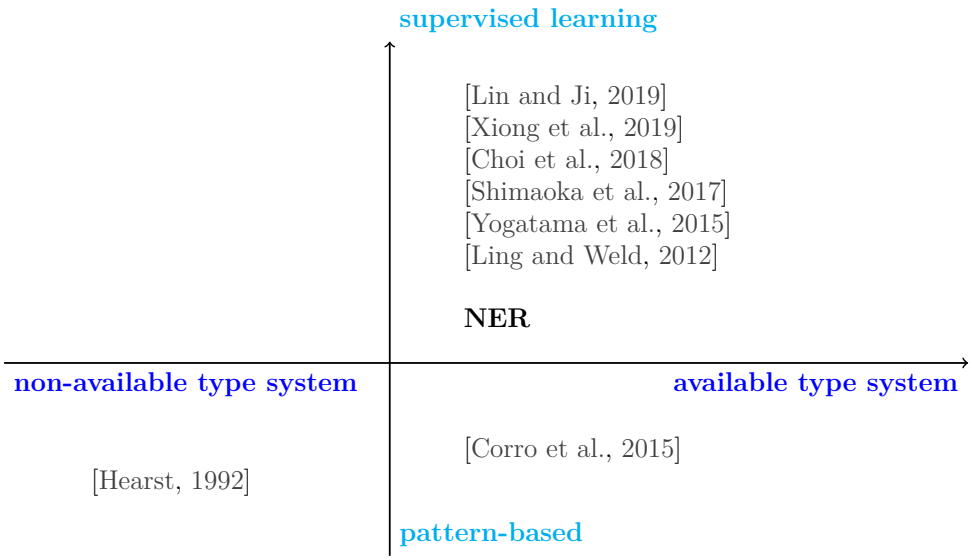


Figure 2.4: Design space for named entity typing.

2019]. Some notable neural models are LSTM with attention mechanism [Choi et al., 2018, Lin and Ji, 2019, Shimaoka et al., 2017] and deep transformers [Eberts et al., 2020, Onoe and Durrett, 2019, 2020].

Relation Extraction

Relation extraction (RE) is the task of identifying semantic relations between two given entities. The input of the task is either semi-structured texts like infoboxes from Wikipedia pages, or unstructured texts like Wikipedia pages and news articles. Based on the input, a wide range of methods have been proposed, which can be classified into two main classes: pattern-based approaches [Carlson et al., 2010a, Kim and Moldovan, 1995, Nakashole et al., 2012, Soderland et al., 1995] and supervised approaches, where deep-learning models currently achieve state-of-the-art results [Han et al., 2020a, Soares et al., 2019, Wang et al., 2020, Zhou et al., 2021]. Figure 2.5 shows design space for relation extraction.

Early methods on RE utilize lexical and syntactic structure from text to manually design patterns. In the case of semi-structure texts, these patterns are induced by using web scraping [Auer et al., 2007, Hoffart et al., 2013]. For example, Figure 2.6 shows a snapshot from the Wikia infobox of the entity Zeus in Greek mythology and the Wiki markup table extracted from the dump file of the Wiki page [1]. In the case of unstructured texts, the lexical and dependency features are usually used to construct the

[1] https://greekmythology.wikia.org/wiki/Zeus

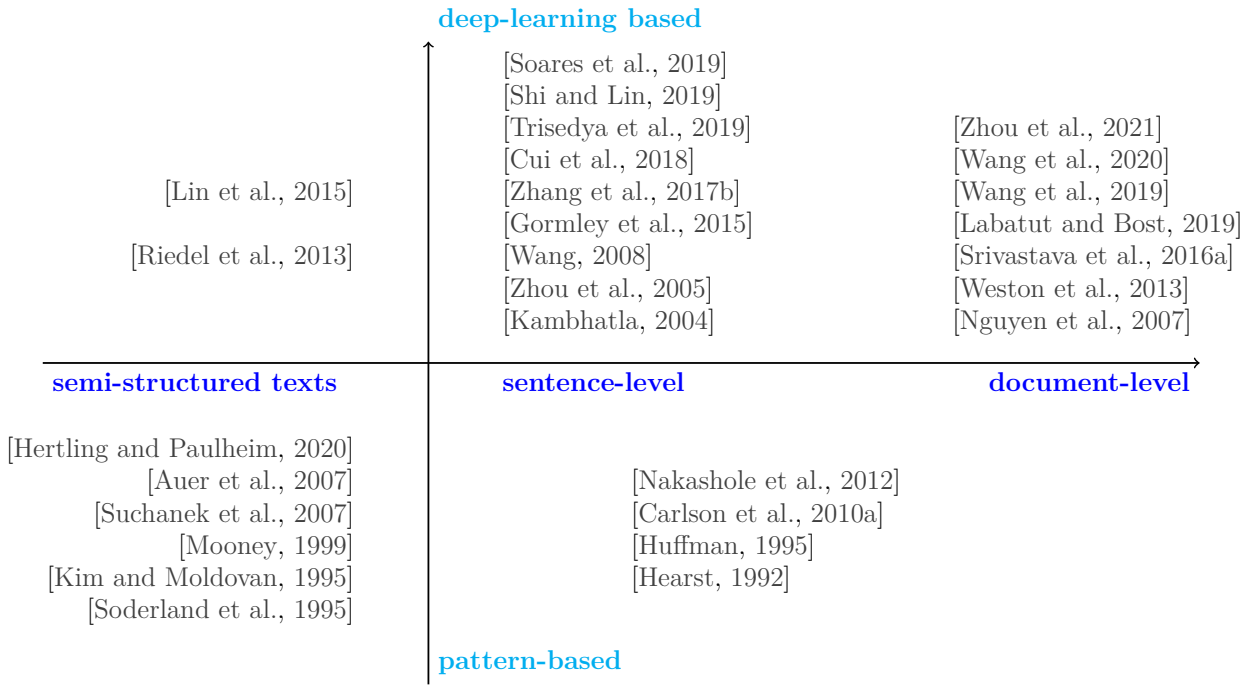
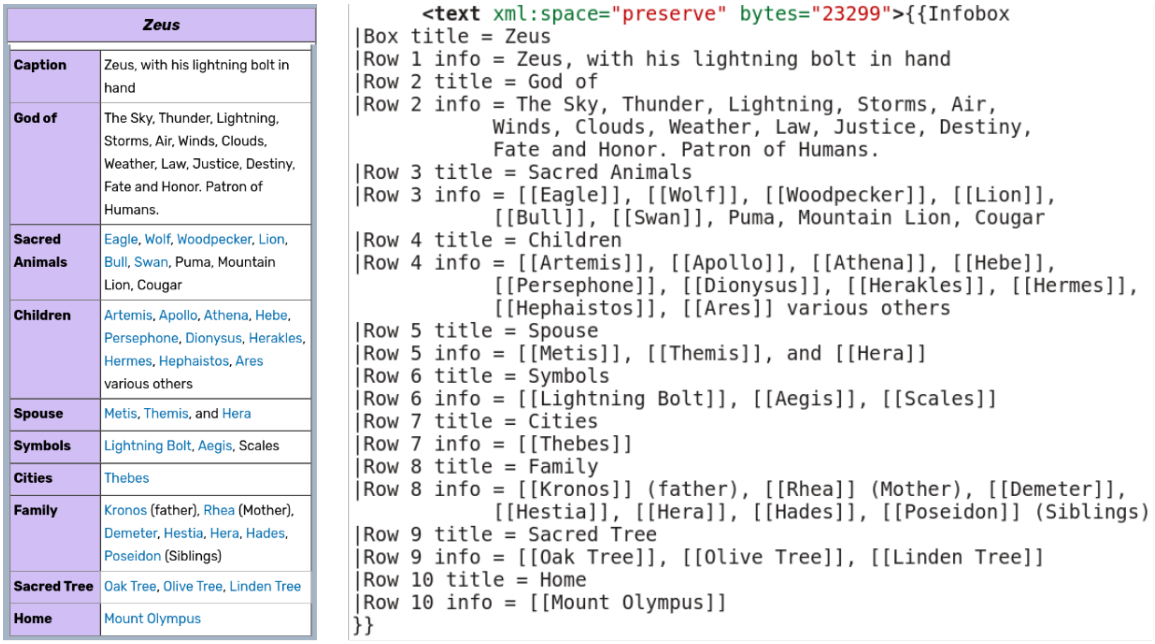


Figure 2.5: Design space for relation extraction.

patterns. For example, a regular expression `<A .* born in .* B>` indicates that entity *A hasBirthplace* `B`, or a simple entity-type-based pattern `<PERSON (write(s?)|wrote) BOOK>` indicates the relation *hasAuthor* between a book and a person. The drawback of pattern-based methods is requiring intervention from human experts, hence costly and not scalable.

Supervised approaches, on the other hand, are more scalable and require less human effort. In terms of models, supervised approaches can be classified into two types: feature-based approaches and deep-learning based approaches. Feature-based approaches design lexical, syntactic and semantic features for the entity pairs, based on their surrounding context, and use these features in the classification models, such as logistic regression, support vector machine or graphical models [Kambhatla, 2004, Nguyen et al., 2007, Zhou et al., 2005]. In contrast, deep-learning based approaches do not require feature engineering and are able to automatically extract hidden semantic features from the text. With the advance of neural network models, a variety of models have been proposed and are able to work on texts with different granularities that include sentence-level RE and document-level RE. For example, *convolutional neural networks (CNNs)* [Wang et al., 2016, Zeng et al., 2014] work on short sequence text, with a fix window size of length, *recurrent neural networks (RNNs)* [Lee et al., 2019, Xu et al., 2016, Zhang et al., 2015] work on longer sequence text, *attention-based neural networks* [Guo et al., 2019, Lin



Zeus	
Caption	Zeus, with his lightning bolt in hand
God of	The Sky, Thunder, Lightning, Storms, Air, Winds, Clouds, Weather, Law, Justice, Destiny, Fate and Honor. Patron of Humans.
Sacred Animals	Eagle, Wolf, Woodpecker, Lion, Bull, Swan, Puma, Mountain Lion, Cougar
Children	Artemis, Apollo, Athena, Hebe, Persephone, Dionysus, Herakles, Hermes, Hephaistos, Ares various others
Spouse	Metis, Themis, and Hera
Symbols	Lightning Bolt, Aegis, Scales
Cities	Thebes
Family	Kronos (father), Rhea (Mother), Demeter, Hestia, Hera, Hades, Poseidon (Siblings)
Sacred Tree	Oak Tree, Olive Tree, Linden Tree
Home	Mount Olympus

```
      <text xml:space="preserve" bytes="23299">{{Infobox
|Box title = Zeus
|Row 1 info = Zeus, with his lightning bolt in hand
|Row 2 title = God of
|Row 2 info = The Sky, Thunder, Lightning, Storms, Air,
            Winds, Clouds, Weather, Law, Justice, Destiny,
            Fate and Honor. Patron of Humans.
|Row 3 title = Sacred Animals
|Row 3 info = [[Eagle]], [[Wolf]], [[Woodpecker]], [[Lion]],
            [[Bull]], [[Swan]], Puma, Mountain Lion, Cougar
|Row 4 title = Children
|Row 4 info = [[Artemis]], [[Apollo]], [[Athena]], [[Hebe]],
            [[Persephone]], [[Dionysus]], [[Herakles]], [[Hermes]],
            [[Hephaistos]], [[Ares]] various others
|Row 5 title = Spouse
|Row 5 info = [[Metis]], [[Themis]], and [[Hera]]
|Row 6 title = Symbols
|Row 6 info = [[Lightning Bolt]], [[Aegis]], [[Scales]]
|Row 7 title = Cities
|Row 7 info = [[Thebes]]
|Row 8 title = Family
|Row 8 info = [[Kronos]] (father), [[Rhea]] (Mother), [[Demeter]],
            [[Hestia]], [[Hera]], [[Hades]], [[Poseidon]] (Siblings)
|Row 9 title = Sacred Tree
|Row 9 info = [[Oak Tree]], [[Olive Tree]], [[Linden Tree]]
|Row 10 title = Home
|Row 10 info = [[Mount Olympus]]
}}
```

Figure 2.6: `Zeus` infobox from Greek Mythology.

et al., 2016] emphasize weight on specific positions in text (e.g. attention mechanism), and *graph-based neural networks (GNNs)* that build entity graphs from text, work on long texts and are able to infer global relations between entity pairs [Wang et al., 2020, Zhou et al., 2021]. Inputs for deep-learning models are usually semantic representations of words (e.g. word embeddings that are learned from pre-trained language models) and position embeddings of words in the context [Mikolov et al., 2013, Zhang et al., 2017a]. With respect to the performance, Transformers [Vaswani et al., 2017] and BERT [Devlin et al., 2019b] have recently achieve new start-of-the-art results on relation extraction.

Different from pattern-based models, supervised approaches require training data, especially for the tasks with pre-specified relations. Besides manually creating training data [Zhang et al., 2017b], a large number of methods use distant supervision techniques to collect more training data [Mintz et al., 2009, Suchanek et al., 2009, Yao et al., 2019]. Distant supervision leverages existing knowledge from KGs to collect positive training samples. The idea is that, for any entity pair with relation r in KGs, if a text (e.g. sentence or paragraph) mentions both of them, the text can be considered as one positive training sample for the relation r. However, producing many false positives is the main drawback of distant supervision. To be able to overcome this, some methods have been proposed to denoise distant supervision, such as selecting informative instances in each training batch or from a batch of instances with the same entity pairs [Li et al., 2020b, Riedel et al., 2010], or incorporating with information from other resources (e.g.

knowledge bases, multilingual datasets) [Ji et al., 2017, Wang et al., 2018].

2.3 Input Sources

Having been investigated for a long time, automated methods leverage a wide range of sources as input for knowledge extraction. In general, these sources can be classified into four main categories as follows.

1. **Handcrafted Data.** This kind of data is manually created by human experts with high quality and clean structure, for example, WordNet and Wikidata. It can be used as seed knowledge to collect more knowledge from other sources (e.g. with distant supervision).

2. **Semi-structured Data.** Not as high quality as handcrafted data, semi-structured data addresses the problem of scalability with better coverage and sufficient quality. The most prominent data in this setting comes from Wikipedia, which includes Wikipedia category networks, infoboxes of entity pages, and other formats like tables and lists. With semi-structured input, knowledge can be extracted by using pattern-based approaches.

3. **Unstructured Data.** Most of the text data on the internet is unstructured, spanning from new articles, web pages into text documents like books, movie scripts or technical descriptions, etc. Knowledge extraction from these sources requires advanced models that are able to infer semantics from text.

4. **Social Media.** Text from online users on social media platforms like social networks, discussion forums, etc., can be classified as unstructured data. However, the average length of text sequences from these sources is usually short and knowledge that is expressed in these texts is quite noisy and sparse. Dealing with these texts requires more cleaning processes and a large amount of data.

Wikipedia Wikipedia is the most popular and richest source for knowledge extraction. It contains encyclopedic knowledge of millions of entities and across over three hundred languages. Wikipedia organizes its pages following a category network which becomes rich resources for taxonomy induction. Each entity page in Wikipedia also contains an infobox that stores basic information about the entity. With semi-structured format, Wikipedia infoboxes are great resources for knowledge or relation extraction. The content in Wikipedia pages is written using the Wiki markup language. With the crisp

content, text from Wikipedia is valuable for entity recognition, disambiguation and linking, and relation extraction. Many large KBs have been built from Wikipedia, such as YAGO, DBpedia, Freebase, etc. However, Wikipedia favors entities in the real world, so that it lacks knowledge in long-tail domains where fiction and fantasy are typical examples.

Wikia (Fandom) Wikia or Fandom [2] is the largest web platform for organized fan communities for fictional universes. As of July 2018, its Alexa rank is 49 worldwide (and 19 in the US). It contains over 380,000 fan-built communities. For example, the *The Lord of the Rings* universe[3] contains 6,229 content pages, while the *Star Wars* universe contains more than 170,000. Wikia is also constructed similarly to Wikipedia, with each universe is organized as a Wiki, so that it also contains pages of entities in the universe, infoboxes and category networks. With tremendous contribution from fans on creating the content, Wikia has become a great source for knowledge extraction in fictional domains [Hertling and Paulheim, 2020].

2.4 NLP for Fictional Texts

With a huge interest in fiction and fantasy, various aspects related to fictional texts have been investigated, especially in literature and culture studies [Labatut and Bost, 2019]. Along with narrative extraction (e.g. storyline analysis), character network extraction is one of the most popular tasks that have been tackled in these domains. This task involves several sub-tasks such as character detection, character interaction detection, and character graph construction. Figure 2.7 shows an overview of the basic character network extraction process [Labatut and Bost, 2019].

In particular, Vala et al. [2015] proposed a graph-based model to detect characters and their occurrences in novels, while a number of authors apply traditional NER systems to run on novels and only keep `PERSON` entities as character names [Chaturvedi et al., 2017, Elson et al., 2010, Srivastava et al., 2016a]. Very few works consider other categories such as `LOCATION` or `ORGANIZATION` [Labatut and Bost, 2019]. In the case of relation extraction, many works focus on character networks whether the characters have the same occurrences, conversations, or directly interact with each other [Chaturvedi et al., 2016a, Makazhanov et al., 2014, Srivastava et al., 2016a]. Makazhanov et al. [2014] propose a heuristic approach to detect family relations between characters. Based on vocative

[2] www.fandom.com
[3] https://lotr.fandom.com/

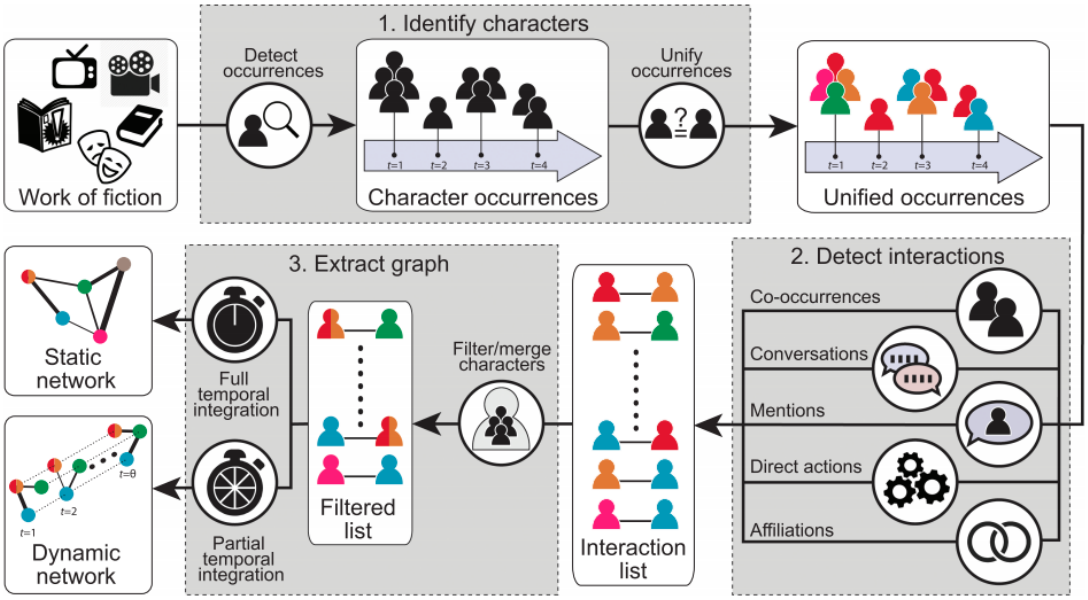


Figure 2.7: Overview of the basic character network extraction process [2019].

utterances, the relation candidates are filtered by manual constraints. Chaturvedi et al. [2016a] present a Markov model to capture interactions between characters and detect friendly vs. hostile signals. Srivastava et al. [2016a] leverage both text-based and structural cues for learning a model to infer interpersonal relations in narrative summaries. The common between all of above methods is taking books or fan fiction as the input source.

In the case of leveraging the richness of Wikia, DBkWik [Hertling and Paulheim, 2018, Hofmann et al., 2017] uses the DBpedia framework to extract a knowledge graph from thousands of Wikis. The framework focuses on extracting information from semi-structured sources such as infoboxes or wiki category networks of Wikia pages.

Chapter 3

TiFi: Taxonomy Induction for Fictional Domains

3.1 Introduction

3.1.1 Motivation and Problem

Taxonomy Induction: Taxonomies, also known as type systems or class subsumption hierarchies, are an important resource for a variety of tasks related to text comprehension, such as information extraction, entity search or question answering. They represent structured knowledge about the subsumption of classes, for instance, that `electric guitar players` are `rock musicians` and that `state governors` are `politicans`. Taxonomies are a core piece of large knowledge graphs (KGs) such as DBpedia, Wikidata, Yago and industrial KGs at Google, Microsoft Bing, Amazon, etc. When search engines receive user queries about classes of entities, they can often find answers by combining instances of taxonomic classes. For example, a query about "left-handed electric guitar players" can be answered by intersecting the classes `left-handed people`, `guitar players` and `rock musicians`; a query about "actors who became politicans" can include instances from the intersection of `state governors` and `movie stars` such as Schwarzenegger. Also, taxonomic class systems are very useful for type-checking answer candidates for semantic search and question answering [Kalyanpur et al., 2011].

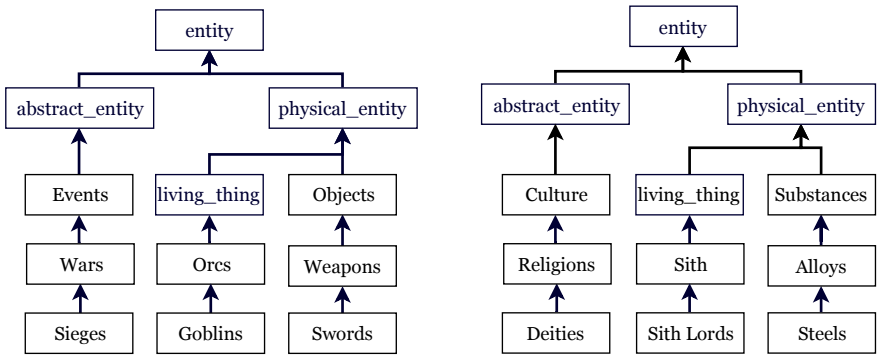
Taxonomies can be hand-crafted, examples being WordNet [Fellbaum and Miller, 1998], SUMO [Niles and Pease, 2001] or MeSH and UMLS [Bodenreider, 2004], or automatically constructed by *taxonomy induction* from textual or semi-structured cues about type instances and subtype relations. Methods for the latter include text mining using Hearst patterns [Hearst, 1992] or bootstrapped with Hearst patterns (e.g., [Wu et al., 2012b]), harvesting and learning from Wikipedia categories as a noisy seed network (e.g.,

[de Melo and Weikum, 2010, Flati et al., 2014, Gupta et al., 2016c, Ponzetto and Navigli, 2009, Ponzetto and Strube, 2007, 2011, Suchanek et al., 2007, Wu et al., 2008]), and inducing type hierarchies from query-and-click logs (e.g., [Gupta et al., 2014, Pasca, 2013, Pasca and Durme, 2007]).

The Case for Fictional Domains: Fiction and fantasy are a core part of human culture, spanning from traditional literature to movies, TV series and video games. Well known fictional domains are, for instance, the Greek mythology, the Mahabharata, Tolkien's Middle-earth, the world of Harry Potter, or the Simpsons. These universes contain many hundreds or even thousands of entities and types, and are subject of search-engine queries – by fans as well as cultural analysts. For example, fans may query about Muggles who are students of the House of Gryffindor (within the Harry Potter universe). Analysts may be interested in understanding character relationships [Bamman et al., 2014, Iyyer et al., 2016, Srivastava et al., 2016b], learning story patterns [Chambers and Jurafsky, 2009, Chaturvedi et al., 2017] or investigating gender bias in different cultures [Agarwal et al., 2015]. Thus, organizing entities and classes from fictional domains into clean taxonomies (see example in Fig. 3.1) is of great value.

Challenges: While taxonomy construction for encyclopedic knowledge about the real world has received considerable attention already, taxonomy construction for fictional domains is a new problem that comes with specific challenges:

1. State-of-the-art methods for taxonomy induction make assumptions on entity-class and subclass relations that are often invalid for fictional domains. For example, they assume that certain classes are disjoint (e.g., living beings and abstract entities, the oracle of Delphi being a counterexample). Also, assumptions about the surface forms of entity names (e.g., on person names: with or without first name, starting with



(a) LoTR (b) Star Wars

Figure 3.1: Excerpts of LoTR and Star Wars taxonomies.

Mr., Mrs., Dr., etc.) and typical phrases for classes (e.g., noun phrases in plural form) do not apply to fictional domains.

2. Prior methods for taxonomy induction intensively leveraged Wikipedia categories, either as a content source or for distant supervision. However, the coverage of fiction and fantasy in Wikipedia is very limited, and their categories are fairly ad-hoc. For example, Lord Voldemort is in categories like `Fictional cult leaders` (i.e., people), `J.K. Rowling characters` (i.e., a meta-category) and `Narcissism in fiction` (i.e., an abstraction). And whereas Harry Potter is reasonably covered in Wikipedia, fan websites feature many more characters and domains such as House of Cards (a TV series) or Hyperion Cantos (a 4-volume science fiction book) that are hardly captured in Wikipedia.
3. Both Wikipedia and other content sources like fan-community forums cover an ad-hoc mixture of in-domain and out-of-domain entities and types. For example, they discuss both the fictional characters (e.g., Lord Voldemort) and the actors of movies (e.g., Ralph Fiennes) and other aspects of the film-making or book-writing.

The same difficulties arise also when constructing enterprise-specific taxonomies from highly heterogeneous and noisy contents, or when organizing types for highly specialized verticals such as medieval history, the Maya culture, neurodegenerative diseases, or nano-technology material science. Methodology for tackling such domains is badly missing. We believe that our approach to fictional domains has great potential for being carried over to such real-life settings. This work focuses on fiction and fantasy, though, where raw content sources are publicly available.

3.1.2 Approach and Contribution

In this work we develop the first taxonomy construction method specifically geared for fictional domains. We refer to our method as the **TiFi** system, for **T**axonomy **i**nduction for **Fi**ction. We address Challenge 1 by developing a classifier for categories and subcategory relationships that combines rule-based lexical and numerical contextual features. This technique is able to deal with difficult cases arising from non-standard entity names and class names. Challenge 2 is addressed by tapping into fan community Wikis (e.g., `harrypotter.wikia.com`). This allows us to overcome the limitations of Wikipedia. Finally, Challenge 3 is addressed by constructing a supervised classifier for distinguishing in-domain vs. out-of-domain types, using a feature model specifically designed for fictional domains.

Moreover, we integrate our taxonomies with an upper-level taxonomy provided by

WordNet, for generalizations and abstract classes. This adds value for searching by entities and classes. Our method outperforms the state-of-the-art taxonomy induction system for the first two steps, HEAD [Gupta et al., 2016c], by 21-23% and 6-8% percentage points in F1-score, respectively. An extrinsic evaluation based on entity search shows the value that can be derived from our taxonomies, where, for different queries, our taxonomies return answers with 24% higher precision than the input category systems. TiFi datasets are available at https://www.mpi-inf.mpg.de/index.php?id=3971.

3.2 Related Work

Text Analysis and Fiction Analysis and interpretation of fictional texts are an important part of cultural and language research, both for the intrinsic interest in understanding themes and creativity [Chambers and Jurafsky, 2009, Chaturvedi et al., 2017], and for extrinsic reasons such as predicting human behaviour [Fast et al., 2016] or measuring discrimination [Agarwal et al., 2015]. Other recurrent topics are, for instance, to discover character relationships [Bamman et al., 2014, Iyyer et al., 2016, Srivastava et al., 2016b], to model social networks [Bamman et al., 2014, Elangovan and Eisenstein, 2015], or to describe personalities and emotions [Elson et al., 2010, Jhavar and Mirza, 2018]. Traditionally requiring extensive manual reading, automated NLP techniques have recently lead to the emergence of a new interdisciplinary subject called *Digital Humanities*, which combines methodologies and techniques from sociology, linguistics and computational sciences towards the large-scale analysis of digital artifacts and heritage.

Taxonomy Induction from Text Taxonomies, that is, structured hierarchies of classes within a domain of interest, are a basic building block for knowledge organization and text processing, and crucially needed in tasks such as entity detection and linking, fact extraction, or question answering. A seminal contribution towards their automated construction was the discovery of Hearst patterns [Hearst, 1992], simple syntactic patterns like *"X is a Y"* that achieve remarkable precision, and are conceptually still part of many advanced approaches. Subsequent works aim to automate the process of discovering useful patterns [Roller and Erk, 2016, Snow et al., 2005]. Recent work by Gupta et al. [Gupta et al., 2017a] uses seed terms in combination with a probabilistic model to extract hypernym subsequences, which are then put into a directed graph from which the final taxonomy is induced by using a minimum cost flow algorithm. Other approaches utilize distributional representations of types [Nguyen et al., 2017b, Roller et al., 2014, Vu and Shwartz, 2018, Yu et al., 2015], or aim to learn them pairwise [Yu et al., 2015]

or hierarchically [Nguyen et al., 2017b].

Taxonomy Construction using Wikipedia A popular structured source for taxonomy construction is the Wikipedia category network (WCN) for taxonomy induction. The WCN is a collaboratively constructed network of categories with many similarities to taxonomies, expressing for instance that the category `Italian 19th century composers` is a subcategory of `Italian Composers`. One project, WikiTaxonomy [Ponzetto and Strube, 2007, 2011] aims to classify subcategory relations in the WCN as *subclass* and *not-subclass* relations. They investigate heuristics based on lexical matching between categories, lexico-syntactic patterns and the structure of the category network for that purpose. YAGO [Hoffart et al., 2013, Suchanek et al., 2007] uses a very simple criterion to decide whether a category represents a class, namely to check whether it is in plural form. It also provides linking to WordNet [Fellbaum and Miller, 1998] categories, choosing in case of ambiguity simply the meaning appearing topmost in WordNet. MENTA [de Melo and Weikum, 2010] learns a model to map Wikipedia categories to WordNet, with the goal of constructing a multilingual taxonomy over both. MENTA creates mean edges and subclass edges between categories and entities across languages, then uses Markov chains to rank edges and induce the final taxonomy. WiBi (Wikipedia Bitaxonomy) [Flati et al., 2014] proceeds in two steps: It first builds a taxonomy from Wikipedia pages by extracting lemmas from the first sentence of pages, and heuristically disambiguating them and linking them to others. In the second step, WiBi combines the page taxonomy and the original Wikipedia category network to induce the final taxonomy. The most recent effort working on taxonomy induction over Wikipedia is HEAD [Gupta et al., 2016c]. HEAD exploits multiple lexical and structural rules towards classifying subcategory relations, and is judiciously tailored towards high-quality extraction from the WCN.

Domain-specific Taxonomies TAXIFY is an unsupervised approach to domain-specific taxonomy construction from text [Alfarone and Davis, 2015]. Relying on distributional semantics, TAXIFY creates subclass candidates, which in a second step are filtered based on a custom graph algorithm. Similarly, Liu et al. [Liu et al., 2012] construct domain-specific taxonomies from keyword phrases augmented with relative knowledge and contexts. Compared with taxonomy construction from structured resources, these text-based approaches usually deliver comparably flat taxonomies.

Fan Wikis Fans are organizing content on fictional universes on a multitude of webspaces. Particularly relevant for our problem are fan Wikis, i.e., community-built web

content constructed using generic Wiki frameworks. Some notable examples of such Wikis are *tolkiengateway.net/wiki*, with 12k articles, *www.mariowiki.com* with 21k articles, or *en.brickimedia.org* with 29k articles. Particularly relevant are also Wiki farms, like Wikia[1] and Gamepedia[2], which host Wikis for 380k and 2k different fictional universes, and have Alexa rank 49 and 340, respectively.

In these Wikis, like on Wikipedia, editors collaboratively create and curate content. These Wikis come with support for categories, the *The Lord of the Rings* Wiki, for instance, having over 900 categories and over 1000 subcategory relationships, the *Star Wars* Wiki having 11k and 14k of each, respectively. Similarly as on Wikipedia, these category networks do not represent clean taxonomies in the ontological sense, containing for instance meta categories such as `1980 films`, or relations such as `Death in Battle` being a subcategory of `Character`.

3.3 Design Rationale and Overview

3.3.1 Design Space and Choices

Input: The input to the taxonomy induction problem is a set of entities, such as locations, characters and events, each with a description in the form of associated text or tags and categories. Entities with textual descriptions are easily available in many forums incl. Wikipedia, wikis of fan communities or scholarly collaborations, and other online media. Tags and categories, including some form of category hierarchy, are available in various kinds of wikis – typically in very noisy form, though, with a fair amount of uninformative and misleading connections. When such sites merely provide tags for entities, we can harness subsumptions between tags (e.g., simple association rules) to derive a *folksonomy* (see, e.g., [Fang et al., 2016, Hotho et al., 2006, Jäschke et al., 2007]) and use this as an initial category system. When only text is available, we can use Hearst patterns and other text-based techniques [Cimiano et al., 2005, Hearst, 1992, Sanderson and Croft, 1999] to generate categories and construct a subsumption-based tree.

Output: Starting with a noisy category tree or graph for a given set of entities, from a domain of interest, the goal of TiFi is to construct a clean taxonomy that preserves the valid and appropriate classes and their instance-of and subclass-of relationships but removes all invalid or misleading categories and connections. Formally, the output of TiFi is a directed acyclic graph (DAG) $G = (V, E)$ with vertices V and edges E such that (i) non-leaf vertices are semantic classes relevant for the domain, (ii) leaf vertices

[1] www.wikia.com/fandom
[2] www.gamepedia.com

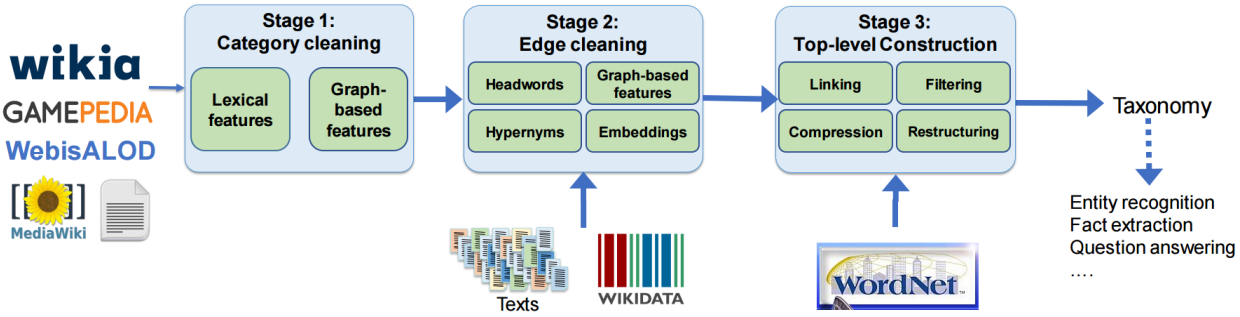


Figure 3.2: Architecture of TiFi.

are entities, (iii) edges between leaves and their parents denote which entities belong to which classes, (iv) edges among non-leaf vertices denote subclass-of relationships.

There is a wealth of prior literature on taxonomy induction methods, and the design space for going about fictitious and other non-standard domains has many options. Our design decisions are driven by three overarching considerations:

- We leverage *whatever input information is available*, even if it comes with a high degree of noise. That is, when an online community provides categories, we use them. When there are only tags or merely textual descriptions, we first build an initial category system using folksonomy construction methods and/or Hearst patterns.
- For the output taxonomy, we *prioritize precision over recall*. So our methods mostly focus on removing invalid vertices and edges. Moreover, to make classes for fictitious domains more interpretable and support cross-domain comparisons (e.g., for search), we aim to align the domain-specific classes with appropriate upper-level classes from a general-purpose ontology, using WordNet [Fellbaum and Miller, 1998]. For example, dragons in Lord of the Rings should be linked to the proper WordNet sense of dragons, which then tells us that this is a subclass of mythical creatures.
- It may seem tempting to cast the problem into an end-to-end machine-learning task. However, this would require sufficient training data in the form of pairs of input datasets and gold-standard output taxonomies. Such training data is not available, and would be hard and expensive to acquire. Instead, we break the overall task down into focused steps at the granularity of individual vertices and individual edges of category graphs. At this level, it is much easier to acquire labeled training data, by crowdsourcing (e.g., mturk). Moreover, we can more easily devise features that capture both local and global contexts, and we can harness external assets like dictionaries and embeddings.

3.3.2 TiFi Architecture

Based on the above considerations, we approach taxonomy induction in three steps, (1) category cleaning, (2) edge cleaning, (3) top-level construction. The architecture of TiFi is depicted in Fig. 3.2. Fig. 3.3 illustrates how TiFi constructs a taxonomy.

The first step, *category cleaning* (Section 3.4), aims to clean the original set of categories V by identifying categories that truly represent classes within the domain of interest, and by removing categories that represent, for instance, meta-categories used for community or Wikia coordination, or concern topics outside of the fictional domain, like movie or video game adaptions, award wins, and similar. Previous work has tackled this step via syntactic and lexical rules [Pasca, 2018, Ponzetto and Strube, 2007, Suchanek et al., 2007]. While such custom-tailored rules can achieve high accuracy, they have limitations w.r.t. applicability across domains. We thus opt for a supervised classification approach that combines rules from above with additional graph-based features. This way, taxonomy construction for a new domain only requires new training examples instead of new rules. Moreover, our experiments show that, to a reasonable extent, models can be reused across domains.

The second step, *edge cleaning* (Section 3.5), identifies the edges from the original category network E that truly represent subcategory relationships. Here, both rule-based [Gupta et al., 2016a, Ponzetto and Strube, 2007] and embedding-based approaches [Nguyen et al., 2017b] appear in the literature. Each approach has its strength, however, rules again have limitations wrt. applicability across domains, while embeddings may disregard useful syntactic features, and crucially rely on enough textual content for learning. We thus again opt for a supervised approach, allowing us to combine existing lexical and embedding-based approaches with various adapted semantic and novel graph-based features.

For the third step, *top-level construction* (Section 3.6), basic choices are to aim to construct the top levels of taxonomies from input category networks [Gupta et al., 2016a, Ponzetto and Strube, 2007], or to reuse existing abstract taxonomies such as WordNet [Suchanek et al., 2007]. As fan Wikis (and even Wikipedia) generally have a comparably small coverage of abstract classes, we here opt for the reuse of the existing WordNet top-level classes. This also comes with the additional advantage of establishing a shared vocabulary across domains, allowing to query, for instance, for *animal species appearing both in LoTR and GoT* (with answers such as dragons).

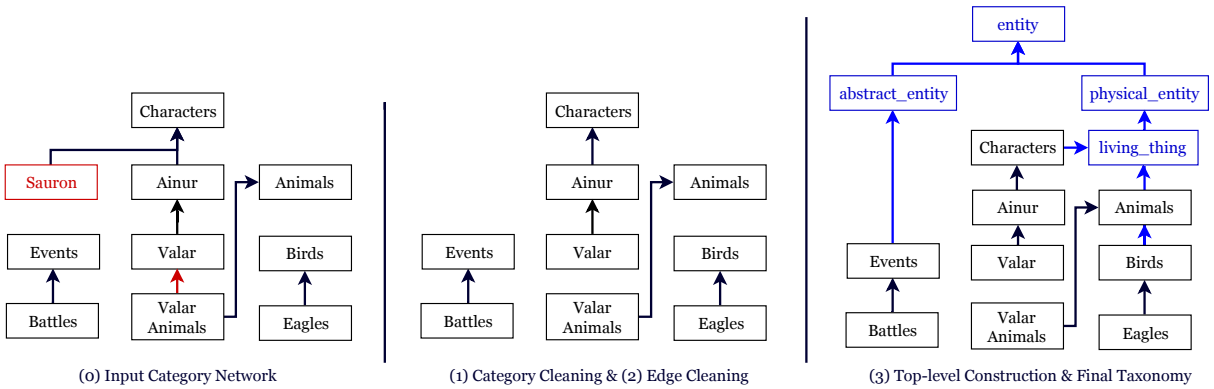


Figure 3.3: Example of three-stage taxonomy induction.

3.4 Category Cleaning

In the first step, we aim to select the categories from the input that actually represent classes in the domain of interest. There are several reasons why a category would not satisfy this criterion, including the following:

- *Meta-categories:* Wiki platforms typically introduce metacategories related to administration and technical setup, e.g., `Meta` or `Administration`.
- *Contextual categories:* Community Wikis usually contain also information about the production of the universes (e.g., inspirations or actors), about the reception (e.g., awards), and about remakes and adaptions, which do not related to the real content of the universes.
- *Instances:* Editors frequently create categories that are actually instances, e.g., Arda or Mordor in *The Lord of The Rings*).
- *Extensions:* Wikis sometimes also contains fan-made extensions of universes that are not universally agreed upon.

Previous works on Wikipedia remove either only meta-categories or instances by using crafted lexical rules [Pasca, 2018, Ponzetto and Strube, 2007, 2011]. As our setting has to deal with a wider range of noise, we instead choose the use of supervised classification. We use a logistic regression classifier with binary (0/1) lexical and integer graph-based features, as detailed next.

A. Lexical Features

- *Meta-categories*: True if a categories' name contains one of 22 manually selected strings, such as `wiki`, `template`, `user`, `portal`, `disambiguation`, `articles`, `file`, `pages`, `administration`, etc.

- *Plural categories*: True if the headword of a category is in plural form. We use shallow parsing to extract headwords, for instance, identifying the plural term `Servants` in `Servants of Morgoth`, a strong indicator for a class.
- *Capitalization*: True if a category starts with a capital letter. We introduced this feature as we observed that in fiction, lowercase categories frequently represent non-classes.

B. Graph-based Features

- *Instance count:* The number of direct instances of a category.
- *Supercategory/subcategory count:* The number of super/subcategories of a category, e.g., 0/2 for `Characters` in Fig. 3.3 (left). Categories with more instances, superclasses or subclasses have potentially more relevance.
- *Average depth:* Average upward path length from a category. Categories with short paths above are potentially more likely not relevant.
- *Connected subgraph size:* The maximal size of connected subgraphs which a given category belongs to. Each connected subgraph is extracted by using depth first search on each root of the input category network. Meta-categories are sometimes disconnected from the core classes of a universe.

While the first two are established features, all other features have been newly designed to especially meet the characteristics of fiction. As we show in Section 5.7, this varied feature set allows to identify in-domain classes with 83%-85% precision.

3.5 Edge Cleaning

Once the categories that represent classes in the domain of interest have been identified, the next task is to identify which subcategory relationships also represent subclass relationships. While most previous works rely on rules [de Melo and Weikum, 2010, Flati et al., 2014, Gupta et al., 2016c, Ponzetto and Strube, 2007], these are again too inflexible for the diversity of fictional universes. We thus tackle the task using supervised learning, relying on a combination of syntactic, semantic and graph-based features for a regression model.

A. Syntatic Features

Head Word Matching Head word matching is arguably the most popular feature for taxonomy induction. Categories sharing the same headword, for instance `Realms` and

`Dwarven Realms` are natural candidates for hypernym relationships.

We use a shallow parsing to extract, for a category c, its headword $head(c)$, its prefix $pre(c)$, and its suffix (postfix) $pos(c)$, that is, $c = pre(c) + head(c) + pos(c)$. Consider a subcategory pair (c_1, c_2):

1. If $head(c_1) = head(c_2)$, $head(c_1)+pos(c_1) = head(c_2)+pos(c_2)$ and $pre(c_2) \subseteq pre(c_1)$ then c_2 is a superclass of c_1.
2. If $head(c_1) = head(c_2)$, $pre(c_1)+head(c_1) = pre(c_2)+head(c_2)$ and $pos(c_2) \subseteq pos(c_1)$ then c_2 is a superclass of c_1.
3. If $head(c_1) \neq head(c_2)$ and $head(c_2) \subseteq pre(c_1)$ or $head(c_2) \subseteq pos(c_1)$ then there is no subclass relationship between c_1 and c_2.

Case (1) covers the example of `Realms` and `Dwarven Realms`, while case (2) allows to infer, for instance, that `Elves` is a superclass of `Elves of Gondolin`. Case (3) allows to infer that certain categories are not superclasses of each other, e.g., `Gondor` and `Lords of Gondor`. Each of subclass and no-subclass inference are implemented as binary 0/1 features.

Only Plural Parent True if for a subcategory pair (c_1, c_2), c_1 has no other parent categories, and c_2 is in plural form [Gupta et al., 2016c].

B. Semantic Features

WordNet Hypernym Matching WordNet is a carefully handcrafted lexical database that contains semantic relations between words and word senses (synsets), including hypo/hypernym relations. To leverage this resource, we map categories to WordNet synsets, using context-based similarity to identify the right word sense in the case of ambiguities. To compute the context vectors of categories, we extract their definitions, that is, the first sentence from the Wiki pages of the categories (if existing), and their parent and child class names. As context for WordNet synsets we use the definition (gloss) of each sense. We then compute cosine similarities over the resulting bags-of-words, and link each category with the position-adjusted most similar WordNet synset (see Alg. 1). Then, given categories c_1 and c_2 with linked WordNet synset s_1 and s_2, respectively, this feature is true if s_2 is a WordNet hypernym of s_1.

Wikidata Hypernym Matching Similarly to WordNet, Wikidata also contains relations between entities. For example, Wikidata knows that `Maiar` is an instance (P31) of

Algorithm 1: WordNet Synset Linking

```
Data: A category c
Result: WordNet synset s of c
c = pre + head + pos, l = null;
l = list of WordNet synset candidate for c;
if l = null then
    l = list of WordNet synset candidates for pre + head;
    if l = null then
        l = list of WordNet synset candidates for head;
if l = null then
    return null;
max = 0, s = null;
for all WordNet synset s_i in l do
    sim(s_i, c) = cosine(V_{s_i}, V_c) with V: context vector;
    sim(s_i, c = sim(s_i, c) + 1/(2R_{s_i}) where R: rank in WordNet;
    if sim(s_i, c) > max then
        max = sim(s_i, c);
        s = s_i;
return s;
```

`Middle-earth races` in the *The Lord of the Rings*. While Wikidata's coverage is generally lower than that of Wordnet, its content is sometimes complementary, as WordNet does not know certain concepts, e.g., `Maiar`.

Page Type Matching One interesting contribution of the WiBi system [Flati et al., 2014] was to use the first sentence of Wikipedia pages to extract hypernyms. First sentences frequently define concepts, e.g., *"The Haradrim, known in Westron as the Southrons and once as the "Swertings" by Hobbits, were a race of Men from Harad in the region of Middle-earth directly south of Gondor"*. For categories having matching articles in the Wikis, we rely on the first sentence from these. We use the Stanford Parser [Manning et al., 2014] on the definition of the category to get a dependency tree. By extracting `nsubj, compound` and `conj` dependencies, we get a list of hypernyms for the category. For example, for `Haradrim` we can extract the relation `nsubj(race-13, Haradrim-2)`, hence `race` is a hypernym of `Haradrim`. After getting hypernyms for a category, we link these hypernyms to classes in the taxonomies by using head word matching, and set this feature to true for any pair of categories linked this way.

WordNet Synset Description Type Matching Similar to page type matching, we also extract superclass candidates from the description of the WordNet synset. For instance, given the WordNet description for `Werewolves`: *"a monster able to change appearance from human to wolf and back again"*, we can identify `Monster` as superclass.

Distributional Similarity The distributional hypothesis states that similar words share similar contexts [Harris, 1954], and despite the subclass relation being asymmetric, symmetric similarity measures have been found to be useful for taxonomy construction [Shwartz et al., 2016]. In this work, we utilize two distributional similarity measures, a symmetric one based on the structure of WordNet, and an asymmetric one based on word embeddings. The symmetric Wu-Palmer score compares the depth of two synsets (the headwords of the categories) with the depth of their least common subsumer (*lcs*) [Wu and Palmer, 1994]. For synsets s_1 and s_2, it is computed as:

$$\mathit{Wu\text{-}Palmer}(s_1, s_2) = \frac{2 * \mathit{depth}(\mathit{lcs}(s_1, s_2)) + 1}{\mathit{depth}(s_1) + \mathit{depth}(s_2) + 1} \tag{3.1}$$

The HyperVec score [Nguyen et al., 2017b] not only shows the similarity between a category and its hypernym, but is also directional. Given categories c_1 and c_2, with stemmed head words h_1, h_2 respectively, the HyperVec score is computed as:

$$\mathit{HyperVec}(c_1, c_2) = \mathit{cosine}(E_{h_1}, E_{h_2}) * \frac{||E_{h_2}||}{||E_{h_1}||}, \tag{3.2}$$

where E_h is the embedding of word h. Specifically, we are using Word2Vec [Mikolov et al., 2013] to train a distributional representation over Wikia documents. The term $\mathit{cosine}(E_{h_1}, E_{h_2})$ represents the cosine similarity between two embeddings, $||E_h||$ the Euclidean norm of an embedding. While WordNet only captures similarity between general concepts, embedding-based measures can cover both conceptual and non-conceptual categories, as often needed in the fantasy domain (e.g. similarity between `Valar` and `Maiar`).

C. Graph-based Features

Common Children Support Absolute number of common children (categories and instances) of two given categories. Presumably, the more common children two categories have, the more related to each other they are.

Children Depth Ratio The ratio between the number of child categories of the parent of the edge, and its average depth in the taxonomy. This feature models the generality

of the parent candidate.

The features for edge cleaning combine existing state-of-the-art features (Head word matching, Page type matching, HyperVec) with adaptations specific to our domain (Wikidata hypernym matching, WordNet synset matching), and new graph-based features. Section 5.7 shows that this feature set allows to surpass the state-of-the-art in edge cleaning by 6-8% F1-score.

3.6 Top-level Construction

Category systems from Wiki sources often rather resemble forests than trees, i.e., do not reach towards very general classes, and miss useful generalizations such as `man-made structures` or `geographical features` for `fortresses` and `rivers`. While works geared towards Wikipedia typically conclude with having identified classes and subclasses [de Melo and Weikum, 2010, Flati et al., 2014, Gupta et al., 2016c, Ponzetto and Strube, 2007, 2011], we aim to include generalizations and abstract classes consistently across universes. For this purpose, TiFi employs as third step the integration of selected abstract WordNet classes. The integration proceeds in three steps:

1. Given the taxonomy constructed so far, nodes are linked to WordNet synsets using Algorithm 1. Where the linking is successful, WordNet hypernyms are then added as superclasses. For example, the category `Birds` is linked to the WordNet synset `bird%1:05:00::`, whose superclasses are `wn_vertebrate` → `wn_chordate` → `wn_animal` → `wn_organism` → `wn_living_thing` → `wn_whole` → `wn_object` → `wn_physical_entity` → `wn_entity`.

2. The added classes are then compressed by removing those that have only a single parent and a single child, for instance, `abstract_entity` and `physical_entity` in Fig. 3.3 (right) would be removed, if they really had only one child.

3. We correct a few WordNet links that are not suited for the fictional domain, and use a self-built dictionary to remove 125 top-level WordNet synsets that are too abstract to add value, for instance, `whole`, `sphere` and `imagination`.

Note that the present step can add subclass relationships between existing classes. In Fig. 3.3, after edge filtering, there is no relation between `Birds` and `Animals`, while after linking to WordNet, the subclass relation between `Birds` and `Animals` is added, making the resulting taxonomy more dense and useful.

3.7 Evaluation

In this section we evaluate the performance of the individual steps of the TiFi approach, and the ability of the end-to-end system to build high-quality taxonomies.

Universes We use 6 universes that cover fantasy (LoTR, GoT), science fiction (Star Wars), animated sitcom (Simpsons), video games (World of Warcraft) and mythology (Greek Mythology). For each of these, we extract their category networks from dump files of Wikia or Gamepedia. The sizes of the respective category networks, the input to TiFi, are shown in Table 3.1.

3.7.1 Step 1: Category Cleaning

Evaluation data for the first step was created using crowdsourcing, which was used to label all categories in LoTR, GoT, and random 50 from each of the other universes. Specifically, workers were asked to decide whether a given category had instances *within* the fictional domain of interest. We collected three opinions per category, and chose majority labels. Worker agreement was between 85% and 91%.

As baselines we employ a rule-based approach by Ponzetto & Strube [Ponzetto and Strube, 2011], to the best of our knowledge the best performing method for general category cleaning, and recent work by Marius Pasca [Pasca, 2018] that targets the aspect of separating classes from instances. Furthermore, we combine both methods into a joint filter. The results of training and testing on LoTR/GoT, respectively, each under 10-fold crossvalidation, are shown in Table 3.2. TiFi achieves both superior precision (+40%) and F1-score (+22%/+23%), while observing a smaller drop in recall (-18%/-15%). On both fully annotated universes the improvement of TiFi over the combined baseline in terms of F1-score is statistically significant (p-value 2.2^{-16} and 1.9^{-13}, respectively). The considerable difference in precision is explained largely by the limited coverage of the

Universe	# Categories	# Edges
Lord of the Rings (LoTR)	973	1118
Game of Thrones (GoT)	672	1027
Star Wars	11012	14092
Simpsons	2275	4027
World of Warcraft	8249	11403
Greek Mythology	601	411

Table 3.1: Input categories from Wikia/Gamepedia.

Method	Universe	Precision	Recall	F1-score
Pasca [2018]	LoTR	0.33	0.75	0.46
	GoT	0.57	0.85	0.68
Ponzetto & Strube [2011]	LoTR	0.44	**1.0**	0.61
	GoT	0.45	**1.0**	0.62
Pasca + Ponzetto & Strube	LoTR	0.41	0.75	0.53
	GoT	0.64	0.85	0.73
TiFi	LoTR	**0.84**	0.82	**0.83**
	GoT	**0.85**	0.85	**0.85**

Table 3.2: Step 1 - In-domain category cleaning.

Train	Test	Precision	Recall	F1-score
LoTR	GoT	0.81	0.85	0.83
GoT	LoTR	0.64	0.88	0.74
LoTR	Star Wars	0.63	0.94	0.75
LoTR	Simpsons	0.91	0.63	0.74
LoTR	World of Warcraft	0.95	0.63	0.75
LoTR	Greek Mythology	0.86	0.6	0.71

Table 3.3: Step 1 - Cross-domain category cleaning.

rule-based baseline. Typical errors TiFi still makes are cases where categories have the potential to be relevant, yet appear to have no instances, e.g., `song` in LOTR. Also, it occasionally misses out on conceptual categories which do not have plural forms, e.g., `Food`.

A characteristic of fiction is variety. As our approach requires labeled training data, a question is to which extent labeled data from one domain can be used for cleaning categories of another domain. We thus next evaluate the performance when applying models trained on LoTR on the other 5 universes, and the model trained on GoT on LoTR. The results are shown in Table 3.3, where for universes other than LoTR and GoT, having annotated only 50 samples. As one can see, F1-scores drop by only 9%/2% compared with same-domain training, and the F1-score is above 70% even for quite different domains.

To explore the contribution of each feature, we performed an ablation test using recursive feature elimination. The most important feature group were lexical features (30%/10% F1-score drop if removed in LoTR/GoT), with plural form checking being the single most important feature. In contrast, removing the graph-based features lead only to a 10%/0% drop, respectively.

Method	Universe	Precision	Recall	F1-score
HyperVec [Nguyen et al., 2017b]	LoTR	0.82	0.8	0.81
	GoT	**0.83**	0.81	0.82
HEAD [Gupta et al., 2016c]	LoTR	**0.85**	0.83	0.84
	GoT	0.81	0.78	0.79
TiFi	LoTR	0.83	**0.98**	**0.90**
	GoT	**0.83**	**0.91**	**0.87**

Table 3.4: Step 2 - In-domain edge cleaning.

Train	Test	Precision	Recall	F1-score	MAP
LoTR	GoT	0.81	0.79	0.80	0.92
GoT	LoTR	0.89	0.87	0.88	0.89
GoT	Star Wars	0.92	0.92	0.92	0.91
GoT	Simpsons	0.86	0.86	0.86	0.92
GoT	Word of Warcraft	0.72	0.71	0.72	0.76
GoT	Greek Mythology	0.92	0.92	0.92	0.92

Table 3.5: Step 2 - Cross-domain edge cleaning.

		Proper-name edges			Concept edges		
Method	Universe	Precision	Recall	F1-score	Precision	Recall	F1-score
HyperVec	LoTR	0.88	0.59	0.71	0.80	0.88	0.84
	GoT	**1.0**	0.16	0.27	0.83	0.9	0.87
HEAD	LoTR	0.91	0.74	0.81	0.83	0.87	0.85
	GoT	0.72	0.68	0.70	0.82	0.8	0.81
TiFi	LoTR	**0.92**	**0.79**	**0.85**	**0.88**	**0.89**	**0.88**
	GoT	0.96	**0.68**	**0.8**	**0.90**	**0.91**	**0.91**

Table 3.6: Step 2 - Edge cleaning: Proper-name vs. concept edges.

3.7.2 Step 2: Edge Cleaning

We used crowdsourcing to label all edges that remained after cleaning noisy categories from LoTR, GoT, and random 100 edges in each of the other universes. For example, we asked Turker whether in LOTR, `Uruk-hai` are `Orc Man Hybrids`. Inter-annotator agreement was between 90% and 94%.

We compare with two state-of-the-art systems: (1) HEAD [Gupta et al., 2016c], the most recent system for Wikipedia category relationship cleaning, and (2) HyperVec [Nguyen et al., 2017b], a recent embedding-based hypernym relationship learning system. The results for in-domain evaluation using 10-fold crossvalidation are shown in Table 3.4. As one can see, TiFi achieves a comparable precision (-2%/+2%), and a superior recall (+15%/+13%), resulting in a gain in F1-score of 6%/8%. Again, the F1-score improvement of TiFi over HyperVec and HEAD on the two fully annotated universes is statistically significant (p-values 7.1^{-9}, 0.01, 5.8^{-5} and 6.5^{-5}, respectively).

To explore the scalability of TiFi, we again perform cross-domain experiments using 100 labeled edges per universe. The results are shown in Table 3.5. In all universes but *World of Warcraft*, TiFi achieves more than 80% F1-score, and the performance is further highlighted by mean average precision (MAP) scores above 89%, meaning TiFi can effectively separate correct from incorrect edges.

As mentioned earlier, taxonomy induction on real-world domain can leverage a lot of semantic knowledge like WordNet synsets, while fiction frequently contains non-standard categories such as `Valar` and `Tatyar`. We further evaluate the performance of TiFi by distinguishing two types of edges:

- *Concept edges:* Both parent and child exist in WordNet.
- *Proper-name edges:* At least one of parent and child does not exist in WordNet.

In *The Lord of the Rings*, there are 145 proper-name edges and 407 concept edges, while in *Game of Thrones*, there are 61 and 329 of each, respectively. Table 3.6 reports the performance of TiFi, comparing to HEAD and HyperVec on both types of edges. As one can see, for proper-name edges, TiFi achieves a very high precision of 92%/96%, outperforms HEAD by 4%/10% and HyperVec by 14%/53% in F1-score, respectively.

We again performed an ablation test in order to understand feature contribution. We found that all three groups of features have importance, observing a 1-4% drop in F1-score when removing any of them. The individually most important features were *Only Plural Parent, Headword Matching, Common Children Support* and *Page Type Matching*.

Universe	#New Types	#New Edges	Precision
LoTR	43	171	0.84
GoT	39	179	0.84
Starwars	373	3387	0.84
Simpsons	115	439	0.92
World of Warcraft	257	2248	0.84
Greek Mythology	22	76	0.84

Table 3.7: Step 3 - WordNet integration.

3.7.3 Step 3: Top-level Construction

The key step in top-level construction is the linking of categories to WordNet synsets (i.e. category disambiguation), hence we only evaluate this step. For this purpose, in each universe, we randomly selected 50 such links and evaluated their correctness, finding precisions between 84% and 92% (see Table 3.7). Overall, this step is able to link 30-72% of top-level classes from Step 2, and adds between 22 to 373 WordNet classes and 76 to 3387 subclass relationships to our universes.

3.7.4 Final Taxonomies

Table 3.8 summarizes the taxonomies constructed for our 6 universes, with the bottom 4 universes built using the models for GoT. Reported precisions refer to the weighted average of the precision of subclass edges from Step 2, and the precision of WordNet linking. Figure 3.4 shows the resulting taxonomy for Greek Mythology, rendered using the R layout *fruchterman.reingold*. All taxonomies will be made available both as CSV and graphically.

Universe	# Types	# Edges	Precision
LoTR	353	648	0.88
Game of Thrones	292	497	0.83
Star Wars	7352	12282	0.90
Simpsons	1029	2171	0.88
World of Warcraft	4063	7882	0.76
Greek Mythology	139	313	0.91

Table 3.8: Taxonomies produced by TiFi.

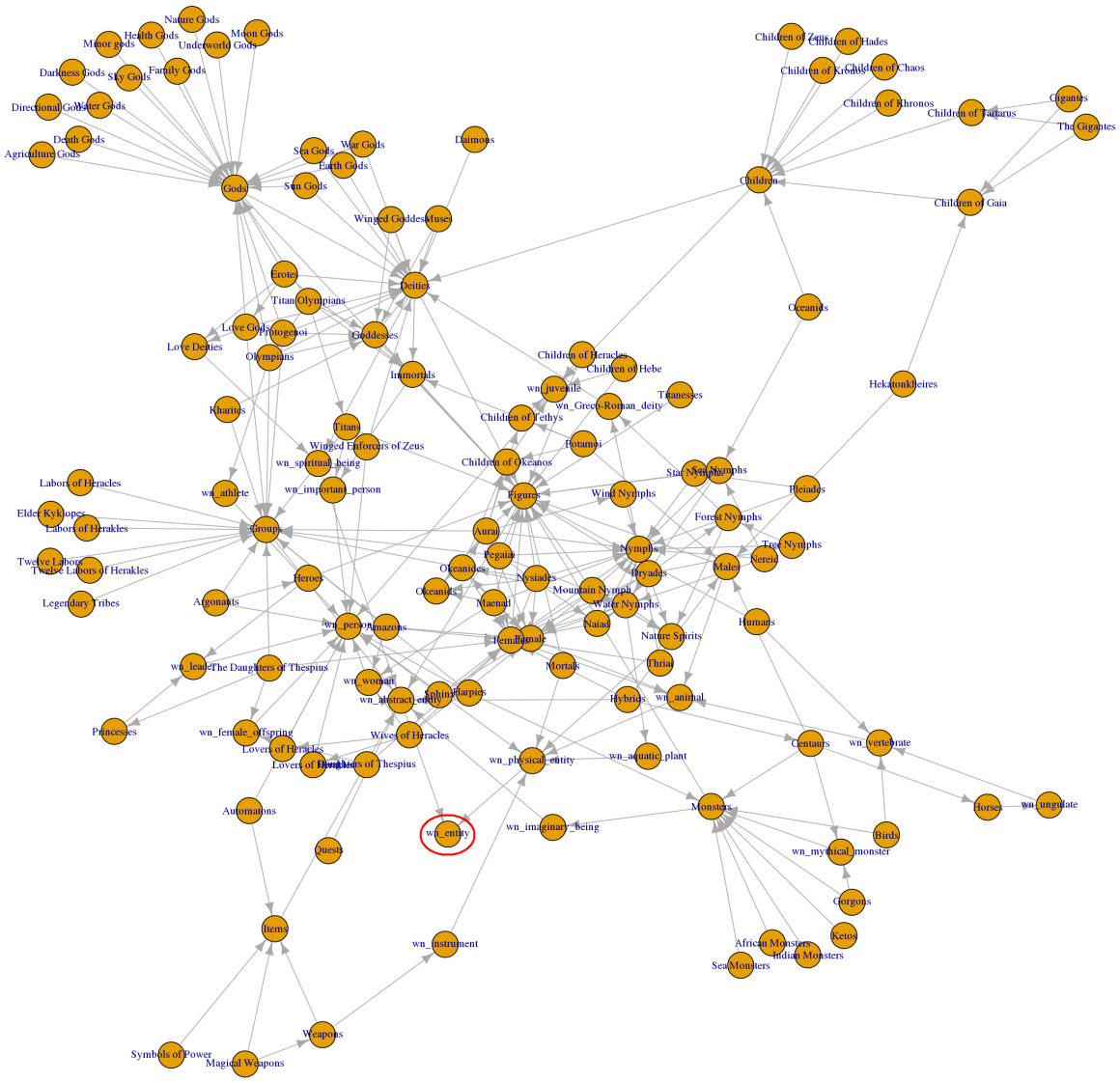


Figure 3.4: Final TiFi taxonomy for Greek Mythology.

3.7.5 Wikipedia as Input

While our method is targeted towards fiction, it is also interesting to know how well it does in the traditional Wikipedia setting. To this end, we extracted a specific slice of Wikipedia, namely all categories that are subcategories of `Desserts`, resulting in 198 categories connected by 246 subcategory relations, which we fully labeled.

Using 10-fold crossvalidation, in the first step, category cleaning, our method achieves 99% precision and 99% recall, which puts it on par with Ponzetto & Strube [Ponzetto and Strube, 2011], which achieves 99% precision and 100% recall. The reason for the excellent performance of both systems is that noise in Wikipedia categories concerns fairly uniformly meta-categories, which can be well filtered by enumerating them. In the second step, edge cleaning, TiFi also achieves comparable results, with a slightly lower precision (83% vs. 87%) and a slightly higher recall (92% vs. 89%), resulting in 87% F1-score for TiFi vs. 88% for HEAD.

3.7.6 WebIsALOD as Input

WebIsALOD [Hertling and Paulheim, 2017] is a large collection of hypernymy relations extracted from the general web (Common Crawl). Relying largely on pattern-based extraction, the data from WebisALOD is very noisy, especially beyond the top-confidence ranks. Being text-based, several features based on category systems become unavailable, making this source an ideal stress test for the TiFi approach.

Data: To get data from WebisALOD, we selected the top 100 most popular entities from two universes, *The Lord of the Rings* and *Simpsons*, 100 per each, based on the frequency of their mentions in text. We then queried the hypernyms of these entities and took the top 3 hypernyms based on ranking of confidences cores (minimum confidence 0.2). We iterated this procedure once with the newly gained hypernyms. In the end, with *The Lord of the Rings*, we get 324 classes and 312 hypernym relations, meanwhile, with *Simpsons*, these numbers are 271 classes and 228 hypernym relations. We fully manual label these datasets by checking whether classes are noisy and hypernym relations are wrong. From the labeled data, only 217 classes (67%) and 167 classes (62%) should be kept in *The Lord of the Rings* and *Simpsons*, respectively. In the case of hypernym relations, only 42% and 47% of them are considered to be correct relations in *The Lord of the Rings* and *Simpsons*, respectively. These statistics confirm that the data from WebisALOD is very noisy.

Method	Universe	Precision	Recall	F1-score
Pasca [2018]	LoTR	0.67	**1.0**	0.80
	Simpsons	0.62	**1.0**	0.76
Ponzetto & Strube [2011]	LoTR	0.67	**1.0**	0.80
	Simpsons	0.62	**1.0**	0.76
TiFi	LoTR	**0.89**	0.94	**0.91**
	Simpsons	**0.95**	0.97	**0.96**

Table 3.9: WebIsALOD input - step 1 - In-domain cat. cleaning.

Method	Universe	Precision	Recall	F1-score
HEAD [Gupta et al., 2016c]	LoTR	0.27	0.05	0.09
	Simpsons	0.31	0.09	0.14
TiFi	LoTR	**0.79**	**0.55**	**0.62**
	Simpsons	**0.61**	**0.32**	**0.42**

Table 3.10: WebIsALOD - step 2 - In-domain edge cleaning.

Results: In Step 1, Ponzetto & Strube [Ponzetto and Strube, 2011] use lexical rules to remove meta-categories, while Pasca [Pasca, 2018] uses heuristics which are based on information extracted from Wikipedia pages to detect entities that are classes. To enable comparison with Pasca's work, we used exact lexical matches to link classes from WebIsALOD to Wikipedia pages titles, then used Wikipedia pages as inputs. In fact, classes from WebisALOD are hardly meta-categories and the additional data from Wikipedia is also quite noisy. Table 3.9 shows that TiFi still performs very well in category cleaning, and significantly outperforms the baselines by 10%/20% F1-score.

In Step 2, HEAD uses heuristics to clean hypernym relations between classes, mostly based on lexical and information from class pages (e.g. Wikipedia pages). Although TiFi also uses the information from class pages, its supervised model uses also a set of other features and is thus more versatile. Table 3.10 reports the results of TiFi, comparing with HEAD in edge cleaning, with TiFi outperforming HEAD by 28%-53% F1-score.

Both steps were also evaluated in the cross-domain settings, with similar results (90%/91% F1-score in step 1, 53%/55% F1-score in step 2).

3.8 Use Case: Entity Search

To highlight the usefulness of our taxonomies, we provide an extrinsic evaluation based on the use case of entity search. Entity search is a standard problem in information retrieval, where often, textual queries shall return lists of matching entities. In the following, we focus on the retrieval of correct entities only, and disregard the ranking

aspect.

Setup We consider three universes, *The Lord of the Rings, Simpsons* and *Greek Mythology*, and manually generated 90 text queries belonging to the following categories (10 of each per universe):

1. Single type: Entities belonging to a class, e.g., *Orcs in the Lords of the Rings*;
2. Type intersection: Entities belonging to two classes, e.g., *Humans that are agents of Saruman*;
3. Type difference: Entities that belong to one class but not another, e.g., *Spiders that are not servants of Sauron.*

We utilize the following resources:

- Unstructured resources: (1) Google Web Search and (2) the Wikia-internal text search function;
- Structured resources: (3) the Wikia category networks and (4) the taxonomies as built by TiFi.

Evaluation For the unstructured resources, we manually checked the titles of the top 10 returned pages for correctness. For the structured resources, we matched the classes in the query against all classes in the taxonomy that contained those class names as substrings. We then computed, in a breadth-first manner, all subclasses and all instances of these classes, truncating the latter to maximal 10 answers, and manually verified whether returned instances were correct or not.

Results Table 3.11 reports for each resource the average number of results and their precision. We find that Google performs worst mainly because its diversification is limited (returns distinct answers often only far down in the ranking), and because it cannot well process conjunction and negation. Wikia performs better in terms of answer size, as by design it contains each entity only once. Still, it struggles with logical connectors. The Wikia categories produce more results than TiFi (9 vs. 6 on average), though due noise, they yield a substantially lower precision (-24%). This corresponds to the core of the TiFi approach, which in step 1 and 2 is cleaning, i.e., leads to a lower recall while increasing precision.

Table 12 lists three sample queries along with their output. Crossed-out entities are incorrect answers. As one can see, text search mostly fails in answering the queries that

	Text		Structured Sources	
Query	**Google**	**Wikia**	**Wikia-categories**	**TiFi**
t	2 (52%)	7 (65%)	10 (62%)	8 (87%)
$t_1 \cap t_2$	1 (23%)	2 (11%)	8 (40%)	3 (70%)
$t_1 \setminus t_2$	1 (20%)	4 (36%)	8 (63%)	6 (79%)
Average	1 (32%)	4 (37%)	9 (55%)	6 (79%)

Table 3.11: Avg. #Answers and precision of entity search.

	Text		Structured Sources	
Query	**Google**	**Wikia**	**Wikia-categories**	**TiFi**
Dragons in LOTR	Glaurung, Túrin, Turambar, Eärendil, Smaug, Ancalagon	Dragons, ~~Summoned Dragon~~, Spark-dragons	~~Urgost~~,Long-worms,Gostir,~~Drogoth the Dragon Lord~~,~~Cave Drake~~, ~~War of the Dwarves and Dragons~~, ~~Dragon spell~~,Stone Dragons, Fire-drake of Gondolin,Spark-dragons, Were-worms, ~~Summoned Dragon~~, Fire-drakes, Glaurung,Ancalagon,Dragons,Cold-drakes, Sea-serpents, ~~User blog:Alex Liooe/Kaltdrache the Dragon~~, Smaug, ~~Dragon (Games, Workshop)~~, ~~Drake~~, Scatha, ~~The Fall of Erebor~~	Long-worms, ~~War of the Dwarves and Dragons~~, ~~Dragon spell~~,Stone Dragons, Fire-drake of Gondolin, Spark-dragons, Were-worms, Fire-drakes, Glaurung, Ancalagon, Dragons, Cold-drakes, Sea-serpents, Smaug, Scatha ,~~The Fall of Erebor~~, Gostir
Which Black Numenoreans are servants of Morgoth	-	Black Númenórean	Men of Carn Dûm,Corsairs of Umbar,Witch-king of Angmar, ~~Thrall Master~~,Mouth of Sauron,Black Númenórean,Fuinur	Men of Carn Dûm,Corsairs of Umbar,Witch-king of Angmar, Mouth of Sauron, Black Númenórean, Fuinur
Which spiders are not agents of Saruman?	-	-	Shelob, ~~Spider Queen and Swarm~~,~~Sacnathra~~, ~~Spiderling~~, Great Spiders, ~~Wicked, Wild, and Wrath~~	Shelob, Great Spiders

Table 12. Example queries and results for the entity search evaluation.

use boolean connectives, while the original Wikia categories are competitive in terms of the number of answers, but produce many more wrong answers.

3.9 Summary

In this chapter we have introduced TiFi, a system for taxonomy induction for fictional domains. TiFi uses a three-step architecture with category cleaning, edge cleaning, and top-level construction, thus building holistic domain specific taxonomies that are consistently of higher quality than what the Wikipedia-oriented state-of-the-art could produce.

Unlike most previous work, our approach is not based on static rules, but uses supervised learning. This comes with the advantage of allowing to rank classes and edges, for instance, in order to distinguish between core elements, less or marginally relevant ones, and totally irrelevant ones. In turn it also necessitates the generation of training data, yet we have shown that training data can be reasonably reused across domains.

Mirroring earlier experiences of YAGO [Suchanek et al., 2007], it also turns out that a crucial step in building useful taxonomies is the incorporation of abstract classes. For TiFi we relied on the established WordNet hierarchy, nevertheless finding the need to adapt a few links, and to remove certain too abstract concepts.

So far we only applied our system to fictional domains and one slice of Wikipedia. In the future, we would like to explore the construction of more domain-specific but real-world taxonomies, such as gardening, Maya culture or Formula 1 racing.

Chapter 4

ENTYFI: Entity Typing in Fictional Texts

4.1 Introductions

Motivation and Problem Entity typing, also known as entity type classification, is an important task in natural language processing, the goal being to assign types to mentions of entities in textual contexts (e.g., `person` or `event`, or `singer, bassist, concert` etc. for finer granularity). Type information is valuable for many other NLP tasks, such as coreference resolution, relation extraction, semantic search and question answering [Carlson et al., 2010b, Lee et al., 2006, Recasens et al., 2013]. While standard NLP suites such as Stanford CoreNLP distinguish a few coarse-grained entity types such as `person, organization`, and `location`, fine-grained entity typing has become a major effort in recent years, with some systems classifying mentions into hundreds to thousands of Wikipedia-based types [Choi et al., 2018, Corro et al., 2015, Lee et al., 2006, Ling and Weld, 2012].

Nonetheless, the world contains a plethora of non-standard long-tail domains, where these methods do not suffice. A particular important case is the professional world, where companies internally use specific job roles, product and supply item categories, project types, collaborator and customer types, etc. An enterprise-level type system cannot be derived from Wikipedia, and established entity typing methods are not geared for such non-standard domains.

Another case in point are fictional universes. Human creativity has led to the creation of fictional universes such as the Marvel Universe, Middle Earth, the Simpsons or the Mahabharata. These universes can be highly sophisticated, containing entities, locations, social structures, and sometimes even languages that are completely different from the real world. In this chapter, we focus on typing entity mentions in fictional

texts, like in the following example from Lord of the Rings:

"After Melkor's defeat in the First Age, Sauron became the second Dark Lord and strove to conquer Arda by creating the Rings"

MELKOR: `Ainur, Villain`	FIRST AGE: `Eras, Time`
SAURON: `Maiar, Villain`	DARK LORD: `Ainur, Title`
RINGS: `Jewelry, Magic Things`	ARDA: `Location`

State-of-the-art methods for entity typing on news and other real-world texts mostly rely on extensive supervised training, often using Wikipedia markup. Such techniques are not suited for typing mentions in fictional universes, where Wikipedia does not have sufficient coverage. Also, existing works typically produce predictions for single mentions, so that different occurrences of the same mention may be annotated with contradictory types, e.g., one occurrence of Gondor typed as `people` and another typed as `country`.

Use cases for entity typing include search and question answering by fans, and also text analytics for cultural or historic studies (incl. modern sub-culture such as mangas and other comics). For example, a Harry Potter fan may want to query for Gryffindor graduates with muggle parents. An analyst may want to discover patterns of character interactions in fantasy literature, or compare different mythologies. With fiction books and movies being a huge market, supporting search and analytics has monetary value.

Approach and Contributions We propose an archetypical method for mention typing in long-tail domains, called **ENTYFI** (fined-grained ENtity TYping on FIctional texts). To address the lack of reference types, we leverage the content of fan-created community Wikis on Wikia.com, from which we extract 205 sanitized reference type systems. Given a specific input text, we then identify the most related type systems from this reference set, and combine supervised typing with unsupervised pattern extraction and knowledge base (KB) lookups, in order to identify the most relevant types for a given mention. To consolidate the type predictions for individual mention occurrences, in the final step, we pass candidate types through an integer linear programming (ILP)-based consolidation stage, which filters out contradictory and overly generic or specific type predictions. Extensive experiments on novel, previously unseen fictional texts highlight the accuracy of ENTYFI. We also apply ENTYFI to historic and satirical texts, showing that our methodology outperforms state-of-the-art methods for real-world types also on these unconventional texts.

Our contributions are fourfold:

1. We study an archetypical problem of entity typing in non-standard domains with

long-tail types.

2. We present a 5-step method for entity typing in fiction, ENTYFI, consisting of type system construction (Sect. 4), reference universe ranking (Sect. 5), mention detection (Sect. 6), mention typing (Sect. 7), and type consolidation (Sect. 8).

3. For the core step – mention typing – we devise three complementary components: supervised classification, textual patterns and KB lookups.

4. Comprehensive experiments show the superior quality of ENTYFI over prior methods for fine-grained typing.

4.2 Related Work

Unsupervised Typing Mention typing is a task where entity mentions shall be assigned one or several relevant types. Earliest approaches to mention typing used unsupervised Hearst patterns [Hearst, 1992], which allow, for instance, to assign the type `Hobbit` to FRODO given the phrase *"Hobbit, such as Frodo."* Hearst patterns can achieve remarkable precision, and are part of many more advanced typing methods [Seitner et al., 2016].

(Semi-) Supervised Typing Named-entity recognition (NER) systems typically use a combination of rule-based and supervised extractions, and often distinguish a few basic types such as `person, location` and `organization` [Collobert et al., 2011, Finkel et al., 2005, Lample et al., 2016, Sang and De Meulder, 2003]. More recently, finer-grained entity detection and typing has received attention [Choi et al., 2018, Corro et al., 2015, Ling and Weld, 2012, Shimaoka et al., 2017]. These methods use much larger sets of targets, Ling and Weld for instance 112 types [Ling and Weld, 2012]. Similar feature based works are [Ren et al., 2016, Yogatama et al., 2015, Yosef et al., 2012]. FINET [Corro et al., 2015] uses the entire WordNet hierarchy with more than 16k types as targets, and builds a context-aware model which extracts information of types from the context of the mention (e.g. pattern-based, mention-based and verb-based extractors). After collecting type candidates for mentions, FINET uses word sense disambiguation technique to filter the results. Also extracting type candidates for the mentions, [Nakashole et al., 2013] and [Xu et al., 2018], on the other hand, use an ILP model to remove noisy types in the final results. While [Nakashole et al., 2013] extracts type candidates based on patterns, [Xu et al., 2018] uses a deep neural network model to classify a given mention.

Neural Methods With the emergence of deep learning, a set of neural methods for entity typing have been developed [Choi et al., 2018, Dong et al., 2015, Shimaoka et al., 2017, Xu et al., 2018]. The first attempt on using neural networks is by Dong et al. [Dong et al., 2015]. They define a set of 22 types and use a two-part neural classifier based on representations of entity mentions and their contexts. However, this model only focuses on single-label classification. [Shimaoka et al., 2017] develops several neural network models for fine-grained entity typing, including LSTM models with an attention mechanism. Recent works integrate neural models with hierarchy-aware loss functions [Xu and Barbosa, 2018], or utilize various kinds of information from knowledge bases [Jin et al., 2018]. Recently, Choi et al. [Choi et al., 2018] developed a method to predict so-called open types, which are collected using distant supervision from Wikipedia. The model is trained using a multitask objective combining head-word supervision with prior supervision from entity linking to Wikipedia, and contains more than 2500 types in its evaluation dataset. While most of existing works focus on typing a single entity mention, based on its surrounding context (e.g. usually in one sentence) and using a single approach, ENTYFI aims to predict types for entity mentions in long texts (e.g. FRODO in the whole book *The Lord of the Rings*). By proposing a hybrid approach which combines supervised and unsupervised-based approaches, ENTYFI is able to leverage both local contexts (e.g. the sentence from which the entity mention appears) to predict type candidates and global contexts (e.g. the whole book and the entity mention can appear more than once) to clean the prediction.

Domain-specific methods Most existing techniques focus on general-world domains, often using Wikipedia and news corpora for training and/or evaluation. One notable exception is the medical domain, which has a strong independent NLP community. Works in this space typically use supervised methods on manually annotated corpora [Dong et al., 2016, Liu et al., 2017, Wu et al., 2015]. Our method, ENTYFI, is the first attempt to entity typing for fictional texts.

Computational Linguistics and Fiction Analysis and interpretation of fiction are important topics for linguists and social scientists, and have recently been greatly helped by NLP tools that automate basic tasks, e.g., entity and topic detection, or sentiment classification. Automated techniques are for instance used to compare books with movie adaptations (via subtitle text alignment) [Tapaswi et al., 2015], to model and predict evolving relationships [Chambers and Jurafsky, 2009, Chaturvedi et al., 2017, Iyyer et al., 2016], or to measure gender bias and discrimination [Agarwal et al., 2015].

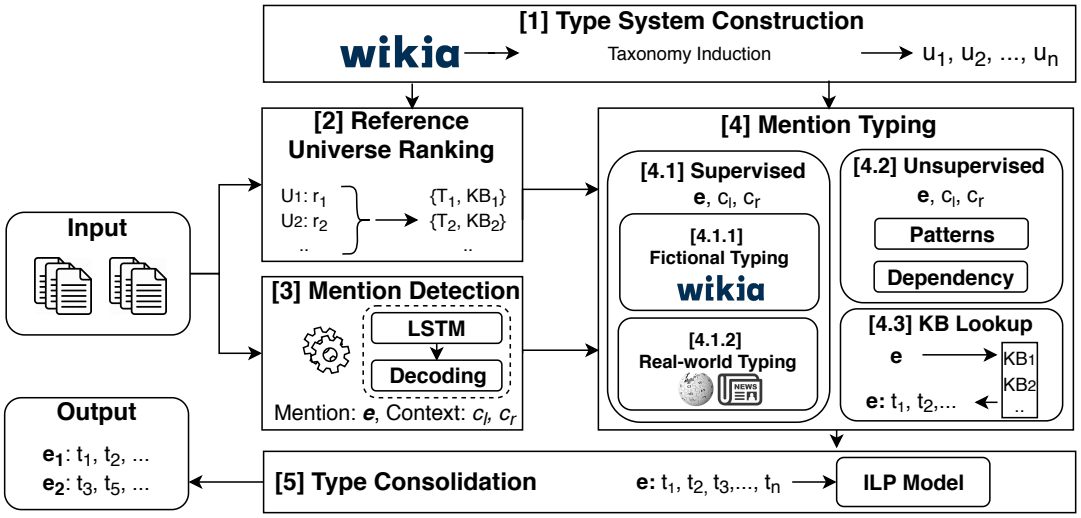


Figure 4.1: Overview of the architecture of ENTYFI.

4.3 Design Space and Approach

Entity typing would be best approached via manually curated training samples, but this does not scale to large domains. As a compromise, Wikipedia categories are frequently used as target classes, and training data is automatically distilled from Wikipedia links. For fiction, however, Wikipedia has too low coverage of entities and relevant types.

To achieve high recall, ENTYFI opts to distill target types for supervised classification from a large fiction community portal, Wikia. In addition, we consider further types expressed via Hearst patterns and dependency patterns, and search for possible type reuse in existing fictional domains. To ensure precision, for the supervised part, we only use types from universes most similar to the given input. Also, we hierarchically organize types, and clean candidate types in a precision-oriented consolidation stage. An overview of the ENTYFI architecture is shown in Figure 4.1.

In the first step, **type system construction**, all universes from Wikia which have over 1000 content pages with available dump file are downloaded and consolidated for use as reference universes. The type systems extracted from these universes are then induced for use as reference type systems.

In the second step, **reference universe ranking**, reference universes are ranked by their similarity to the input text, and the type systems of the most similar universes are used for supervised typing. As our experiments show, our reference type systems capture a great variety of fictional themes.

In the third step, **mention detection**, we identify text spans that are entity mentions. Inspired by [He et al., 2017], we develop a framework which uses highway connections

between Bi-LSTM layers to recognize entity mentions and decode the output with constraints of NER tasks, which does not add more complexity to the training process.

In the fourth step, **mention typing**, we run four modules in parallel.

a) Supervised fiction typing: We predict types from the reference type systems, along with 7 abstract types (`living_thing`, `location`, `object`, `organization`, `time`, `event` and `substance`), which are always predicted.

b) Supervised real-world typing: As fictional texts frequently overlap with reality, we utilize the model from [Choi et al., 2018] for predicting fine-grained real-world types.

c) Unsupervised typing: In this module, we use pattern-based and dependency-based method to extract types directly from the input text.

d) KB lookup: Given an entity mention, we attempt lookups in the reference universes based on surface form matches.

In the final step, **type consolidation**, type candidates for each mention are consolidated along taxonomical and statistical constraints. For example, Arda in *The Lord of the Rings* may have both `person` and `location` as candidates, which are unlikely to be both true. Also, mentions may occur several times in input texts, with conflicting type candidates. As sequence models like CRF, RNN or even LSTM are not suited for such scenarios, we develop an explicit ILP-based resolution model on top of individual mentions.

4.4 Type System Construction

While some parts of fiction are close to the real world (e.g., Big Bang theory), fantasy, science fiction and mythology have gone much beyond reality, be it in Middle-Earth, Star Wars, or Greek Mythology.

Wikia Wikia is the largest web platform for fandom, that is, organized fan communities on fictional universes. Wikia essentially provides a farm of Wikis, hosting as of July 2018 over 365,000 individual Wikis, each in its organization similar to Wikipedia. Wikia is a very popular website, as evidenced by its Alexa rank 49 worldwide (and 19 in the US).

Wikia covers a wide range of universes in fiction and fantasy domains, spanning from old folks and myths like *Greek Mythology, Egyptian Mythology* or *One Thousand and One Nights*, to modern stories like *The Lord of the Rings* and *Harry Potter*. It also hosts

Universe	#Pages	Rank
marvel.wikia.com - Comics, films	213,804	6
starwars.wikia.com - Movies	145,816	10
narutofanon.wikia.com - Mangas, TV series	36,521	51
simpsons.wikia.com - TV Series	19,996	102
harrypotter.wikia.com - Books, movies	15,742	147
lotr.wikia.com - Books, movies	6,386	402
gameofthrones.fandom.com - Boooks, TV series	4,206	616
greekmythology.wikia.com - Mythology	1,726	1,537
mario.wikia.com - Console games	7,602	337
leagueoflegends.wikia.com - Video game	3,374	764

Table 4.1: Example of universes on Wikia.

universes around popular movies (e.g. *Star wars*), TV series (e.g. *Game of Thrones, Breaking Bad*), console games (e.g. *Super Mario*) and recent online games (e.g. *World of Warcraft, Leaque of Legends*). Table 4.1 shows the size of some well known universes and their ranks (w.r.t. size) on Wikia.

Method Wikia universes consist of pages, which are tagged with categories. E.g., the page of `Gimli` on the LoTR wiki[1] is tagged with the categories `Dwarves, Members of the Fellowship`, and `Elf friends`. Categories can be arranged hierarchally, for instance, the category `Maiar` is a subcategory of `Ainur`. We use the Wikia categories as starting points for distilling reference type systems.

Consolidation of the raw category systems is needed because (i) they frequently contain categories that are not types in the ontological sense, and (ii), because categories are frequently not properly semantically organized, i.e., contain disconnected low-level categories, and do not form a tree. We adopt techniques from the TiFi system to clean and structure the input categories. In particular, we remove irrelevant categories by use of a dictionary of meta-terms such as `wiki, template, user, portal`. We ensure a connected directed acyclic graph structure by linking top-level categories to the WordNet taxonomy. For this purpose, we use the descriptions of entities in a category as context, and link these contexts to most similar WordNet glosses. Having established the link to WordNet, we can then add further hypernyms as supertypes. The added types are compressed again by removing those that have only a single parent and a single child and those that are too abstract [Chu et al., 2019]. In the final type system, the root is `entity`, with two subclasses `physical_entity` and `abstract_entity`. Resulting type

[1] https://lotr.fandom.com/wiki/Gimli

systems typically contain between 700 to 10,000 types per universe.

4.5 Reference Universe Ranking

The goal of this step is selecting the reference type systems which are most useful for a given input text. To this end, we rely on cosine similarity between the bag of words in the input, and the texts that are hosted on Wikia for each reference universe. For the bag of words of the reference universes, we only use the entities and types, as these contain the most important information for determining suitability as reference. The top-ranked reference universes are then used for supervised classification as discussed in Section 4.7.1.

4.6 Mention Detection

Mention detection is an anterior step of entity typing. The goal is to detect the text spans that refer to entities. We treat this problem as BIOES tagging problem, i.e., each mention can be either an *S-mention* (singleton mention), or a combination of *B-mention* (begin of mention), *I-mention* (inside of mention) and *E-mention* (end of mention). At the same time, *non-mentions* are tagged as *O* (other).

Definition 4.6.1. ***Mention Tagging:*** *Given a sequence of words X, predict a sequence y, $\{y_i \in \{B, I, O, E, S\} | y_i \in y\}$ by maximizing the score of tag sequence:*

$$\hat{y} = f(X, y) \tag{4.1}$$

where $y \in Y$, is a collection of all possible tag sequences.

Inspired by the work in [He et al., 2017] from the field of semantic role labeling, we use a bidirectional 4-layer LSTM (BiLSTM) with embeddings and POS tags as input, with highway connections for avoiding vanishing gradients [Zhang et al., 2016], and recurrent dropout to reduce over-fitting [Gal and Ghahramani, 2016]. The final score of each label at each position is computed via a softmax layer.

BiLSTM Model The BiLSTM is defined as follow:

$$i_{l,t} = \sigma(W_i^l[h_{l,t}, x_{l,t}] + b_i^l) \quad (1)$$
$$o_{l,t} = \sigma(W_o^l[h_{l,t}, x_{l,t}] + b_o^l) \quad (2)$$
$$f_{l,t} = \sigma(W_f^l[h_{l,t}, x_{l,t}] + b_f^l + 1) \quad (3)$$
$$\tilde{c}_{l,t} = \tanh(W_c^l[h_{l,t}, x_{l,t}] + b_c^l) \quad (4)$$
$$c_{l,t} = i_{l,t} \bigodot \tilde{c}_{l,t} + f_{l,t} \bigodot c_t \quad (5)$$
$$h_{l,t} = o_{l,t} \bigodot \tanh(c_{l,t}) \quad (6)$$

where σ is the element-wise sigmoid function and $\odot$ is element-wise product, x_l, t is the input of LSTM at layer l and position t, represented as a d-dimensional vector which combine pre-trained embedding and POS tag features. The model combines multiple LSTM layer with bi-directionality interleavedly.

In particular, the input combines pre-trained embeddings and POS tag features, which are then processed in a multi-layer bidirectional LSTM (4 layers in our experiments). The final score of each label at each position is computed via a softmax layer.

$$p(y_t|X) \propto \exp(W_{tag}^y h_L^t + b_{tag}) \quad (7)$$

To further alleviate the vanishing gradient problem, transform gates r_t are also added between LSTM layers to control the weights.

$$r_{l,t} = \sigma(W_r^l[h_{l,t-1}, x_{l,t}] + b_r^l) \quad (8)$$
$$h'_{l,t} = o_{l,t} \bigodot \tanh(c_{l,t}) \quad (9)$$
$$h_{l,t} = r_{l,t} \bigodot h'_{l,t} + (1 - r_{l,t}) \bigodot W_h^l x_{l,t} \quad (10)$$

Figure 4.2 show an example of the model.

BIOES Constraint Decoding The output of the softmax layer is a collection of all possible tag sequences. Each prediction for a word w_i in the sequence is followed by a confidence score and in general, the BiLSTM model will return the tag sequence with maximum score. However, the final tag sequence (e.g. with maximum score) possibly harnesses BIOES constraints, for example, B tag should be followed by an I or E tag. Therefore, we propose a decoding step by using dynamic programming to select the tag sequence with maximum score and satisfying BIOES constraints.

- Tag O cannot be followed by tag I and E
- Tag B cannot be followed by tag O, B and S

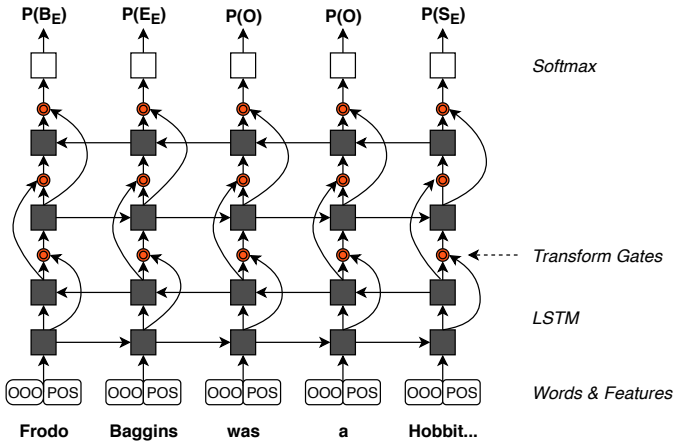


Figure 4.2: BiLSTM with highway connections between four layers

- Tag I cannot be followed by tag B, O and S
- Tag E cannot be followed by tag I and E
- Tag S cannot be followed by tag I and E

This decoding step improves the prediction results without adding complexity to the training stage. Our model is trained on the CoNLL-2003 datasets [Sang and De Meulder, 2003], a popular corpus for named entity recognition. We found that training on this data is also suited for mention detection in fiction, and retraining the model on Wikia texts would require extensive manual labelling, as the Wikia hyperlink markup would introduce too many false negatives.

4.7 Mention Typing

We next produce candidate types for mentions by a combination of supervised, unsupervised and lookup approaches.

4.7.1 Supervised Fiction Types

For predicting types from the reference type systems, as common for Wikipedia-centric approaches, we use textual mentions of hyperlinked entities with and without the type of interest as positive and negative training samples.

Our classification model resembles recent work on entity typing by using an attentive neural architecture [Shimaoka et al., 2017]. Although LSTMs can encode longer information in sequential data, this is not possible for selective encoding that focuses on local

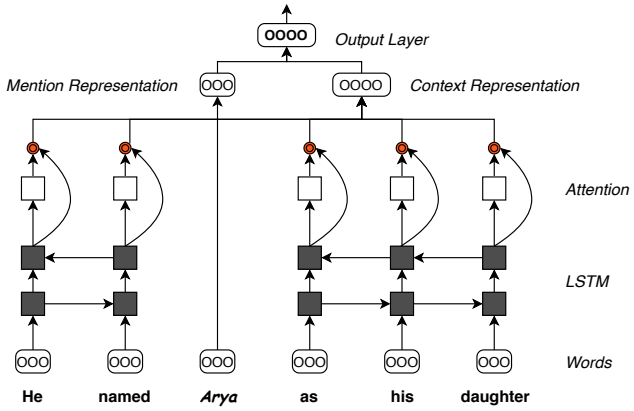


Figure 4.3: Attention model for supervised typing.

information relevant to the task, especially when the the input is long and rich. Attention mechanisms, on the other hand, can handle these issue by allowing the decoder to refer back to the input sequence [Young et al., 2018]. The model represents the mention and its context separately, before joining them into a final logistic regression layer (see Figure 4.3).

Mention Representation Averaging of all embeddings of tokens in the mention. Where available, we use precomputed embeddings to represent mentions (300-dimensional GloVe embeddings [Pennington et al., 2014]). In the case of out-of-vocabulary tokens, these are represented with a generic "unk" token.

Context Representation We consider both left and right context around mentions. First, the model encodes the sequences using BiLSTM models [Graves, 2012], and returns the output of the left and right context, respectively: $\overrightarrow{h_1^l}, \overleftarrow{h_1^l}, ..., \overrightarrow{h_C^l}, \overleftarrow{h_C^l}$, and $\overrightarrow{h_1^r}, \overleftarrow{h_1^r}, ..., \overrightarrow{h_C^r}, \overleftarrow{h_C^r}$ where C is the window size, and $\leftarrow, \rightarrow$ are directionalities of LSTM models ($C = 8$ in our experiments, mirroring [Shimaoka et al., 2017]). After that, an attention mechanism is used to compute weight factors (i.e. attentions) and integrates them to the output of BiLSTM layers. [Hermann et al., 2015].

Logistic Regression

In the end, the label of the entity mention is computed as:

$$y = \frac{1}{1 + \exp(-W_y \begin{bmatrix} v_m \\ v_c \end{bmatrix})}$$

where v_m, v_c are representations of the mention and its context. The loss function for a prediction is cross entropy loss:

$$L(y,t) = \sum_{k=1}^{K} -t_k \log(y_k) - (1-t_k)\log(1-y_k)$$

Target Classes We use two kinds of target classes: *(i) General types* - 7 disjunct and virtually exhaustive high-level WordNet types that we manually chose, mirroring existing coarse typing systems: `living_thing`, `location`, `organization`, `object`, `time`, `event`, `substance`. *(ii) Top-performing types* - As mentioned in Section 4.4, each reference universe has a type system containing between hundreds to thousands of types. Due to a large number of types as well as insufficient training data, predicting all types in the type systems is not effective. Therefore, from each reference universe, we predict those types for which, on withheld test data, at least 0.8 F1-score was achieved. This results an average of 75 types per reference universe.

4.7.2 Supervised Real-world Types

Fictional universes frequently overlap with the real world. A classic example is The Simpsons, a satire of middle class American life, but also fictional universes like Lord of the Rings or Game of Thrones contain types present in the real-world, like `King` or `Fortress`. To leverage the extensive training data available for these types, we incorporate the Wikipedia- and news-trained typing model from [Choi et al., 2018], which is theoretically able to predict up to 10,331 real-world types.

4.7.3 Unsupervised Typing

Types are frequently mentioned explicitly in context, e.g., *"King Robert was the ruler of Dragonstone Castle"* directly gives away that ROBERT is `King` and that DRAGONSTONE is a `Castle`. While supervised methods could in principle also predict these types, they would fail if the type is not in the type system, or comes with too few instances for training.

We therefore implement unsupervised extractors for explicit type mentions, relying on (i) Hearst-style patterns and (ii) dependency parses.

Hearst-style patterns We use 36 manually crafted Hearst-style patterns for type extraction, inspired by works in [Corro et al., 2015, Seitner et al., 2016]. Table 4.2 shows sample occurrences of these patterns.

Name	Example
Hearst I	{Valar} such [Varda] (and) [Mandos]
Hearst II	{Valar} like [Varda] (and) [Mandos]
Hearst III	[Varda] and other {Valar}
Hearst IV	{Valar} including [Varda] (and) [Mandos]
Other	[Varda] as {Valar}
Other	[Varda] among (other) {Valar}

Table 4.2: Examples of Hearst-style patterns.

Dependency parses We use the Stanford dependency parser to extract type candidates from the sentences. A noun phrase is considered as a type candidate if there exists a *noun compound modifier (nn)* relation between the noun phrase and the given mention. For example, from the sentence *"King Thranduil participated in the Battle of the Five Armies."* with the given mention Thranduil, the type candidate for Thranduil is `King`. In addition, in the case of the type term being part of the mention, we extract headwords of mentions and check whether they exist in WordNet as nouns. Headwords then become type candidates if the lookup is successful, for example, the mention Battle of Five Armies has the type candidate `Battle`.

4.7.4 KB Lookup

While human creativity is huge, many fictional texts, especially from fan fiction, are extensions or adaptions of existing story lines. The KB lookup aims to leverage entity reuse in similar context.

Specifically, we use the top-ranked reference universes as per Section 4.5 as basis for the lookup. For these universes, it is most likely that name matches refer to entities of same type, and are not just spurious homonyms. We map entity mentions to entities in the reference universes by exact lexical matching, deriving confidence scores from their frequency, in case a surface form appears several times across universes. We then return the types of the entity in the reference type system as type candidates for the input text.

In our test cases of fan fiction (i.e., texts that extend existing stories), lookups returned matches for typically between 5% and 30% of mentions.

4.8 Type Consolidation

Using type systems from multiple reference universes as the target of predictions may produce some noise. For example, Arda, a `location` in *The Lord of the Rings* can be

predicted as `wizard` using a deep learning model which is trained on *Harry Potter*. To resolve or mitigate such issues, we propose a consolidation stage based on an integer linear programming model (ILP).

Constraints Following constraints are defined for output types:

1. **Type Disjointness:** An entity cannot belong to two different general classes (section 4.7.1), for instance, `living_thing` and `location`.
2. **Transitive Type Disjointness:** Type disjointness is enforced also across hierarchies, e.g., `living_thing` and `city` are also incompatible.
3. **Hierarchical coherence:** If two type candidates stand in a hypernym relation, then either both or neither is returned.
4. **Cardinality limit**: To force ENTYFI to choose most relevant types only, we define a maximal number of types.
5. **Soft correlations:** In many cases, types exhibit positive or negative correlations. For instance, `Dwarves` are frequently portrayed as `Axe-wielders`, and rarely as `Archers`, or secret agents are frequently `Middle-aged single men`. To utilize such knowledge, we compute Pearson correlation coefficients v_{ij} between all type pairs (t_i, t_j) based on co-occurrences of types within entities. Knowledge about positive or negative correlations is then incorporated in the objective function below.

ILP Model Given an entity mention $\boldsymbol{e}$ with a list of type candidates with corresponding weights, we define a decision variable T_i for each type candidate t_i. $T_i = 1$ if $\boldsymbol{e}$ belongs to t_i, otherwise, $T_i = 0$. With the constraints above, the objective function is:

maximize

$$\alpha \sum_i T_i * w_i + (1 - \alpha) \sum_{i,j} T_i * T_j * v_{ij}$$

subject to

$$\begin{aligned} T_i + T_j &\leq 1 \quad \forall (t_i, t_j) \in D \\ T_i - T_j &\leq 0 \quad \forall (t_i, t_j) \in H \\ \sum_i T_i &\leq \delta \end{aligned}$$

where T_i is the decision variable for the type t_i with its weight w_i, α is a hyper parameter, D is the set of disjoint type pairs, H is the set of (transitive) hyponym pairs (t_i, t_j) - t_i is the (transitive) hyponym of t_j, and δ is the threshold for the cardinality limit.

In mention typing step, each mention appearing in the text is labeled separately, based on the context. Therefore, two mentions, even with the same surface form, can have different sets of type candidates. We aggregate type candidates of all mention with the same surface form and run ILP on it, using the Pulp library[2]. For example, FRODO has type candidates `character` (weight 0.6, returned by supervised-module) and `ring bearer` (weight 0.8, returned by KB lookup) in context 1, but `character` (weight 0.5, return by supervised-module), `hobbit` (weight 1.0, return by unsupervied-module) and `ring bearer` (weight 0.8, returned by KB lookup) in context 2. After aggregation, ILP model will run on the entity mention FRODO, which have the list of type candidates: `character` (weight 1.1), `ring bearer` (weight 1.6) and `hobbit` (weight 1.0).

4.9 Experiments

We conducted extensive experiments to assess the viability of our approach and the quality of the resulting entity typing. Our main experiments include two parts, (1) automatic end-to-end evaluation which automatically creates the test data and doing entity typing on them (Section 4.9.2), (2) crowdsourced end-to-end evaluation, on the other hand, takes the input from random texts, and evaluates the results by using crowdsourcing (Section 4.9.3). We also examine the performance of each module in our system by doing an ablation study (section 4.9.4) and finally, testing ENTYFI in unconventional real-world domains (Section 4.9.5).

4.9.1 Test Data

We downloaded all Wikia domains which have a dump file and contain at least 1000 content pages, resulting in a total of 205 universes. Using these universes as references, after type system ranking, we then focus on types from the top-3 most similar universes.

For automated evaluation, as the test data, we use five randomly selected Wikia universes that are withheld from the reference set. Since Wikia type systems are typically noisy, we apply the following cleaning steps before considering their entity types as ground truth. First, lexicon-based heuristics are applied to remove meta-categories. The type systems are then integrated with top-level types from WordNet [Chu et al., 2019]. Second, we only keep types for which the number of entities exceeds a threshold, set to 5 for the experiments. This heuristics removes overly specific types. Third, we enforce disjointness constraints to remove spurious subclass relations. For example, an entity can

[2]https://pypi.org/project/PuLP/

not belong to both `physical_entity` and `abstract_entity`. Fourth, we consider only the headword of multi-word type names as target. This serves to map overly specific types onto more general types. For example, `hobbits from the Brandywine valley` become `hobbits`, and `red-scaled dragons` become `dragons`.

These pre-processing steps result in 5 universes: *Ghost Recon*[3], *Dead or Alive*[4], *Reindeers*[5], *Injustice Fanon*[6] and *Hawaii Five-O*[7]. The text of each universe is extracted from articles about entities (e.g. `character` NOMAD in *Ghost Recon*), as well as plots/summaries which containt narrative information (e.g. episode HE MOHO HOU of season 7 in *Hawaii Five-O*). The number of entity mentions in the test data of each universe varies from 385 to 3002, with an average of 1602, and the number of entity types in the original type systems extracted from Wikia is 317 on average. After cleaning the type systems, the total number of distinct ground-truth types is about 30 per universe. This reduction serves to focus on notable types for which entity mentions in the Wikia articles have markup with linkage to an entity repository with ground-truth types.

4.9.2 Automated End-to-End Evaluation

Baselines We compare ENTYFI against two state-of-the-art baselines and their variations:

- **NFGEC-WP** [Shimaoka et al., 2017] devised an attentive neural network for fine-grained entity type classification. In our experiments, the model is trained using the original code and the original data of [Shimaoka et al., 2017]. The dataset includes 2,000,000 instances for training, 10,000 for development and 563 for testing, with total of 112 fine-grained types. The train and dev set are extracted from Wikipedia articles while the test set is manually annotated from new articles.
- **UF-WP** [Choi et al., 2018] uses neural learning with attention for ultra-fine entity typing with a multi-task objective model. We employ the released model trained on a large dataset extracted from Wikipedia and OntoNotes, with total of 10,331 fine-grained types. To the best of our knowledge, this is the state-of-the-art method for entity typing on regular texts.
- **NFGEC-Wikia** and **UF-Wikia** The same models as NFGEC-WP and UF-WP, respectively, but re-trained by us using top-k Wikia universes with the highest

[3]https://ghostrecon.fandom.com
[4]https://deadoralive.fandom.com
[5]https://reindeers.fandom.com
[6]https://injusticefanon.fandom.com
[7]https://hawaiifiveo.fandom.com

Metric	Method	w/o relaxation			w/ 2-relaxation		
		P	**R**	**F1**	**P**	**R**	**F1**
Loose macro	NFGEC-WP	6.39	4.55	5.30	44.74	26.25	32.76
	UF-WP	12.27	10.96	11.32	46.99	47.86	45.67
	NFGEC-Wikia	**27.31**	20.98	23.02	36.75	34.86	34.48
	UF-Wikia	20.50	22.88	21.10	34.12	40.46	36.36
	NFGEC-All	3.57	2.34	2.82	35.71	19.62	25.10
	UF-All	24.55	13.80	17.11	**50.98**	37.00	41.58
	ENTYFI	22.61	**26.68**	**23.47**	40.22	**65.90**	**49.37**
Loose micro	NFGEC-WP	7.76	2.54	3.80	44.39	25.82	32.37
	UF-WP	13.18	7.93	9.73	42.71	47.45	43.30
	NFGEC-Wikia	**25.49**	19.09	21.41	34.59	31.98	32.33
	UF-Wikia	19.96	19.02	19.19	33.13	37.25	34.46
	NFGEC-All	4.44	1.28	1.97	35.88	19.99	25.47
	UF-All	25.11	19.96	14.74	**45.41**	35.81	38.94
	ENTYFI	22.69	**23.95**	**22.40**	40.36	**65.90**	**49.18**

Table 4.3: Avg. precision, recall and F1 in automated eval.

bag-of-words similarity to the input texts. For a fair comparison, the top-k Wikia universes are the same as for ENTYFI, i.e., $k = 3$.

- **UF-All** and **NFGEC-All**. The same models as UF-WP and NFGEC-WP, respectively, but re-trained using original data (e.g. Wikipedia) and top-k Wikia universes ($k = 3$).

Metrics We use precision, recall and F1 metrics for evaluation, following [Ling and Weld, 2012]. Consider a set of mentions with ground-truth types as E_R, and a set of mentions with predicted types as E_P. For each mention e, the set of ground-truth types of e is denoted as r_e and the set of headwords of the predicted types as p_e. Two metrics are defined on this basis.

In *loose macro*, precision and recall are computed for each mention, and these measures are then averaged over all mentions.

$$precision = \frac{1}{|E_P|} \sum_{e \in E_P} \frac{|r_e \cap p_e|}{|p_e|} \qquad recall = \frac{1}{|E_R|} \sum_{e \in E_R} \frac{|r_e \cap p_e|}{|r_e|}$$

In *loose micro*, precision and recall are computed for each mention-type pair, and these measures are then averaged over all pairs.

$$precision = \frac{\sum_{e \in E_P} |r_e \cap p_e|}{\sum_{e \in E_P} |p_e|} \qquad recall = \frac{\sum_{e \in E_R} |r_e \cap p_e|}{\sum_{e \in E_R} |r_e|}$$

Note that E_p and E_r are sets, and the above measures consider the set overlap rather than equality of sets. Hence the term *Loose* macro/micro precision and recall [Ling and Weld, 2012]. In both macro and micro averaging, the F1-score is defined as follows.

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Relaxed Metrics The original metrics treat all mismatches between ground-truth and classification output uniformly as errors. However, the classifier may yield a type that is semantically near the ground-truth, for example, by predicting a type that is a hypernym or hyponym of the ground-truth type (e.g., predicting `Urukai Orks` for a mention of type `Orks`). Therefore, we consider also the following relaxed metrics for evaluation, called k-relaxation, which reflects the relatedness between prediction and ground-truth. Under this metric, we consider all pairs $\langle p_e, r_e \rangle$ of predicted and truly valid types as a match if their distance in the hypernymy graph of the type system is at most k. That is, p_e is either a hyponym of r_e at most k hops down or a hypernym at most k hops up. In practice, we set k to 2.

Results For each universe in the test data, we take the top 3 universes for the ranking step (section 4.5). For the ILP model, we limit the number of predicted types to 5. For fair comparison to the baselines, we also consider only their top 5 predicted types (based on their scoring models).

Table 4.3 shows the results of ENTYFI and the baselines, for both original metrics and relaxed metrics. ENTYFI achieves substantially higher F1 scores than all baselines. Without using relaxed metrics, the original baselines (NFGEC-WP and UF-WP) achieve F1 scores of no more than 11.32%, while ENTYFI achieves F1 scores of over 20% (23.47% macro- and 22.40% micro-averaged). Although the baselines perform considerably better when using Wikia for training, their F1-scores are still 1% to 3% lower than ENTYFI. We observed that the baselines often predict rather coarse-grained types such as `person, location`; these predictions are correct albeit not exactly specific. Thus, the baselines tend to be better than our method in terms of precision. On the other hand, ENTYFI predicts more fine-grained types for entities (e.g. `wizard, hobbit`), hence achieving much better recall.

When applying relaxed metrics that account for outputs that are semantically close to the ground-truth, ENTYFI outperforms all baselines by a large margin. ENTYFI achieves an F1 score of 49%, while NFGEC-WP only achieves F1 scores of 32.8% and 32.4% macro- and micro-averaged, respectively. For UF-WP, these numbers are 45.7%

and 43.3%

4.9.3 Crowdsourced End-to-End Evaluation

Data For human evaluation on text from totally unseen genres, we randomly selected inputs from the following sources.

- **Books** are a stress test for entity typing methods. We randomly selected a fiction book from the website wikisource.org, namely, *The Book of Dragons*[8], and randomly selected a chapter with a total of 40k words.

- **Short Stories** in the fantasy domain are sometimes written by fans and amateur writers, either based on existing universes (e.g., your own alternative ending of Game of Thrones) or having totally new fantasy content. **Fanfiction**[9] is a community that features such stories; we randomly selected three stories from this site:

  - **The Sisters, the Compass and the Lion**, based on the book *Chronicles of Narnia*: 4 chapters, 15k words.

  - **Stigmata Reign**, based on the book *Darkside series, Tom Becker*: 1 chapter, 1251 words.

  - **Lies That Wear the Crown**, based on the book *Hobbit*: 6 chapters, 10k words.

Crowdsourcing Task Design We devised a crowdsourcing task for the assessment of the typing outputs, using the Figure-Eight platform. In addition to a short overview of the book or story, we provided workers with the context of a given entity mention (e.g. for stories a single sentence). Then the worker is asked if a mention does indeed belong to the types predicted by the various methods under test. A sample question posed to the workers is *Following the above story, is it the case that the entity* Gondolin *belongs to the class* `city`*?*. Since the content of books is large, with each mention, we provide three different contexts (e.g. small paragraph) in which the mention appears. For each mention to be assessed, we had at least three workers, and interpret the majority label as ground-truth. We observed very high inter-annotator agreement, with average label confidence of 0.88 as computed by the platform.

[8] https://en.wikisource.org/wiki/The_Book_of_Dragons
[9] https://www.fanfiction.net/

Source		UF-WP		ENTYFI	
		Macro	Micro	Macro	Micro
Fan fiction	Hobbit	41.86	37.19	**64.78**	**64.81**
	Tom Becker	32.66	20.06	**57.92**	**55.36**
	Chronicles of Narnia	34.42	17.10	**75.44**	**76.07**
	Average	36.31	24.78	**66.05**	**65.42**
Books	The Book of Dragons	37.05	36.50	**49.92**	**52.46**

Table 4.4: Loose- macro and micro precision in crowd. eval.

Results Table 4.4 shows the results. ENTYFI outperforms the best baseline UF-WP on these texts by 12%-41% in loose macro precision, and 14%-59% in loose macro precision. Although UF-WP is trained on a large dataset with over 10000 types, these results emphasize that there is still a significant gap between real-world and fiction typing.

4.9.4 Component Evaluation

Type System Construction Our type system construction uses the technique from [Chu et al., 2019], which includes removing meta-categories (e.g., `Season 8`) and aligning universe-specific types with (generalizations in) WordNet. To evaluate meta-category cleaning, for each of 5 random universes, we randomly select 50 categories which are removed by our method and check whether they are indeed meta-categories. The results show that our technique achieves near-perfect precision of 99% on removing meta-categories. For the alignment with WordNet, categories need to be linked to corresponding WordNet synsets. To evaluate this step, we randomly select 50 such links and evaluate their correctness, resulting in precision between 84% and 92% (comparable to the results in [Chu et al., 2019]). Table 4.5 shows examples of type systems of several universes after applying our method for type system construction. Note that we also add new types to the type systems by linking to WordNet. For example, in GoT, 57 nodes from WordNet are added into the type system, while in LoTR, this number is 91.

Universe	#Types	#Edges	Max. depth	Avg. #Child./Type
Lord of the Rings	637	1,163	18	4.4
Game of Thrones	536	1,219	15	6.8
Harry Potter	2,039	4,267	28	4.6
Star wars	8,491	16,110	26	6.1
Disney	1,332	3,665	19	5.4

Table 4.5: Examples of constructed reference type systems.

Model	4 Tags PER, LOC, ORG, MISC	1 Tag ENTITY
Ori. Model (OM)	86.66	90.05
OM + Decoding	87.22	90.17
OM + Pos	88.42	**93.51**
OM + Pos + Decoding	**88.95**	93.24

Table 4.6: F1-score of mention detection on CoNLL-2003.

Mention Detection In this experiment, we test our mention detection method on the CoNLL-2003 dataset [Sang and De Meulder, 2003], a popular corpus for evaluating named entity recognition. We compare the original model (LSTM + highway connection) with our proposed model, with 4 LSTM layers and using POS tags as additional features and decoding on the prediction step.

Table 4.6 gives the results of our method for two different outputs: (1) detecting and labeling mentions into 4 tags: PER, LOC, ORG, MISC, and (2) simply detecting mentions (1 tag: ENTITY). The results show that using POS tags and the decoding step help our method to outperform the original model in F1 score by approximatetely 2.5% and 3.5%, respectively.

Ablation Study The experiments presented here serve to evaluate the influence of the various components of ENTYFI. We compare the complete end-to-end ENTYFI system against variants where ILP (sec. 4.8), supervised fiction typing (SUPWKA - sec. 4.7.1), supervised real-world typing (SUPWKP - sec. 4.7.2), unsupervised (UNSUP - sec. 4.7.3) and KB lookups (sec. 4.7.4) are disabled. Table 4.7 shows how these variants perform on the test data. The supervised modules are most important, followed by unsupervised and KB lookups.

Method	Loose Macro			Loose Micro		
	P	**R**	**F1**	**P**	**R**	**F1**
w/o SupWKA	11.48	14.39	12.46	11.69	11.21	11.16
w/o SupWKP	20.64	21.60	20.29	20.81	21.42	20.22
w/o UNSUP	19.91	22.97	20.50	19.94	20.86	19.59
w/o KB	19.87	23.01	20.51	19.96	20.94	19.64
w/o ILP	20.46	**27.78**	22.76	20.57	**24.75**	21.78
Full ENTYFI	**22.61**	26.68	**23.47**	**22.69**	23.95	**22.40**

Table 4.7: ENTYFI ablation study – without relaxation.

Sample Context	Sample Mention	ENTYFI	UF-WP
...The Wizard counted , and it turned out the **Halfling** was nowhere to be seen...	Halfling	characters, living_thing, ~~mobs~~, races	~~communicator~~, ~~location~~
...With Steve now innocent , Jameson s replacement , **Governor Samuel Denning** reinstates Five 0 except for Kono who is still being investigated by Internal Affairs...	Governor Samuel Denning	living_thing, person, governor politician, ~~reference~~	person, politician, governor
..."A lot of these cartoons were aimed at convincing Americans of German heritage they were victims of a Jewish-led assault on their culture , especially the shorts starring Heinrich , Diedrick , and Ludwig , " said **Bryant**, referencing the duckling brothers better known as **Huey** , Dewey , and Louie ...	Bryant	~~actor~~, ~~artist~~, person	~~city~~, ~~artist~~, person, ~~location~~, ~~actor~~, ~~basketball_player~~
	Huey	animated_characers, characters, disney_characters, ~~people~~, ~~television~~	~~person~~, ~~actor~~, ~~artist~~
...They sell furs ... But the journey to **Rohan** became unsafe in the latest years...	Rohan	kingdoms, location, ~~mobs~~, places, ~~races~~	~~person~~, ~~title~~

Table 4.8: Anecdotal examples for the outputs of ENTYFI and the baseline.

Anecdotal Examples Table 4.8 shows examples of ENTYFI outputs, compared to the strongest baseline UF-WP. The crossed-out words denote false positive. Generally, UF-WP performs well with entities which have real types (e.g. `person, company`) but is not able to predict types for fictional entities. Moreover, following the results returned by UF-WP, an entity can belong to two semantically unrelated types (e.g. BRYANT is both a `city` and `person`), which is unreasonable. ENTYFI, on the other hand, by using consolidation, can remove this incompatibility. Although there are still mispredictions (e.g. real-world and fictional types), ENTYFI is able to predict reasonable types for entity mentions at fine-grained level on fictional texts.

4.9.5 Unconventional Real-world Domains

Historical Texts Historical texts differ from fantasy and mythology, as they refer to entities and events of real-world history. Many of the types in these domains are reasonably mainstream (e.g., `soldier, battle, politician`), but the entities themselves (e.g., `centurion Gaius Crastinus`) and the language in historical texts are rather non-standard – so methods geared for today's news do not easily carry over.

As test data for this genre, we selected three long Wikipedia articles about the Maya civilization[10], the Viking Age[11] and the Roman Empire[12]. We compare ENTYFI against the best performing baseline, UF-WP.

To evaluate the outputs of these methods, we conducted a crowd-sourcing task, similar to Section 4.9.3. The results show that ENTYFI significantly outperforms UF-WP on two texts, Maya Civilization and Roman Empire, and achieves comparable results on Viking Age. Overall, ENTYFI achieves substantially higher precision for both macro and micro averaging: 71.64% and 70.88%, compared to 63.07% and 56.85% by UF-WP, respectively. Interestingly, because UF-WP uses distant supervision to collect training

[10]https://en.wikipedia.org/wiki/Maya_civilization#History
[11]https://en.wikipedia.org/wiki/Viking_Age#Historic_overview
[12]https://en.wikipedia.org/wiki/Roman_Empire#History

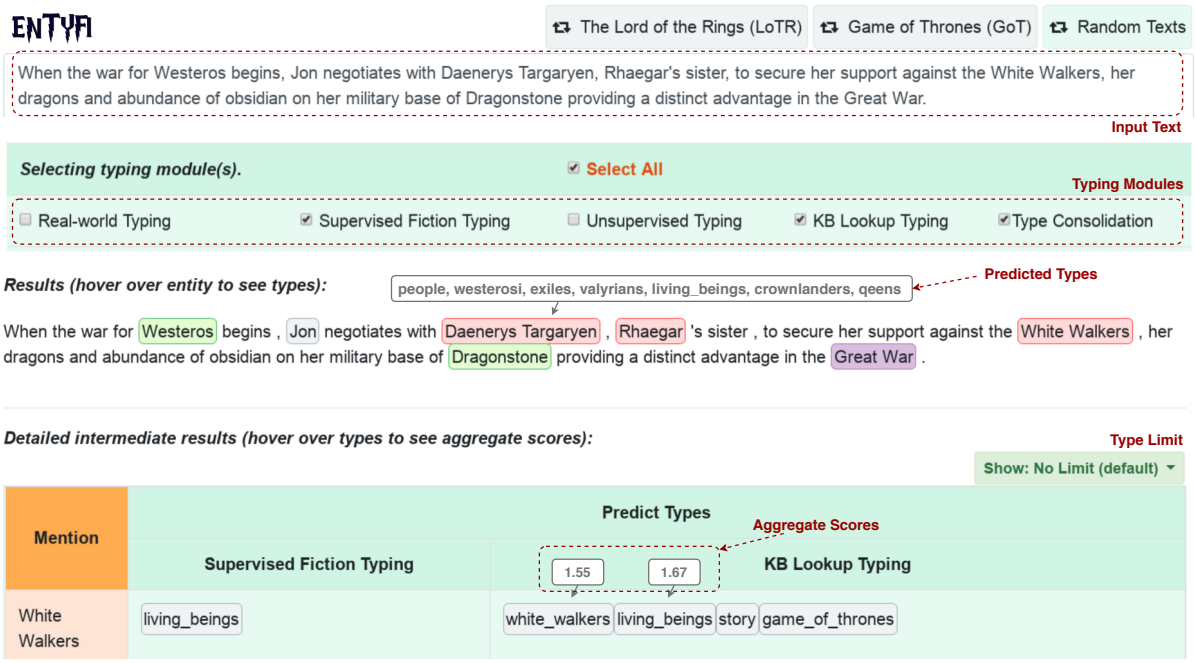


Figure 4.4: ENTYFI Web interface.

data with texts from Wikipedia including history articles, UF-WP performs much better on these texts, compared to fictional texts. ENTYFI, by integrating a real-world typing module, achieves good results also on these unconventional texts.

Satirical News Satirical news often feature both real-world entities and fictional ones (e.g., invented characters in a story). Their content is exaggerated or absurd, but many aspects and the language style still mimic genuine news. An additional challenge here is that some entities may be associated with exotic types (e.g., Donald Trump featured as a musician).

To study the performance of ENTYFI on these texts, we randomly selected three satirical news from the magazine theonion.com. We also compare ENTYFI with UF-WP by crowd-sourced assessment of the typing outputs. The results show that ENTYFI significantly outperforms UF-WP, with substantially higher precision for both macro and micro averaging: 54.02% and 53.98%, compared to 46.47% and 43.70% of UF-WP, respectively.

4.10 ENTYFI Demonstration

To illustrate ENTYFI, a web-based system of ENTYFI was deployed. Users can exploit the richness and diversity of these reference type systems for fine-grained supervised typing, in addition, they can choose among and combine four other typing modules: pre-trained real-world models, unsupervised dependency-based typing, knowledge base lookups, and constraint-based candidate consolidation. The demonstrator is available at https://d5demos.mpi-inf.mpg.de/entyfi. We also provide a screencast video demonstrating our system, at: https://youtu.be/g_ESaONagFQ.

4.10.1 Web Interface

Input The web interface allows users to enter a text as input. To give a better experience, we provide various sample texts from three different sources: Wikia, books and fan fiction[13]. With each source, users can try with either texts from Lord of the Rings and Game of Thrones or random texts, as well as some cross-overs between different universes written by fans.

Output Given an input text, users can choose different typing modules to run. The output is the input text marked by entity mentions and their predicted types. The system also shows the predicted types with their aggregate scores and the typing modules from which the types are extracted. Figure 4.4 shows an example input and output of the ENTYFI demo system.

Typing module selector ENTYFI includes several typing modules, among which users can choose. If only the real-world typing module is chosen, the system runs typing on the text immediately, using one of the existing typing models which are able to predict up to 112 real-world types [Shimaoka et al., 2017] or 10,331 types [Choi et al., 2018]. *Note:* If the later model is selected to run the real-world typing, it requires more time to load the pre-trained embeddings [Pennington et al., 2014].

On the other hand, if supervised fiction typing or KB lookup typing are chosen, the system computes the similarity between the given text and reference universes from the database. With the default option, the type system of the most related universe is being used as targets for typing, while with the alternative case, users can choose different universes and use their type systems as targets. Users are also able to decide whether the consolidation step is executed or not.

[13]https://www.fanfiction.net/

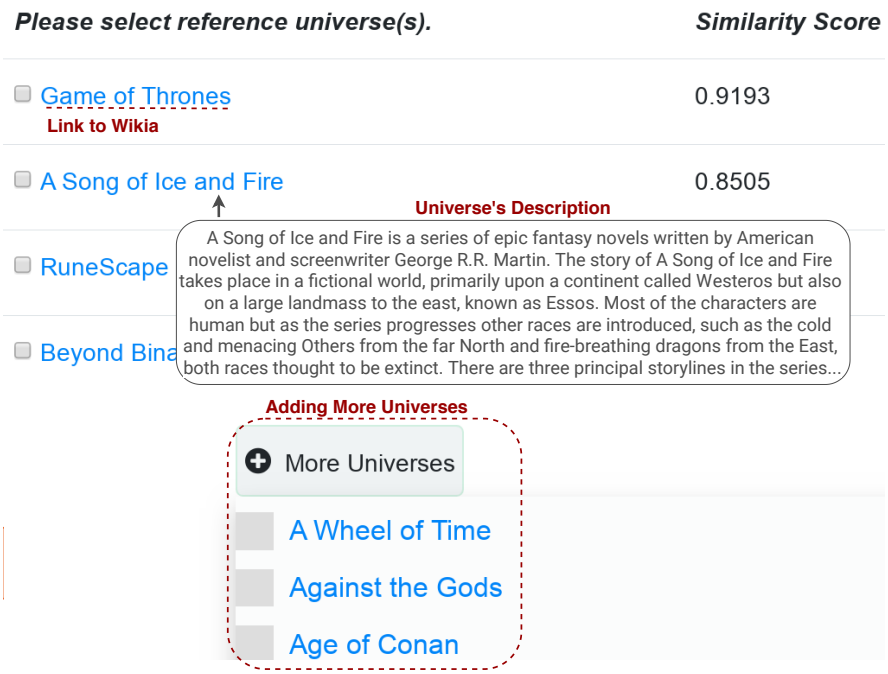


Figure 4.5: ENTYFI Reference Universes.

Exploration of reference universes ENTYFI builds on 205 automatically induced high-quality type systems for popular fictional domains. Along with top 5 most relevant universes showing up with similarity scores, users can also choose other universes in the database. For a better overview, with each universe, we provide a short description about the universe and a hyperlink to its Wikia source. Figure 4.5 show an example of reference universes presented in the demonstration.

Logs To help users understand how the system works inside, we provide a log box that shows which step is running at the backend, step by step, along with timing information (Figure 4.6).

4.10.2 Demonstration Experience

A common use of entity typing is as building block of more comprehensive NLP pipelines that perform tasks such as entity linking, relation extraction or question answering. We envision that ENTYFI could strengthen such pipelines considerably (see also extrinsic evaluation in [Chu et al., 2020a]). Yet to illustrate its workings in isolation, in the following, we present a direct expert end-user application of entity typing in fictional texts.

Suppose a literature analyst is doing research on a collection of unfamiliar short stories

Figure 4.6: ENTYFI Logs.

Mention	Settings		
	Default (Ref. universes + all modules)	Default without type consolidation	Only real-world typing
Elladan & Elrohir	men, hybrid peoples, elves of rivendell, real world, elves, characters, living thing, antagonists, supernatural, species, etc.	organization, men, the silmarillion characters, hybrid peoples, elves of rivendell, elves, characters, living thing, location, antagonists, vampire diaries characters, supernatural, etc.	athlete, god, character, body part, arm, person, goddess, companion, brother, child
Redhorn	creatures, villains, servants of morgoth, real world, minions of angmar, servants of sauron, species, living thing, characters, witches, supernatural, one	creatures, villains, evil, death, deaths in battle, servants of morgoth, minions of angmar, servants of sauron, characters, witches, places, arda, races, living thing, organization, etc.	city, god, tribe, county, holiday, body part, society, product, mountain, act
Imladris	kingdoms, location, realms, landforms, places, elven cities, eriador, elven realms, mordor, etc.	kingdoms, location, realms, arda, landforms, places, continents, organization, elven cities, etc.	city, writing, setting, castle, clan, location, character, eleven, etc.

Table 4.9: Results of ENTYFI on different settings.

from fanfiction.net. Their goal is to understand the setting of each story, to answer questions such as what the stories are about (e.g. politics or supernatural), what types of characters the authors create, finding all instances of a type or a combination of types (e.g. female elves) or to do further analysis like if female elves are more frequent than male elves and if there are patterns regarding where female villains appear mostly. Due to time constraints, the analyst cannot read all of stories manually. Instead of that, they can run ENTYFI on each story to extract the entity type system automatically. For instance, to analyze the story *Time Can't Heal Wounds Like These*[14], the analyst would paste the introduction of the story into the web interface of ENTYFI.

"Elladan and Elrohir are captured along with their mother, and in the pits below the unforgiving Redhorn one twin finds his final resting place. In a series of devastating events Imladris loose one of its princes and its lady. But everything is not over yet, and those left behind must lean to cope and fight on."

Since they have no prior knowledge on the setting, they could let ENTYFI propose related universes for typing. After computing the similarity between the input and the ref-

[14] *https://www.fanfiction.net/s/13484688/1/Time-Can-t-Heal-Wounds-Like-These*

erence universes from the database, ENTYFI would then propose *The Lord of the Rings, Vampires Diaries, Kid Icarus, Twin Peaks* and *Crossfire* as top 5 reference universes, respectively. The analyst may consider *The Lord of the Rings* and *Vampires Diaries*, top 2 in ranking, of particular interest, and in addition, select the universe *Forgotten Realms*, because that is influential in their literary domain. The analyst would then run ENTYFI with default settings, and get a list of entities with their predicted types as results. They could then see that ELLADAN and ELROHIR are recognized as `living thing, elves, hybrid people` and `characters`, while REDHORN as `living thing, villains, servants of morgoth`, and IMLADRIS as `location, kingdoms, landforms` and `elven cities`.

They could then decide to rerun the analysis with reference universes *The Lord of the Rings* and *Vampires Diaries* but without running type consolidation. By ignoring this module, the number of predicted types for each entity increases. Especially, ELLADAN & EHROHIR now are classified as `living thing, elves, characters`, but also `location` and `organization`. Similarly, REDHORN belongs to both `living thing` and `places`, while IMLADRIS is both a `kingdom` and a `devastating event`. Apparently, these incompatibilities in predictions appear when the system does not run type consolidation.

The analyst may wonder how the system performs when no reference universe is being used. By only selecting the real-world typing module [Choi et al., 2018], the predicted types for ELLADAN & ELROHIR would change to `athlete, god, body part, arm, etc.` REDHORN now becomes a `city, god, tribe` and even an `act`, while IMLADRIS is a `city, writing, setting` and `castle`. The results show not only incompatible predictions, but also that the existing typing model in the real world domain lacks coverage on fictional domains. By using a database of fictional universes as reference, ENTYFI is able to fill these gaps, predict fictional types in a fine-grained level and remove incompatibilities in the final results. From this interaction, the literature analyst could conclude that the story is much related to *The Lord of the Rings*, which might help them to draw parallels and direct further manual investigations. Table 4.9 shows the result of this demonstration experience in details.

4.11 Summary

In this chapter, we have presented ENTYFI, a 5-step methodology towards typing mentions in non-standard domains with long-tail types. For the specific use case of fiction, we have distilled high-quality reference type systems from fan Wikis, and shown that a combination of supervised fiction typing, supervised real-world typing, unsupervised typing and KB lookups significantly outperforms state-of-the-art supervised-

only typing methods. Experiments showed that ENTYFI is also useful for real-world texts such as history or satire. Code and data of ENTYFI are available at https://www.mpi-inf.mpg.de/yago-naga/entyfi.

Chapter 5

KnowFi: Knowledge Extraction from Long Fictional Texts

5.1 Introduction

Motivation and Problem: Relation extraction (RE) from web contents is a key task for the automatic construction of knowledge bases (KB). It involves detecting a pair of entities in a text document and inferring if a certain relation (predicate) holds between them. Extracted triples of the form (subject, predicate, object) are used for populating and growing the KB. Besides this major use case, RE also serves other applications like text annotation and summarization, semantic search, and more.

Work on KB construction has mostly focused on general-purpose encyclopedic knowledge, about prominent people, places, products etc. and basic relations of wide interest such as birthplaces, spouses, writing of books, acting in movies etc. Vertical domains have received some attention, including health, food, and consumer products. Yet another case are KBs about fictional works [Hertling and Paulheim, 2020, Labatut and Bost, 2019], such as Game of Thrones (GoT), the Marvel Comics (MC) universe, Greek Mythology or epic books such as War and Peace by Leo Tolstoy or the Cartel novels by Don Winslow. For KBs about fictional domains, the focus is less on basic relations like birthplaces or spouses, but more on relations that capture traits of characters and key elements of the narration. Relations of interest are allies, enemies, membership in clans, betrayed, killed etc.

Applications of fiction KBs foremost including supporting fans in entity-centric search. Some of the fictional domains have huge fan communities, and search engines frequently receive queries such as "Who killed Catelyn Stark?" (in GoT). Entity summarization is a related task, for example, a user asking for the most salient traits of Ygritte (in GoT). Although fiction serves to entertain, some of the more complex domains reflect sub-

cultural trends and the zeitgeist of certain epochs. Analyzing their narrative structures and networks of entities is of interest to humanities scholars. For example, superhero comics originated in the 1940s and boomed in post-war years, reflecting that era's zeitgeist (revived now). War and Peace has the backdrop of the Napoleonic wars in Russia, and the Cartel trilogy blends facts and fiction about drug trafficking. KBs enable deeper analyses of such complex texts for historians, social scientists, media psychologists and cultural-studies scholars.

State of the Art and its Limitations: RE with pre-specified relations for canonicalized entities is based on distant supervision via pre-compiled seed triples [Mintz et al., 2009, Suchanek et al., 2009]. Typically, these training seeds come from initial KBs, which in turn draw on Wikipedia infoboxes. The best RE methods are based on this paradigm of distant supervision, leveraging it for neural learning (e.g., [Han et al., 2020b, Soares et al., 2019, Wang et al., 2020, Yao et al., 2019, Zhang et al., 2017a]). They work well for basic relations, as there is no shortage of training samples (e.g., for birthplace or spouse). One of their key limitations is the bounded size of input text passages, typically a few hundred tokens only. This is not a bottleneck for basic relations where single sentences (or short paragraphs) with all three SPO components are frequent enough (e.g., in the full text of Wikipedia articles). However, for RE with non-standard relations over long fictional texts such as entire books, these limitations are major bottlenecks, if not showstoppers. This paper addresses the resulting challenges (also included among the open challenges in the overview by [Han et al., 2020b]):

- How to go about distant supervision for RE targeting non-standard relations that have only few seed triples?
- How to cope with very long input texts, such as entire books, where relevant cues for RE are spread across passages?

Approach and Contributions: This chapter presents a complete methodology and system for relation extraction from long fictional texts, called *KnowFi* (Knowledge extraction from Fictional texts). Our method leverages semi-structured content in wikis of fan communities on `fandom.com` (aka `wikia.com`). We extract an initial KB of background knowledge for 142 popular domains (TV series, movies, games). This serves to identify interesting relations and to collect distant supervision samples. Yet for many relations this results in very few seeds. To overcome this sparseness challenge and to generalize the training across the wide variety of relations, we devise a similarity-based ranking technique for matching seeds in text passages. Given a long input text, KnowFi judiciously selects a number of context passages containing seed pairs of entities. To infer if a certain relation holds between two entities, KnowFi's neural network is trained

jointly for all relations as a multi-label classifier.

Extensive experiments with long books on five different fictional domains show that KnowFi clearly outperforms state-of-the-art RE methods. Even on conventional short-text benchmarks with standard relations, KnowFi is competitive with the best baselines. As an extrinsic use case, we demonstrate the value of KnowFi's KB for the task of entity summarization. The paper's novel contributions are:

- a system architecture for the new problem of relation extraction from long fictional texts, like entire novels and text contents by fan communities (Section 5.3).
- a method to overcome the challenge of sparse samples for distant supervision for non-standard relations (Section 5.4).
- a method to overcome the challenge of limited input size for neural learners, by judiciously selecting relevant contexts and aggregating results (Section 5.5).
- a comprehensive experimental evaluation with a novel benchmark for relation extraction from very long documents (Section 5.6), with code and data release upon publication.

5.2 Related Work

Relation Extraction (RE): Early work on RE from text sources has used rules and patterns, (e.g., [Agichtein and Gravano, 2000, Craven et al., 1998, Etzioni et al., 2004, Reiss et al., 2008]), with pattern learning based on the principle of relation-pattern duality [Brin, 1998]. Open IE [Banko et al., 2007, Mausam, 2016, Stanovsky et al., 2018] uses linguistic cues to jointly infer patterns and triples, but lacks proper normalization of SPO arguments. RE with pre-specified relations, on the other hand, is usually based on distant supervision via pre-compiled seed triples [Mintz et al., 2009, Suchanek et al., 2009]. A variety of methods have been developed on this paradigm, from probabilistic graphical models (e.g., [Pujara et al., 2015, Sa et al., 2017]) to deep neural networks (e.g., [Han et al., 2020b, Soares et al., 2019, Wang et al., 2020, Yao et al., 2019, Zhang et al., 2017a]). Distantly supervised neural learning has become the method of choice, with different granularities.

Sentence-level RE: Most neural methods operate on a per-sentence level. Distant-supervision samples of SPO triples serve to identify sentences that contain an entity pair (S and O) which stand in a certain relation. The sentence is then treated as a positive training sample for the neural learner. At test-time, the trained model can tag entity mentions and predict if the sentence expresses a given relation or not. This

basic architecture has been advanced with bi-LSTMs, attention mechanisms and other techniques (e.g., [Cui et al., 2018, Trisedya et al., 2019, Zhang et al., 2017a]). A widely used benchmark for sentence-level RE is TacRed [Zhang et al., 2017a].

With recent advances on pre-trained language models like BERT [Devlin et al., 2019a] (or ElMo, GPT-3, T-5 and the like), the currently best RE methods leverage this asset for representation learning [Shi and Lin, 2019, Soares et al., 2019, Wadden et al., 2019, Yu et al., 2020].

Document-level RE: To expand the scope of inputs, Wang et al. [2019] proposed RE from documents, introducing the DocRed benchmark. However, the notion of documents is still very limited in size, given the restrictions in neural network inputs, typically around 10 sentences (e.g., excerpts from Wikipedia articles). Wang et al. [2020] is a state-of-the-art method for this document-level RE task, utilizing BERT and graph convolutions for representation learning. Zhou et al. [2021] further enhanced this approach. None of these methods can handle input documents that are larger than a few tens of sentences. KnowFi is the first method that is geared for book-length input.

Fiction Knowledge Bases: Unterstanding characters in literary texts and constructing networks of their relationships and interactions has become a small topic in NLP (e.g., [Chaturvedi et al., 2016b, Labatut and Bost, 2019, Srivastava et al., 2016a]). The work of [Chu et al., 2019, 2020a] has advanced this theme for entity typing and type taxonomies for fictional domains. However, this work does not address learning relations between entities for KB population.

The DBkWik project [Hertling and Paulheim, 2020] has leveraged structured infoboxes of fan communities at wikia (now renamed to `fandom.com`), to construct a large KB of fictional characters and their salient properties. However, this is strictly limited to relations and respective instances that are present in infoboxes. Our work leverages wikia infoboxes for distant supervision, but our method can extract more knowledge from a variety of text sources, including storylines and synopses by fans and, most demandingly, the full text of entire books.

5.3 System Overview

The architecture of the KnowFi system is illustrated in Figure 5.1. There are two major components:

- **Distant supervision** involves pre-processing infoboxes from Wikia-hosted fan communities, to obtain seed pairs of entities. These are used to retrieve relevant passages

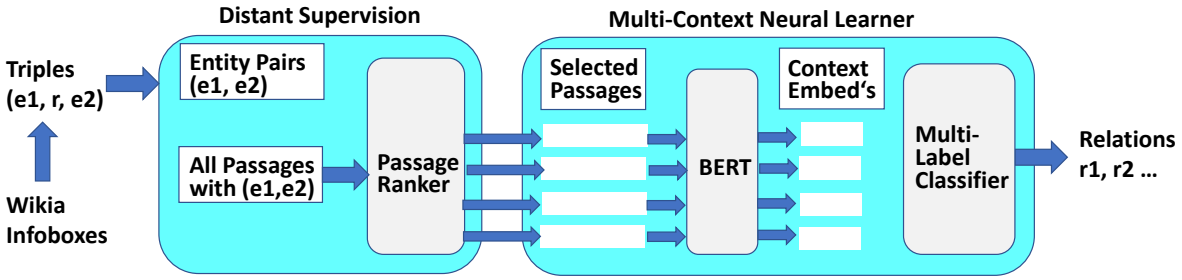


Figure 5.1: Overview of the KnowFi architecture.

Excerpt from Game of Thrones synopses at Wikia:
*Eighteen years before the War of the Five Kings, Rhaegar Targaryen allegedly abducted **Lyanna Stark** in a scandal that led to the outbreak of Robert's Rebellion. Rhaegar eventually returned to fight n the war, but not before leaving Lysanna behind at the Tower of Joy, guarded by Lord Commander Gerold Hightower and Ser Arthur Dayne of the Kingsguard. **Eddard Stark** rode to war along her betrothed, Robert Baratheon, to rescue **his sister** and avenge the deaths of their father and brother at the orders of Aerys II, the Mad King.*

Excerpt from Harry Potter book:
***Harry** had been a year old the night that **Voldemort** - the most powerful **Dark** wizard for a century, a wizard who had been gaining power steadily for eleven years, arrived at his house and **killed his father and mother**. **Voldemort** had then **turned his wand on Harry**; he had performed the **curse** that had **disposed** of many full-grown witches and wizards in his steady rise to power and, incredibly, it had not worked. Instead of **killing** the small boy, the curse had rebounded upon **Voldemort**. **Harry** had **survived** with nothing but a lightning-shaped cut on his forehead, and Voldemort had been **reduced** to something **barely alive**.*

Figure 5.2: Examples of input texts.

from the underlying text corpora: either synopses of storylines in Wikia or full-fledged content of original books. As the number of passages per entity pair can be very large in books, we devise a judicious ranking of passages and feed only the top-k passages into the next stage of training the neural network. Details are in Section 5.4.

- **Multi-context neural learning** feeds the top-k passages, with entity markup, jointly into a BERT-based encoder [Devlin et al., 2019b]. On top of this representation learning, a multi-label classifier predicts the relations that hold for the input entity pair. Details are in Section 5.5.

Note that a passage can vary from a single sentence to a long paragraph. The two seed entities would ideally occur in the same sentence, but there are many cases where they are one or two sentences apart. Figure 5.2 shows example texts from a GoT synopses in Wikia and from one of the original books.

The pre-processing of Wikia infoboxes resulted in 2.37M SPO triples for ca. 8,000 different relation names between a total of 461.4k entities, obtained from 142 domains (movie/TV series, games etc.). This forms our background knowledge for distant super-

vision. For obtaining matching passages, we focused on the 64 most frequent relations, including friend, ally, enemy and family relationships. Note that this stage is not domain-specific. Later we apply the learned model to specific domains such as GoT or Marvel Comics.

5.4 Distant Supervision with Passage Ranking

The KnowFi approach to distant supervision differs from prior works in two ways:

- **Passage ranking:** Identifying the best passages that contain seed triples, by judicious ranking, and using only the top-k passages as positive training samples.
- **Passages with gaps:** Including passages where the entities of a seed triple merely occur in separate sentences with other sentences in between.

Passage ranking: Seed pairs of entities are matched by many sentences or passages in the input corpora. For example, the pair (Herminone, Harry) appears in 1539 sentences in the the seven volumes of the Harry Potter series together. Many of these contain cues that they stand in the `friends` relation, but there are also many sentences where the co-occurrence is merely accidental. This is a standard dilemma in distant supervision for multi-instance learning [Li et al., 2020b, Riedel et al., 2010]. Our approach is to identify the best passages among the numerous matches, by judicious ranking on a per-relation basis.

For each relation, we build a *prototype representation* by selecting sentences that contain lexical matches of all three SPO arguments, where the predicate is matched by its label in the background knowledge or a short list of synonyms and close hyponyms or hypernyms (e.g., "allegiance" or "loyalty" matching `ally`). Newly seen passages for entity pairs can then be scored against the per-relation prototypes by casting both into tf-idf-weighted bag-of-word models (or alternatively, word2vec-style embeddings) and computing their cosine distance. This way, we rank candidate passage for each seed pair and target relation.

Passages with gaps: Unlike encyclopedic articles, long texts on fictional domains have a narrative style where single sentences are unlikely to give the full information in the most compact way. Therefore, we consider multi-sentence contexts where entity mentions across different sentences. In addition to simple paragraphs, we consider passages with gaps where we include sentences that are not necessarily contiguous but leave out uninformative sentences. This way, we maximize the value of limited-size text spans fed into the neural learner. This is in contrast to earlier techniques that consume whole

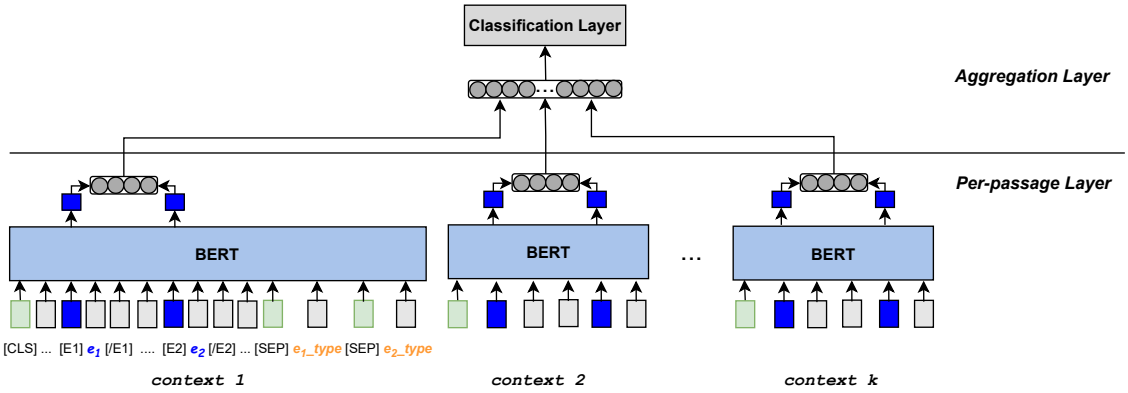


Figure 5.3: Neural network architecture for multi-context RE.

paragraphs and rely on attention mechanism for giving higher weight to informative parts.

KnowFi has two configuration parameters: the maximum number of sentences allowed between sentences that contain seed entities, and the number of sentences directly preceding or following the occurrence of a seed entity. In our experiments, we include text where the two entities appear at most 2 sentences apart and 1 preceding and 1 following sentence for each of the entity mentions, up to 512 tokens which is the current limit of BERT-based networks.

Negative training samples: In addition to the positive training samples by the above procedure, we generate negative samples by the following random process. For each relation r, we pick random entities $e1$ and $e2$ for each of the S and O roles such that there are other entities x and y for which the background knowledge asserts (e1, r, x) and (y, r, e2) with $x \neq e2$ and $y \neq e1$. This improves on the standard technique of simply choosing any pair $e1, e2$ for which $(e1, r, e2$ does not hold, by selecting more difficult cases and thus strengthening the learner. For example, both Herminone and Malfoy have some friends, but they are not friends of each other. The training of KnowFi uses a 1:1 ratio of positive to negative samples.

5.5 Multi-Context Neural Extraction

KnowFi is trained with and applicable to multiple passages as input to an end-to-end Transformer-based network with full backpropagation of cross-entropy loss. Our neural architecture has two specific components: a per-passage layer to learn BERT-based representations for each passage, and an aggregation layer that combines the signals from all input passages. In the experiments in this paper, the aggregation layer is configured

to concatenate the representations of all passages, but other options are feasible, too.

Each input passage is encoded with markup of entity mentions. In addition, we determine semantic types for the entities, using the SpaCy tool (https://spacy.io/) that provides one type for each mention, chosen from a set of 18 coarse-grained types (person, nationality/religion, event, etc.). The type of each entity mention in a passage is appended to the input vector. Figure 5.3 illustrates the neural network for multi-context RE.

5.6 LoFiDo Benchmark

To evaluate RE from long documents, we introduce the LoFiDo corpus (Long Fiction Documents). We compile SPO triples from infoboxes of 142 Wikia fan communities. After cleaning extractions and clustering synonyms, we obtain a total of 64 relations such as `enemy`, `friend`, `ally`, `religion`, `weapon`, `ruler-of`, etc.

For evaluating KnowFi and various baselines, we focus on 5 especially rich and diverse domains (i) Lord of the Rings (a series of three epic novels by J.R.R Tolkien), (ii) A Song of Ice and Fire (a series of five fantasy novels by George R.R. Martin, well-known for the Game of Thrones TV series based on it), (iii) Harry Potter (a series of seven books, written by J.K Rowling), (iv) Dune (a science-fiction novel by Frank Herbert), and (v) War and Peace (a classic novel by Leo Tolstoy). For the first four, Wikia infoboxes provide ground truth; for War and Peace, we manually crafted a small ground-truth KB. 20% of the triples from each of these universes are withheld for testing.

For the first four domains, we consider both original novels as well as narrative synopses from Wikia as input sources. War and Peace is not covered by Wikia.

LoFiDo Statistics Our LoFiDo corpus contains 81,025 instances for training and 20,257 instances for validation. For testing, we use five specific universes, which take input from both books and Wikia texts. The total number of instances in the test data from Wikia texts is 14,610, while in the case of books, it is 64,120. Ground-truth data for five test universes are provided for evaluation. Table 5.1 shows statistics on the training and validation data, while Table 5.2 shows statistics on the ground-truth of five domains in the test data. Further details on this dataset are in 5.7.4 and Appendix A. Code and data of KnowFi are available at https://www.mpi-inf.mpg.de/yago-naga/knowfi.

Dataset	# Instances	# Rel.	# Pos. Inst.	# Neg. Inst.	avg. # Pos. Inst./Rel.	avg. # Pas./Inst.
Train	81,025	64	40,920	40,105	640	1.5
Dev	20,257	64	10,363	9,894	162	1.5

Table 5.1: Statistics on training and validation set. (Rel.: relation, Inst.: instances, Pos.: positive instances, Neg.: negative instances, avg. #Pos.Inst./Rel.: average number of positive instance per relation, avg. #Pas./Inst.: average number of passages per instance)

Universe	**# rel.**	**# facts**	**top relations**
Lord of the Rings	13	1,143	race, culture, realm, weapon
Game of Thrones	18	2,547	ally, culture, title, religion
Harry Potter	20	4,706	race, ally, house, owner
Dune	11	133	homeworld, ruler, commander
War and Peace	10	101	relative, child, spouse, sibling

Table 5.2: Statistics on test data of the five test universes.

5.7 Experiments

5.7.1 Setup

Baselines We compare KnowFi to three state-of-the-art baselines on RE:

- **BERT-Type** [Shi and Lin, 2019] which uses BERT-based encodings augmented with entity type information, also based on SpaCy output in our experiments for fair comparison.
- **BERT-EM** [Soares et al., 2019] which include entity markers in input sequences;
- **GLRE** [Wang et al., 2020] which additionally computes global entity representations and uses them to augment the text sequence encodings.

The first two baselines run on a per-sentence basis, whereas GLRE is a state-of-the art method for extractions from short documents, which we train on paragraph-level inputs. The inputs for these models (i.e. sentences or paragraphs) are randomly selected.

KnowFi Parameters For context selection, we rely on TF-IDF-based bag-of-words similarity, chosing the top-100 tokens per relation as its context. For selecting passages as multi-context input, we compute the cosine between tf-idf-based vectors of each passage against the relation-specific prototype vector; we select all passages with cosine above 0.5 as positive training samples. For the neural network, we use BERT_{LARGE} (https://huggingface.co/transformers/model_doc/bert.html) with 24 layers, 1024 hidden size and 16 heads. The learning rate is $5e-5$ with Adam, the batch size is 8, and the number of training epochs is 10.

Models	Books			Wikia Texts		
	Precision	Recall	F1-score	Precision	Recall	F1-score
BERT-Type (Shi and Lin)	0.00	0.07	0.00	0.02	0.05	0.00
BERT-EM (Soares et al.)	0.06	0.11	0.08	0.11	0.20	0.14
GLRE (Wang et al.)	**0.17**	0.03	0.05	**0.18**	0.07	0.10
KnowFi	0.14	**0.11**	**0.12**	0.17	**0.26**	**0.21**

Table 5.3: Automated evaluation: average precision, recall and F1 scores.

Models	Books				Wikia Texts			
	HIT@1	HIT@3	HIT@5	MRR	HIT@1	HIT@3	HIT@5	MRR
BERT-Type (Shi and Lin)	0.01	0.02	0.04	0.02	0.09	0.20	0.23	0.16
BERT-EM (Soares et al.)	0.24	0.35	0.37	0.35	0.49	0.59	0.61	0.54
GLRE (Wang et al.)	0.40	0.53	0.54	0.46	0.47	0.62	0.68	0.57
KnowFi	**0.45**	**0.54**	**0.55**	**0.50**	**0.60**	**0.71**	**0.72**	**0.66**

Table 5.4: Automated evaluation: average *HIT@K* and *MRR* scores.

Evaluation Metrics The evaluation uses standard metrics like precision, recall and F1, averaged over all extracted triples. We report micro-averaged numbers for all relations together, and drill down on selected relations of interest. In addition, we report numbers for HITS@k and MRR. As ground-truth, we perform two different modes of evaluation:

- **Automated evaluation** is based on ground-truth from Wikia infoboxes. This is demanding on precision, but penalizes recall because of its limited coverage.
- **Manual evaluation** is based on obtaining assessments of extracted triples via crowdsourcing. This way, we include correct triples that are not in Wikia infoboxes, and thus achieve higher recall.

5.7.2 Results

Automated Evaluation Table 5.3 shows average precision, recall and F1 score. We can see that sentence-level baselines achieve comparatively high coverage, due to considering every sentence. Yet their precision is extremely low. GLRE and KnowFi achieve much higher precision, though GLRE fails to achieve competitive recall, presumably because its training on all paragraphs lowers its predictive power. As an illustration, GLRE produces only 173 assertions from all Harry Potter books, while KnowFi produces 600.

We also observe that for all methods, extraction from books is considerably harder than from the more concise synopses in Wikia.

In addition to the P/R/F1 scores, in Table 5.4 we also take an entity-centric view and evaluate how well correct extractions rank. The HITs@k metric reports how often a correct result appears among the top extractions per entity-relation pair (e.g., among

Models	Books					Wikia Texts				
	LoTR	GOT	HP	WP	**Avg.**	LoTR	GOT	HP	WP	**Avg.**
BERT-Type (Shi and Lin)	0.01	0.54	0.09	0.11	0.19	0.09	0.12	0.15	0.19	0.14
BERT-EM (Soares et al.)	0.45	0.66	0.37	0.29	0.44	0.70	0.78	0.48	0.50	0.62
GLRE (Wang et al.)	0.27	0.25	**0.56**	0.47	0.39	0.37	0.56	**0.71**	0.56	0.55
KnowFi	**0.45**	**0.76**	0.55	**0.50**	**0.57**	**0.71**	**0.83**	0.71	**0.67**	**0.73**

Table 5.5: Manual evaluation - average precision scores over 4 input texts (*LoTR:* Lord of the Rings, *GOT:* Game of Thrones, *HP:* Harry Potter, *WP:* War and Peace).

Sources	friend (top k objects)			enemy (top k objects)			ally (top k objects)		
	1	**3**	**5**	**1**	**3**	**5**	**1**	**3**	**5**
Books	0.78	**0.82**	0.80	**0.55**	0.45	0.47	0.63	**0.67**	0.63
Wikia Texts	0.73	**0.76**	0.75	**0.60**	0.48	0.49	**0.70**	0.67	0.62

Table 5.6: Manual evaluation - precision of *friend*, *enemy* and *ally* relations.

top-5 extracted enemies of Harry Potter), while MRR reports the mean reciprocal rank of the first extraction. We can observe that KnowFi outperforms all baselines on both metrics.

Manual Evaluation The low absolute scores in the above evaluation largely stem from incomplete automated ground truth. We therefore conducted an additional manual evaluation. For each domain, we select top 100 extractions from the results and used crowdsourcing to manually label their correctness. The annotators were Amazon master workers with all time approval rate $> 90\%$, and additional test questions were used to filter responses. We observed high inter-annotator agreement, on average of 0.81.

Table 5.5 shows results of our manual evaluation on four domains (Dune was left out due to complexity). As one can see, KnowFi outperforms the baselines on most input texts, and achieves a remarkable precision on both books and wikia texts (average of 0.57 on books and 0.73 on wikia texts).

We repeat the entity-centric evaluation with manual labels for three relations of special interest in fiction, *friend*, *enemy* and *ally*. We select 10 popular entities each from LoTR, GoT and Harry Potter. The resulting precision scores are shown in Table 5.6. As one can see, KnowFi is achieves high precision among its top extractions, e.g., 78% and 73% precision at rank 1 for *friend* assertions from books/Wikia texts.

Evaluation on Short-Text Datasets To evaluate the robustness of KnowFi, we also evaluate its performance on the existing sentence-level RE dataset TACRED, and the short document-level RE dataset DocRED. The results are shown in Tbl. 5.7. We find KnowFi's performance on TACRED is on par with BERT-Type and BERT-EM (0.66

Models	TACRED		DocRED	
	F1 - Dev	**F1 - Test**	**F1 - Dev**	**F1 - Test**
BERT-Type (Shi and Lin)	0.65	0.64	-	-
BERT-EM (Soares et al.)	0.64	0.62	-	-
GLRE (Wang et al.)	-	-	-	**0.57**
KnowFi	**0.67**	**0.66**	0.52	0.51

Table 5.7: Automated evaluation - short text datasets TACRED and DocRED.

test-F1, versus 0.63 and 0.62 for the baselines), the modest gain indicating that the combination of entity types and markers is beneficial. On DocRED, KnowFi achieved 0.51 F1-score, slightly below the GLRE model at 0.57 F1-score. We hypothesize that the modest losses stem from the fact that GLRE is specifically tailored for the short documents of TACRED, where multi-context aggregation is not relevant. At the same time, the single contexts GLRE considers have no inherent size limitation, unlike the 2-sentence distance threshold used in KnowFi.

Ablation Study To evaluate the impact of passage ranking, we ran KnowFi *without passage ranking* for both training and prediction. Instead, passages were randomly selected. In automated evaluation, without passage ranking, KnowFi achieves comparable recall but lower precision: 0.07 vs. 0.14 on books and 0.12 vs. 0.17 on Wikia texts. This pattern is also observed in manual evaluation, where KnowFi, without passage ranking, achieves a precision of 0.43 vs. 0.57 on books and 0.55 vs. 0.73 on Wikia texts.

Further experiments can be found at Appendix B.

Error Analysis The precision gain from automated to manual evaluation (Table 5.3 vs. Table 5.5) indicates that ground-truth incompleteness is a confounding factor. We further investigated this by inspecting a sample of 50 false positives. We found that 20% originated from incomplete ground truth, while 54% were indeed not inferrable from the given contexts (e.g., extracting friendship from the sentence *"Thorin came to Bilbo's door"*). Another 15% were errors in determining the subject or object in complex sentences with many entity mentions. Finally, 7% of the false positives captured semantically related relations but missed the correct ones.

By sampling false negatives, we found that in 52% of the cases the retrieved contexts did not allow the proper inference, indicating limitations in the context retrieval and ranking. In 33% of the cases, a human reader could spot the relation in the top-ranked contexts (e.g., `hasCulture (Legolas, Elf)` in *"He saw Legolas seated with three other*

Source	Relation	Context(s)	BERT-EM	GLRE	KnowFi	GT
Books	*enemy*	C1: So to gain time Gollum challenged Bilbo to the Riddle-game, saying that if he asked a riddle which Bilbo could not guess, then he would kill him and eat him. C2: There Gollum crouched at bay, smelling and listening; and Bilbo was tempted to slay him with his sword.	✓	✗	✓	-
	weapon	C1: They watched him rejoin the rest of the Slytherin team, who put their heads together, no doubt asking Malfoy whether Harry's broom really was a Firebolt. C2: Faking a look of sudden concentration, Harry pulled his Firebolt around and sped off toward the Slytherin end. C3: Harry was prepared to bet everything he owned, including his Firebolt, that it wasn't good news...	✓	✗	✓	-
	ally	C1:...Lord Blackwood shall be required to confess his treason and abjure his allegiance to the Starks ... C2:..."I swore an oath to Lady Stark, never again to take up arms against the Starks", said Blackwood ...	✗	✗	✓	✓
	founder	There was a great roar and a surge toward the foot of the stairs; he was pressed back against the wall as they ran past him, the mingled members of the Order of the Phoenix, Dumbledore's Army, and Harry's old Quidditch team, all with their wands drawn, heading up into the main castle.	✗	✓	✓	✓
Wikia Texts	*friend*	Mulciber was also a friend of Severus Snape, which upset Lily Evans, who was Snape's best friend at the time.	✓	✗	✓	-
	spouse	...Later, after sweets and nuts and cheese had been served and cleared away, Margaery and Tommen began the dancing, looking more than a bit ridiculous as they whirled about the floor. The Tyrell girl stood a good foot and a half taller than her little husband, and Tommen was a clumsy dancer at best ...	✗	✓	✓	✓
	weapon	Randyll repeatedly berates Sam: he insults his weight, tells him the Night's Watch failed to make a man out of him, and says he will never be a great warrior , or inherit Heartsbane, the Tarly family's ancestral Valyrian steel sword.	✓	✗	✓	✓
	culture	C1: The most powerful Ainu, Melkor (later called Morgoth or "Dark Enemy" by the elves), Tolkien's equivalent of, disrupted the theme, and in response, Eru Ilúvatar introduced new themes that enhanced the music beyond the comprehension of the Ainur. C2: Melkor's brother was Manwë, although Melkor was greater in power and knowledge than any of the Ainur.	✓	✓	✓	✓

Table 5.8: Anecdotal examples for the outputs of KnowFi (**GT**: ground-truth, subject in red, object in blue).

Elves").

5.7.3 Anecdotal Examples

Table 5.8 gives examples for the output of the various methods on sample contexts. The red color texts denote subjects and the blue color texts denote objects.

5.7.4 Background KB Statistics

One of our contribution is the background KB dataset on popular universes in fictional domains. To have an overview about the dataset, Table 5.9 shows some statistics on our background KBs database, which include information about universes, entities, type systems, relations and facts.

From the 5 domains used for testing, the number of relations varies from 13 to 21, and the number of ground-truth triples varies from 1,100 to 4,600 for the first three domains, and was between 100 and 200 the last two.

Statistics		**Top Universes**		**Top Relations**		
# Universes	142	**Universes**	**# Facts**	**Relations**	**#**	**# universes**
	per universe	Star Wars	282,440	name	238,290	111
# Facts	13,539	Monster Hunter	153,178	type	112,347	94
# Relations	163	World of Warcraft	144,586	gender	95,972	77
# Entities	158,066	Marvel	77,826	affiliation	85,676	61
# Entity Mentions	224,782	DC Comics	69,190	era	53,871	12
# Entity Types	1246	Forgotten Realms	63,360	hair(color)	50,325	41

Table 5.9: Statistics on background KBs.

In **Lord of the Rings**, which summary is more informative for `Frodo Baggins`:	
Summary 1:	<Frodo, has parent, Drogo>, <Frodo, has culture, Shire>, <Frodo, has enemy, Sauron>, <Frodo, has friend, Sam>, <Frodo, has weapon, Sting>
Summary 2:	<Frodo, has owner, Gandalf>, <Frodo, has weapon, Ring>, <Frodo, has parent, Drogo>, <Frodo, has affiliation, Sam>, <Frodo, has culture, Marish>

Table 5.10: Sample task for assessing entity summaries.

5.8 Extrinsic Use Case: Entity Summarization

To assess the salience in the extractions produced by KnowFi, we pursued a user study to compare entity summaries, one by KnowFi and one by a baseline. Each entity summary includes at most 5 best extractions (distinct relations) from the book series Lord of the Rings, Game of Thrones and Harry Potter. For each domain, we generate summaries for 5 popular entities. We give pairs of summaries, with randomized order, to Amazon master workers for selecting the more informative one. Table 5.10 shows an example of this crowdsourcing task. We compare KnowFi to all baselines. The annotators preferred KnowFi-based summaries over BERT-Type, BERT-EM and GLRE in 93%, 64% and 81% of the cases, respectively.

5.9 Summary

To the best of our knowledge, this work is the first attempt at relation extraction (RE) from long fictional texts, such as entire books. The presented method, KnowFi, is specifically geared for this task by its judicious selection and ranking of passages. KnowFi outperforms strong baselines on RE by a substantial margin, and it performs competitively even on the short-text benchmarks TacRed and DocRed. The absolute numbers for precision and recall show that there is still a lot of room for improvement. This underlines our hypothesis that long fictional texts are a great challenge for RE. Our LoFiDo corpus of Wikia texts, book contents, and ground-truth labels will be made available to foster further research. All information can be found at https://www.mpi-inf.mpg.de/yago-naga/knowfi.

Chapter 6

Conclusions

6.1 Contributions

This dissertation is about information extraction and knowledge acquisition. We specifically addressed the long-tail domain of fiction and fantasy – core parts of our human culture.

The first contribution, TiFi, is a method for taxonomy induction for fictional domains. TiFi uses noisy category systems from fan wikis or text extraction as input and builds the taxonomies through three main steps: (i) category cleaning, by identifying candidate categories that truly represent classes in the domain of interest, (ii) edge cleaning, by selecting subcategory relationships that correspond to class subsumption, and (iii) top-level construction, by mapping classes onto a subset of high-level WordNet categories. TiFi is able to construct taxonomies for a diverse range of fictional domains such as Lord of the Rings, The Simpsons or Greek Mythology with very high precision and it outperforms state-of-the-art baselines for taxonomy induction by a substantial margin (82% vs. 89% F1-scores).

The second contribution, ENTYFI, is a method for typing entities in fictional texts coming from books, fan communities, or amateur writers. ENTYFI builds on 205 automatically induced high-quality type systems for popular fictional domains, and exploits the overlap and reuse of these fictional domains for fine-grained typing in previously unseen texts. ENTYFI comprises five steps: type system induction, domain relatedness ranking, mention detection, mention typing, and type consolidation. The typing module combines a supervised neural model, unsupervised Hearst-style and dependency patterns, and knowledge base lookups. The consolidation stage utilizes co-occurrence statistics in order to remove noise and to identify the most relevant types. Extensive experiments on newly seen fictional texts demonstrate the quality of ENTYFI over the state of the arts on entity typing (43% vs. 49% F1-scores)

The third contribution, KnowFi, is an end-to-end model for extracting relations between entities coming from very long texts such as books, novels, or fan fan-built wikis. KnowFi leverages semi-structured content in wikis of fan communities on fandom.com (aka wikia.com) to extract an initial KB of background knowledge for 142 popular domains (TV series, movies, games). This serves to identify interesting relations and to collect distant supervision samples. Yet for many relations this results in very few samples. To overcome this sparseness challenge and to generalize the training across the wide variety of relations, a similarity-based ranking technique is devised for matching seeds in text passages. Given a long input text, KnowFi judiciously selects a number of context passages containing seed pairs of entities. To infer if a certain relation holds between two entities, KnowFi's neural network is trained jointly for all relations as a multi-label classifier. Experiments with several fictional domains demonstrate the gains that KnowFi achieves over the best prior methods for neural relation extraction (44% vs. 57% average precision).

Along with the publications, code and data are also published and available at https://www.mpi-inf.mpg.de/yago-naga/fiction-fantasy to accelerate further research in fictional domains. Approaches to fictional domains also have potential for being carried over to real-life settings, such as enterprise-specific domains, medieval history, neurodegenerative diseases, or nanotechnology material science.

6.2 Discussion and Future Work

With potentially considerable impact on downstream applications, the task of constructing KBs has received a lot of attention. However, developing an end-to-end model for KB construction is not straightforward.

Schema-free KBs do not follow any ontology; therefore, models to construct these KBs mostly use open information extraction techniques, which are flexible and able to produce a large number of extractions [Etzioni et al., 2011, Mausam, 2016]. However, neither entities nor relations are canonicalized; these methods then face issues regarding the quality and informativeness of their extractions. In the end, downstream tasks based on these KBs still need to deal with named entity disambiguation and relational paraphrases [Nguyen et al., 2017a]. On the other hand, schema-based KBs, such as encyclopedic KBs, require the model to pre-define the ontology which includes the entity type system and relations between entities. To construct these KBs, it is essential to address a wide range of tasks such as taxonomy induction, named entity recognition and disambiguation, and

relation extraction, where the output of one task can affect to other tasks. For example, the performance of relation extraction could be affected by the performance of named entity recognition and disambiguation. Along with tackling on each single task, people also try to develop end-to-end systems for constructing these KBs. Notable systems are Deepdive [Shin et al., 2015], SystemT [Krishnamurthy et al., 2009, Li et al., 2011], NELL [Mitchell et al., 2018] and QKBfly [Nguyen et al., 2017a]. Although these systems are called end-to-end, they still require specifications of relations or prior knowledge about the input domain, or human intervention to improve the extracted results. These systems are not suitable for cold-start KB construction, especially on a new domain, where prior knowledge about the domain is not available. Due to the unscalability and expert knowledge required, even large KBs which are used in commercial search engines like Google and Bing still lack knowledge about specific domains. Moreover, running on a huge data collection, a trade-off between efficiency and effectiveness is also a challenge of these methods.

Therefore, working on knowledge extraction in fictional domains, it is reasonable to deal with each single task. While a number of key challenges have been addressed throughout this dissertation, there are still some drawbacks and other directions which can be explored in the future.

An end-to-end model to build a large-scale taxonomy for all fictional domains: TiFi presents a pipeline for constructing taxonomies for fictional domains, which includes three steps: noisy category cleaning, edge cleaning, and WordNet linking. Although it is more convenient to control the quality of the output when working on each step separately, such multi-step methods may not effectively optimize features between steps. It also takes more effort to design features for each step. An end-to-end model for this task can resolve these problems. For example, Mao et al. [2018] propose an end-to-end reinforcement learning for automatic taxonomy induction. Alternatively, a neural model with two prediction heads, node cleaning, and edge cleaning, may be able to combine the two first steps of TiFi.

In terms of the output, TiFi returns a taxonomy for each fictional domain. A taxonomy for all fictional domains might be interesting and useful for later tasks, such as entity typing on any given input text without understanding which domain it belongs to. A such taxonomy is also similar to the one in the real-world domain (e.g. Wikipedia category network). Although the experiments show that TiFi is able to learn across domains, building this taxonomy requires further effort, including the consolidation of types between domains (e.g. *dragons* in Western novels and Chinese novels).

A more efficient model for entity typing in long fictional texts: ENTYFI presents a comprehensive method for entity typing which not only exploits taxonomies in fictional domains, but also leverages different approaches, such as pattern-based, unsupervised, supervised, and KB lookup. Although this method is able to achieve high recall, it requires the system to execute many components, which is computationally expensive, especially when deploying the system online. Moreover, to be able to run ENTYFI, many external resources are required, such as taxonomies and background KBs from over 200 fictional domains, and constraints for type consolidation like disjointness or correlations. Several directions might help to improve the efficiency of ENTYFI. For example, using a taxonomy for all fictional domains (if available) could avoid the ranking step and reduce external resources. An end-to-end neural model with a graph embedding layer to learn representations of entity mentions across long texts and a classification layer to detect types of them might avoid the type consolidation step. With the outstanding performance of BERT on different NLP tasks recently, a BERT-based model is also worth to be explored.

KB enrichment from KnowFi output: KnowFi addresses relation extraction in long fictional texts. To construct KBs from KnowFi output, further tasks need to be investigated. First, it is necessary to consolidate extracted triples. A fictional story includes many characters and other entities, which appear in different contexts across the story. An entity, hence, may have different relations to another entity that is conflicting. The relation between different entity pairs might be also incompatible. Consolidation is able to reduce these incompatibilities in the predictions and improve precision. During developing KnowFi, we tried to solve the consolidation problem by transferring extracted assertions into a weighted MaxSAT model. However, the results showed that the proposed method did not significantly contribute to the final result. In the end, the consolidation step was removed from KnowFi. This issue is left for future work.

Second, it is necessary to link entities from extracted triples to existing KBs. Entity linking is an important task in KB enrichment. For example, a triple <`Mr. Potter`, *hasFriend*, `Hermione`>, extracted from Harry Potter books, should be normalized as <`Harry_Potter`, *hasFriend*, `Hermione_Granger`> and linked to the knowledge base of *Harry Potter* universe. KnowFi constructs background KBs for over 140 popular fictional domains by extracting semi-structured texts from Wikia (e.g. category networks, infoboxes). These KBs can be enriched by updating new facts produced by KnowFi from unstructured texts such as books.

Finally, there are several options that can be exploited to improve KnowFi results.

For example, a BERT-based retrieval model could be used for passage ranking. Type information from ENTYFI could be used in the relation extraction neural model. Or coreference resolution could be used to improve the recall.

Connecting the dots: TiFi, ENTYFI, and KnowFi address three main problems in knowledge extraction and can be combined into a complete pipeline. Although ENTYFI takes the output from TiFi as the targets for entity typing tasks, KnowFi, however, has not used any information from ENTYFI or TiFi for relation extraction. Conceptually, type information can be used to improve the relation extraction and entity disambiguation/linking task. Due to the computational cost of ENTYFI, KnowFi currently uses an external library (e.g. SpaCy) to extract type information for entities. Since these types are coarse-grained and suitable for the real-world domain, it is promising to use entity types produced by ENTYFI for the relation extraction model in KnowFi.

Era of large scale pre-trained language models: Nowadays, language models play a big role in most NLP tasks. Language models are trained by predicting a target word from a given context or context words from a target word and also learning latent representations of words. These embeddings are then used as the input of NLP models. Recently, many large-scale pre-trained language models, such as BERT and GPT-3, have been introduced and significantly improve the performance of NLP tasks. BERT and GPT-3 are built as Transformer architectures to encode huge text corpora. Based on the original objective of language models which is predicting the missing words, it is possible to use these language models to extract knowledge directly [Jiang et al., 2020, Petroni et al., 2019]. For example, we could ask GPT-3 to complete an input sentence: *"Joe Biden is president of ..."*. By filling the blank, the answer given by GPT-3 might become a candidate for the object of the triple <`Joe_Biden`, *presidentOf*, [object]>, here is `the United States` with a highest confident score of 0.76.

However, with a given query *"In Game of Thrones, Jon Snow is the child of ..."*, GPT-2 returns some predictions such as "his mother and mother" and "the Dothraki"; or with the query *"In Harry Potter series, Harry is the [MASK] of Hermione"*, the top answers from BERT are "father", "son" and "husband", which are all incorrect. Although the performance of language models on long-tail domains such as fiction is currently far from satisfactory, extracting knowledge using language models is promising, especially when the text corpora for training these models keeps increasing, day by day.

Appendix A

KnowFi – Training Data Extraction

Many current KBs like Yago, DBpedia or Freebase have been built by extracting the information from infoboxes, category network and leveraging the markup language of Wikipedia. The relations of these KBs are then used as schema for many later supervised relation extractors. However, for fiction, Wikipedia has too low coverage of entities and relevant relations.

Wikia Wikia (or Fandom) is the largest web platform for fiction and fantasy. It contains over 385k communities with total of over 50M pages. Each community (usually discusses about one fictional universe) is organized as a single Wiki. With a wide range of coverage on fiction and fantasy, Wikia is one of the 10 most visited websites in the US (as of 2020)[1].

Crawling We download all universes which contain over 1000 content pages and have available dump files from Wikia, and get total of 142 universes in the end. From these universes, we extract all information from their category networks and infoboxes, and build a background knowledge base for each universe.

Definition A.0.1. ***Background KB*** *of an universe is a collection of entities, entity mentions, simple facts that describe relations between entities and a type system of the universe.*

Background knowledge extraction To extract the background KBs, we follow a simple procedure:

- **Type system construction:** The type system is extracted from Wikia category network. We adapt the technique from the TiFi system [Chu et al., 2019] to structure and clean the type system.
- **Entity detection:** Entities and entity mentions can be easily extracted from the dump file. We consider page titles as the entities in the universe (except administration and category pages). On the other hand, entity mentions only

[1] https://ahrefs.com/blog/most-visited-websites/

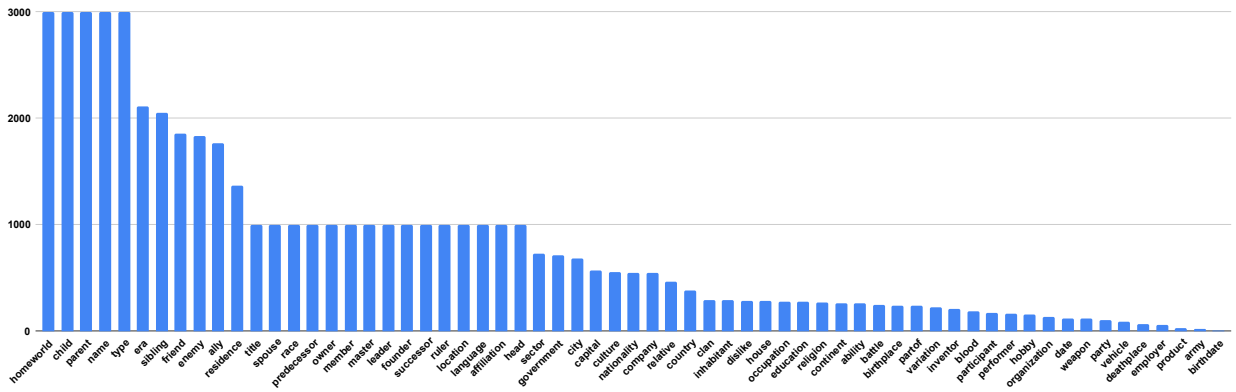


Figure A.1: Statistics on training data.

appear in texts. By using Wiki markup, each mention can be extracted and linked to the entity with a confident score which is computed based on its frequency.

- **Infobox extraction:** Facts about each entity are extracted from its infobox. Infobox is presented in table format with the entity's attributes and their values. Each extracted fact is presented in a triple with subject, predicate (relations) and object. In particular, we consider the main entity as subject, the attributes as predicates, and the values as objects. We manually check if there is any misspelling in the relations and merge them if necessary.

This results an average of 158k entities and 13.5k facts in each universe. The information from these background KBs is then used for all three later steps.

Relation Filtering After extracting the background KBs, we get all relations from the facts of all universes and consider them as relation candidates that can be extracted in fictional domains. However, beside meta relations which are not really related to the content of universes, such as *season, page, episode,...*, there is much noise in the relations since they are manually created by fans. To remove noise and keep popular relations, we do **relation filtering** as follows:

- Pre-processing: a combination of stemming and keeping relations with length at least 3 (except for some relations like *job, age, son, etc.).*
- Infrequent-relation removing: we only keep relations which are in at least 5 universes and appear in over 20 facts.
- Meta-relation removing: we manually check if the relation is a meta-relation. In total, there are 247 relations considered as meta-relations.

- Misspelling detection: Misspelling relations are manually detected and grouped with the correct relations, for example, *affilation* and *affiliation*.
- Grouping: Synonym relations are manually grouped together, for example, *leader* and *commander*.

After relation filtering, we reduce the number of relations from over 8,000 to 64 relations. These relations are considered as *popular relations* in fictional domains and used as targets for the relation extraction step. We realize that in fictional domains, the relations expressing the friendly or hostile relationship between two entities are interesting, hence, we keep *friend* and *enemy* as two relations which are always extracted. Figure A.1 shows statistics on training data. We publish the training data as supplementary material.

Appendix B

KnowFi – Additional Experiments

Similarity Threshold In our experiments, we consider all passages with cosine above 0.5 as positive training samples (section 5.7.1). To assess the effect of the similarity threshold, we conduct an ablation study on it. Table B.1 reports the automated results of KnowFi on both books and Wikia texts, where the threshold varies. For the author response we completed two other runs (threshold 0.4 and 0.6) that indicate modest influence, for the final version we would provide insights for all threshold values from 0 to 1 (in 0.1 step size). The results show that, with a similarity threshold around 0.5, the model achieves the best F1. By increasing the similarity threshold, the model is able to achieve higher precision but lower recall and vice versa.

Embedding-based Passage Ranking KnowFi uses a simple TF-IDF-based schema for passage ranking. To assess the effectiveness of this method, we conduct an ablation study on the ranking step. Instead of using TF-IDF, we compute the embeddings of the passages and the relation contexts using Sentence-BERT [Reimers and Gurevych, 2019]. The cosine similarity between the passage embedding and the relation context embedding is then computed using the *sklearn* library. We select all passages with cosine above 0.0 (range [-1,1]), as positive training samples, with maximum of 5 passages per each training instance. Table B.2 shows the automated results of KnowFi on both books and Wikia texts. The higher scores on recall shows that the embeddings can help the model capture the semantic relationships between passages and relation contexts better, especially when handling the cases of synonymy, while TF-IDF only handles the cases of lexical matching. However, in general, both techniques are on par, and embeddings do not improve the results, in terms of F1-score.

Threshold	**Books**			**Wikia Texts**		
	Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
0.4	0.07	**0.11**	0.09	0.12	**0.32**	0.17
0.5	0.14	**0.11**	0.12	0.17	0.26	**0.21**
0.6	**0.20**	0.10	**0.13**	**0.21**	0.17	0.19

Table B.1: Automated Evaluation: Study on the similarity threshold.

Ranking Methods	Books			Wikia Texts		
	Precision	Recall	F1-score	Precision	Recall	F1-score
TF-IDF-based	**0.14**	**0.11**	**0.12**	**0.17**	0.26	**0.21**
Embedding-based	0.12	**0.11**	**0.12**	0.10	**0.30**	0.15

Table B.2: Automated Evaluation: Study on the ranking method.

Ranking Methods	Books			Wikia Texts		
	Precision	Recall	F1-score	Precision	Recall	F1-score
GLRE (Wang et al.)	0.17	0.03	0.05	0.18	0.07	0.10
GLRE + Passage Ranking	**0.20**	0.03	0.06	**0.21**	0.10	0.13
KnowFi	0.14	**0.11**	**0.12**	0.17	**0.26**	**0.21**

Table B.3: Automated Evaluation: GLRE with Passage Ranking.

GLRE with Passage Ranking In our experimental setup, the inputs of GLRE [Wang et al., 2020] (for both train and test) are randomly selected. To assess the effect of passage ranking on GLRE, we conduct a study where the inputs of GLRE are selected by using our method for passage ranking. Table B.3 shows that, by using passage ranking to filter the inputs, GLRE is able to achieve higher precision and recall, compared to GLRE without passage ranking. However, this enhanced variant is still inferior to KnowFi by a substantial margin.

Impact of Training Data Quality Training data is one of the most important factors that impact the quality of the supervised models, therefore, it is essential to maintain the quality of the training data, especially when working on specific domains where the training data is usually not available. To evaluate the quality of our training data, we compare KnowFi and it variant (e.g. without using passage ranking on training data collection) with two other methods which are trained using manual training datasets:

- TACRED [Zhang et al., 2017a], a popular dataset for relation extraction on the sentence level. We train our relation extraction model using TACRED and use the model to extract the relations from the test data.
- Diffbot [Mesquita et al., 2019], a commercial api for relation extractions. We run Diffbot API on our test data to extract the relations.

We automatically evaluate the extractions on three popular relations, *spouse, sibling, child*, since they are contained in all datasets.

Results Table B.4 reports the results on two universes, *Lord of the Rings* and *Game of Thrones*. The results show that, our training data achieves comparable results with

Universes	Models	Books			Wikia Texts		
		Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
LoTR	**Diffbot**	0.68	1.75	0.98	1.69	54.39	3.28
	TACRED-based	**28.57**	0.93	1.79	5.34	37.96	9.36
	KnowFi - w/o ranking	1.34	**4.38**	2.05	2.20	**79.82**	4.27
	KnowFi	15.1	2.00	**3.53**	**8.19**	27.19	**12.58**
GOT	**Diffbot**	6.10	**18.97**	9.24	7.85	**61.46**	13.92
	TACRED-based	8.45	4.61	5.96	19.66	40.11	26.39
	KnowFi - w/o ranking	8.29	15.81	10.87	9.64	47.43	16.03
	KnowFi	**11.63**	18.83	**12.64**	**19.8**	50.59	**28.47**

Table B.4: Average scores on three popular relations: *spouse, sibling, child*

other datasets and even higher F1-scores, in both books and Wikia texts.

List of Figures

List of Tables

Bibliography

Apoorv Agarwal, Jiehan Zheng, Shruti Kamath, Sriramkumar Balasubramanian, and Shirin Ann Dey. Key female characters in film have more to talk about besides men: Automating the Bechdel test. In *the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2015.

Eugene Agichtein and Luis Gravano. Snowball: extracting relations from large plain-text collections. In *Joint Conference on Digital Libraries (JCDL)*, 2000.

Daniele Alfarone and Jesse Davis. Unsupervised learning of an is-a taxonomy from a limited domain-specific corpus. In *the International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A nucleus for a web of open data. In *The Semantic Web 2007*. 2007.

David Bamman, Brendan O'Connor, and Noah A Smith. Learning latent personas of film characters. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2014.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. Open information extraction from the web. In *the International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

Mohit Bansal, David Burkett, Gerard De Melo, and Dan Klein. Structured learning for taxonomy induction with belief propagation. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2014.

Oliver Bender, Franz Josef Och, and Hermann Ney. Maximum entropy models for named entity recognition. In *Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT)*, 2003.

William J Black, Fabio Rinaldi, and David Mowatt. Facile: Description of the ne system used for muc-7. In *MUC-7*, 1998.

Olivier Bodenreider. The unified medical language system (UMLS): Integrating biomedical terminology. *Nucleic acids research*, 2004.

Sergey Brin. Extracting patterns and relations from the world wide web. In *WebDB Workshop*, 1998.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. Toward an architecture for never-ending language learning. In *the AAAI Conference on Artificial Intelligence*, 2010a.

Andrew Carlson, Justin Betteridge, Richard C Wang, Estevam R Hruschka Jr, and Tom M Mitchell. Coupled semi-supervised learning for information extraction. In *ACM International WSDM Conference*, 2010b.

Nathanael Chambers and Dan Jurafsky. Unsupervised learning of narrative schemas and their participants. In *Annual Meeting of the Association for Computational Linguistics International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, 2009.

Snigdha Chaturvedi, Shashank Srivastava, Hal Daume III, and Chris Dyer. Modeling evolving relationships between characters in literary novels. In *the AAAI Conference on Artificial Intelligence*, 2016a.

Snigdha Chaturvedi, Shashank Srivastava, Hal Daumé III, and Chris Dyer. Modeling evolving relationships between characters in literary novels. In *the AAAI Conference on Artificial Intelligence*, 2016b.

Snigdha Chaturvedi, Mohit Iyyer, and Hal Daumé III. Unsupervised learning of evolving relationships between literary characters. In *AAAI Conference on Artificial Intelligence*, 2017.

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. Ultra-fine entity typing. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.

Sreyasi Nag Chowdhury, Niket Tandon, and Gerhard Weikum. Know2look: commonsense knowledge for visual search. *the 5th Workshop on Automated Knowledge Base Construction (AKBC)*, 2019.

Philipp Christmann, Rishiraj Saha Roy, Abdalghani Abujabal, Jyotsna Singh, and Gerhard Weikum. Look before you hop: Conversational question answering over knowledge graphs using judicious context expansion. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*, 2019.

Cuong Xuan Chu, Simon Razniewski, and Gerhard Weikum. Tifi: Taxonomy induction for fictional domains. In *The Web Conference*, 2019.

Cuong Xuan Chu, Simon Razniewski, and Gerhard Weikum. Entyfi: Entity typing in fictional texts. In *ACM International WSDM Conference*, 2020a.

Cuong Xuan Chu, Simon Razniewski, and Gerhard Weikum. Entyfi: A system for fine-grained entity typing in fictional texts. In *EMNLP Demo*, 2020b.

Cuong Xuan Chu, Simon Razniewski, and Gerhard Weikum. Knowfi: Knowledge extraction from long fictional texts. In *Conference on Automated Knowledge Base Construction (AKBC)*, 2021.

Philipp Cimiano, Andreas Hotho, and Steffen Staab. Learning concept hierarchies from text corpora using formal concept analysis. *J. Artif. Intell. Res.*, 2005.

Michael Collins and Yoram Singer. Unsupervised models for named entity classification. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. In *the Proceedings of Machine Learning Research (JMLR)*, 2011.

Luciano del Corro, Abdalghani Abujabal, Rainer Gemulla, and Gerhard Weikum. Finet: Context-aware fine-grained named entity typing. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2015.

Mark Craven, Dan DiPasquo, Dayne Freitag, Andrew McCallum, Tom M. Mitchell, Kamal Nigam, and Seán Slattery. Learning to extract symbolic knowledge from the world wide web. In *the AAAI Conference on Artificial Intelligence*, 1998.

Lei Cui, Furu Wei, and Ming Zhou. Neural open information extraction. *the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.

Gerard de Melo and Gerhard Weikum. MENTA: Inducing multilingual taxonomies from Wikipedia. In *the Conference on Information and Knowledge Management (CIKM)*, 2010.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT)*, 2019a.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT)*, 2019b.

Li Dong, Furu Wei, Hong Sun, Ming Zhou, and Ke Xu. A hybrid neural model for type classification of entity mentions. In *the International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

Xin Luna Dong, Xiang He, Andrey Kan, Xian Li, Yan Liang, Jun Ma, Yifan Ethan Xu, Chenwei Zhang, Tong Zhao, Gabriel Blanco Saldana, et al. Autoknow: Self-driving knowledge collection for products of thousands of types. In *ACM SIGKDD Conference on Knowledge Discovery Data Mining*, 2020.

Xishuang Dong, Lijun Qian, Yi Guan, Lei Huang, Qiubin Yu, and Jinfeng Yang. A multiclass classification method based on deep learning for named entity recognition in electronic medical records. In *New York Scientific Data Summit*, 2016.

Markus Eberts, Kevin Pech, and Adrian Ulges. Manyent: A dataset for few-shot entity typing. In *the International Conference on Computational Linguistics (COLING)*, 2020.

Vinodh Krishnan Elangovan and Jacob Eisenstein. "You're Mr. Lebowski, I'm the Dude": inducing address term formality in signed social networks. In *the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2015.

David K Elson, Nicholas Dames, and Kathleen R McKeown. Extracting social networks from literary fiction. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2010.

Faezeh Ensan and Ebrahim Bagheri. Document retrieval model through semantic linking. In *Proceedings of the tenth ACM international conference on web search and data mining (WSDM)*, 2017.

Oren Etzioni, Michael J. Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Web-scale information extraction in knowitall (preliminary results). In *the Web Conference*, 2004.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 2005.

Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam. Open information extraction: The second generation. In *the International Joint Conference on Artificial Intelligence (IJCAI)*, 2011.

Quan Fang, Changsheng Xu, Jitao Sang, M. Shamim Hossain, and Ahmed Ghoneim. Folksonomy-based visual ontology construction and its applications. *IEEE Trans. Multimedia*, 2016.

Stefano Faralli, Alexander Panchenko, Chris Biemann, and Simone Paolo Ponzetto. The contrastmedium algorithm: Taxonomy induction from noisy knowledge graphs with just a few links. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2017.

Stefano Faralli, Irene Finocchi, Simone Paolo Ponzetto, and Paola Velardi. Webisagraph: A very large hypernymy graph from a web corpus. In *Italian Conference on Computational Linguistics*, 2019.

Ethan Fast, William McGrath, Pranav Rajpurkar, and Michael S Bernstein. Augur: Mining human behaviors from fiction to power interactive systems. In *Conference on Human Factors in Computing Systems*, 2016.

Christiane Fellbaum and George Miller. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. Building watson: An overview of the deepqa project. *AI magazine*, 2010.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2005.

Tiziano Flati, Daniele Vannella, Tommaso Pasini, and Roberto Navigli. Two is bigger (and better) than one: the Wikipedia bitaxonomy project. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2014.

Tiziano Flati, Daniele Vannella, Tommaso Pasini, and Roberto Navigli. Multiwibi: The multilingual wikipedia bitaxonomy project. *Artificial Intelligence*, 2016.

Christopher Funk, William Baumgartner, Benjamin Garcia, Christophe Roeder, Michael Bada, K Bretonnel Cohen, Lawrence E Hunter, and Karin Verspoor. Large-scale biomedical concept recognition: an evaluation of current automatic annotators and their parameters. *BMC bioinformatics*, 2014.

Yarin Gal and Zoubin Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *the Annual Conference on Neural Information Processing Systems (NIPS)*, 2016.

Chongming Gao, Wenqiang Lei, Xiangnan He, Maarten de Rijke, and Tat-Seng Chua. Advances and challenges in conversational recommender systems: A survey. *AI Open. Vol. 2*, 2021.

Matthew R Gormley, Mo Yu, and Mark Dredze. Improved relation extraction with feature-rich compositional embedding models. *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

Alex Graves. Supervised sequence labelling. In *Supervised sequence labelling with recurrent neural networks*. 2012.

Ralph Grishman and Beth M Sundheim. Message understanding conference-6: A brief history. In *the International Conference on Computational Linguistics (COLING)*, 1996.

Xiaoyu Guo, Hui Zhang, Haijun Yang, Lianyuan Xu, and Zhiwen Ye. A single attention-based combination of cnn and rnn for relation classification. *IEEE Access*, 2019.

Amit Gupta, Francesco Piccinno, Mikhail Kozhevnikov, Marius Pasca, and Daniele Pighin. Revisiting taxonomy induction over Wikipedia. In *the International Conference on Computational Linguistics (COLING) 2016*, 2016a.

Amit Gupta, Francesco Piccinno, Mikhail Kozhevnikov, Marius Pasca, and Daniele Pighin. Revisiting taxonomy induction over wikipedia. In *the International Conference on Computational Linguistics*, number EPFL-CONF-227401, 2016b.

Amit Gupta, Francesco Piccinno, Mikhail Kozhevnikov, Marius Pasca, and Daniele Pighin. Revisiting taxonomy induction over Wikipedia. In *the International Conference on Computational Linguistics (COLING)*, 2016c.

Amit Gupta, Rémi Lebret, Hamza Harkous, and Karl Aberer. Taxonomy induction using hypernym subsequences. In *the Conference on Information and Knowledge Management (CIKM)*, 2017a.

Amit Gupta, Rémi Lebret, Hamza Harkous, and Karl Aberer. Taxonomy induction using hypernym subsequences. *the 2017 ACM on Conference on Information and Knowledge Management (CIKM)*, 2017b.

Rahul Gupta, Alon Y. Halevy, Xuezhi Wang, Steven Euijong Whang, and Fei Wu. Biperpedia: An ontology for search applications. *International Conference on Very Large Data Bases (PVLDB)*, 2014.

Xu Han, Tianyu Gao, Yankai Lin, Hao Peng, Yaoliang Yang, Chaojun Xiao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. More data, more relations, more context and more openness: A review and outlook for relation extraction. *Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics International Joint Conference on Natural Language Processing (AACL)*, 2020a.

Xu Han, Tianyu Gao, Yankai Lin, Hao Peng, Yaoliang Yang, Chaojun Xiao, Zhiyuan Liu, Peng Li, Jie Zhou, and Maosong Sun. More data, more relations, more context and more openness: A review and outlook for relation extraction. In *Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics International Joint Conference on Natural Language Processing (AACL/IJCNLP)*, 2020b.

Daniel Hanisch, Katrin Fundel, Heinz-Theodor Mevissen, Ralf Zimmer, and Juliane Fluck. Prominer: rule-based protein and gene entity recognition. *BMC bioinformatics*, 2005.

Zellig S Harris. Distributional structure. *Word*, 1954.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. Deep semantic role labeling: What works and what's next. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017.

Marti A Hearst. Automatic acquisition of hyponyms from large text corpora. In *the International Conference on Computational Linguistics (COLING)*, 1992.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *the Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.

Sven Hertling and Heiko Paulheim. Webisalod: providing hypernymy relations extracted from the web as linked open data. In *International Semantic Web Conference (ISWC)*, 2017.

Sven Hertling and Heiko Paulheim. Dbkwik: A consolidated knowledge graph from thousands of wikis. In *International Conference on Big Knowledge (ICBK)*, 2018.

Sven Hertling and Heiko Paulheim. Dbkwik: extracting and integrating knowledge from thousands of wikis. *Knowl. Inf. Syst.*, 2020.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2011.

Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence*, 2013.

Alexandra Hofmann, Samresh Perchani, Jan Portisch, Sven Hertling, and Heiko Paulheim. Dbkwik: Towards knowledge graph creation from thousands of wikis. In *the International Semantic Web Conference (ISWC)*, 2017.

Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Information retrieval in folksonomies: Search and ranking. In *the Extended Semantic Web Conference (ESWC)*, 2006.

Scott B Huffman. Learning information extraction patterns from examples. In *the International Joint Conference on Artificial Intelligence (IJCAI)*, 1995.

Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. Feuding families and former friends: Unsupervised learning for dynamic fictional relationships. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2016.

Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. A survey on conversational recommender systems. *ACM Computing Surveys*, 2020.

Robert Jäschke, Leandro Balby Marinho, Andreas Hotho, Lars Schmidt-Thieme, and Gerd Stumme. Tag recommendations in folksonomies. In *European Conference on Principles of Data Mining and Knowledge Discovery (PKDD)*, 2007.

Harshita Jhavar and Paramita Mirza. EMOFIEL: mapping emotions of relationships in a story. In *The Web Conference*, 2018.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. Distant supervision for relation extraction with sentence-level attention and entity descriptions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics (TACL)*, 2020.

Zhanming Jie and Wei Lu. Dependency-guided lstm-crf for named entity recognition. *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Hailong Jin, Lei Hou, Juanzi Li, and Tiansi Dong. Attributed and predictive entity embedding for fine-grained entity typing in knowledge bases. In *the International Conference on Computational Linguistics (COLING)*, 2018.

Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. A neural layered model for nested named entity recognition. In *the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018.

Aditya Kalyanpur, J. William Murdock, James Fan, and Christopher A. Welty. Leveraging community-built knowledge for type coercion in question answering. In *the International Semantic Web Conference (ISWC)*, 2011.

Nanda Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2004.

Arzoo Katiyar and Claire Cardie. Nested named entity recognition revisited. In *the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018.

Jun-Tae Kim and Dan I. Moldovan. Acquisition of linguistic patterns for knowledge-based information extraction. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 1995.

Rajasekar Krishnamurthy, Yunyao Li, Sriram Raghavan, Frederick Reiss, Shivakumar Vaithyanathan, and Huaiyu Zhu. Systemt: a system for declarative information extraction. *ACM SIGMOD Record*, 2009.

Vijay Krishnan and Christopher D Manning. An effective two-stage model for exploiting non-local dependencies in named entity recognition. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2006.

Onur Kuru, Ozan Arkan Can, and Deniz Yuret. Charner: Character-level named entity recognition. In *the International Conference on Computational Linguistics (COLING)*, 2016.

Vincent Labatut and Xavier Bost. Extraction and analysis of fictional character networks: A survey. *ACM Computing Surveys (CSUR)*, 2019.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT)*, 2016.

Changki Lee, Yi-Gyu Hwang, Hyo-Jung Oh, Soojong Lim, Jeong Heo, Chung-Hee Lee, Hyeon-Jin Kim, Ji-Hyun Wang, and Myung-Gil Jang. Fine-grained named entity recognition using conditional random fields for question answering. In *Asia Information Retrieval Symposium*, 2006.

Joohong Lee, Sangwoo Seo, and Yong Suk Choi. Semantic relation classification via bidirectional lstm networks with entity-aware attention using latent entity typing. *Symmetry*, 2019.

Douglas B Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 1995.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2020a.

Peng-Hsuan Li, Ruo-Ping Dong, Yu-Siang Wang, Ju-Chieh Chou, and Wei-Yun Ma. Leveraging linguistic structures for named entity recognition with bidirectional recursive neural networks. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017.

Yang Li, Guodong Long, Tao Shen, Tianyi Zhou, Lina Yao, Huan Huo, and Jing Jiang. Self-attention enhanced selective gate with entity-aware embedding for distantly supervised relation extraction. In *the AAAI Conference on Artificial Intelligence*, 2020b.

Yunyao Li, Frederick Reiss, and Laura Chiticariu. Systemt: A declarative information extraction system. In *Proceedings of the ACL-HLT 2011 System Demonstrations*, 2011.

Wenhui Liao and Sriharsha Veeramachaneni. A simple semi-supervised algorithm for named entity recognition. In *Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT)*, 2009.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *the AAAI Conference on Artificial Intelligence*, 2015.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Neural relation extraction with selective attention over instances. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2124–2133, 2016.

Ying Lin and Heng Ji. An attentive fine-grained entity typing model with latent type representation. In *Conference on Empirical Methods in Natural Language Processing International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

Xiao Ling and Daniel S Weld. Fine-grained entity recognition. In *AAAI Conference on Artificial Intelligence*, 2012.

Hugo Liu and Push Singh. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 2004.

Xitong Liu and Hui Fang. Latent entity space: a novel retrieval approach for entity-bearing queries. *Information Retrieval Journal*, 2015.

Xueqing Liu, Yangqiu Song, Shixia Liu, and Haixun Wang. Automatic taxonomy construction from keywords. In *ACM SIGKDD Conference on Knowledge Discovery Data Mining*, 2012.

Zengjian Liu, Ming Yang, Xiaolong Wang, Qingcai Chen, Buzhou Tang, Zhe Wang, and Hua Xu. Entity recognition from clinical texts via recurrent neural network. In *BMC medical informatics and decision making*, 2017.

Xusheng Luo, Luxin Liu, Yonghua Yang, Le Bo, Yuanpeng Cao, Jinghang Wu, Qiang Li, Keping Yang, and Kenny Q Zhu. Alicoco: Alibaba e-commerce cognitive concept net. In *the ACM SIGMOD Conference*, 2020.

Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.

Aibek Makazhanov, Denilson Barbosa, and Grzegorz Kondrak. Extracting family relationship networks from novels. *arXiv preprint arXiv:1405.0603*, 2014.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The stanford CoreNLP natural language processing toolkit. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2014.

Yuning Mao, Xiang Ren, Jiaming Shen, Xiaotao Gu, and Jiawei Han. End-to-end reinforcement learning for automatic taxonomy induction. *the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.

Félix Martel and Amal Zouaq. Taxonomy extraction using knowledge graph embeddings and hierarchical clustering. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, 2021.

Mausam. Open information extraction systems and downstream applications. In *the International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. 2003.

Edgar Meij. Understanding news using the bloomberg knowledge graph. *Invited talk at the Big Data Innovators Gathering (TheWebConf)*, 2019.

Filipe Mesquita, Matteo Cannaviccio, Jordan Schmidek, Paramita Mirza, and Denilson Barbosa. Knowledgenet: A benchmark dataset for knowledge base population. In

Conference on Empirical Methods in Natural Language Processing International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019.

Andrei Mikheev, Marc Moens, and Claire Grover. Named entity recognition without gazetteers. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 1999.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *the Annual Conference on Neural Information Processing Systems (NIPS)*, 2013.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 2009.

Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bishan Yang, Justin Betteridge, Andrew Carlson, Bhavana Dalvi, Matt Gardner, Bryan Kisiel, et al. Never-ending learning. *Communications of the ACM*, 2018.

Seungwhan Moon, Leonardo Neves, and Vitor Carvalho. Multimodal named entity recognition for short social media posts. *the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018.

R Mooney. Relational learning of pattern-match rules for information extraction. In *the International Joint Conference on Artificial Intelligence (IJCAI)*, 1999.

David Nadeau, Peter D Turney, and Stan Matwin. Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. In *Conference of the Canadian society for computational studies of intelligence*, 2006.

Sreyasi Nag Chowdhury, Simon Razniewski, and Gerhard Weikum. Sandi: Story-and-images alignment. In *the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2021.

Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. Patty: A taxonomy of relational patterns with semantic types. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2012.

Ndapandula Nakashole, Tomasz Tylenda, and Gerhard Weikum. Fine-grained semantic typing of emerging entities. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2013.

Vivi Nastase, Michael Strube, Benjamin Boerschinger, Cäcilia Zirn, and Anas Elghafari. Wikinet: A very large scale multi-lingual concept network. In *the International Conference on Language Resources and Evaluation (LREC)*, 2010.

Roberto Navigli and Simone Paolo Ponzetto. Babelnet: Building a very large multilingual semantic network. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2010.

Dat Ba Nguyen, Abdalghani Abujabal, Khanh Tran, Martin Theobald, and Gerhard Weikum. Query-driven on-the-fly knowledge base construction. *the International Conference on Very Large Data Bases (VLDB)*, 2017a.

Dat PT Nguyen, Yutaka Matsuo, and Mitsuru Ishizuka. Relation extraction from wikipedia using subtree mining. In *the AAAI Conference on Artificial Intelligence*, 2007.

Kim Anh Nguyen, Maximilian Köper, Sabine Schulte im Walde, and Ngoc Thang Vu. Hierarchical embeddings for hypernymy detection and directionality. *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017b.

Thien Huu Nguyen, Avirup Sil, Georgiana Dinu, and Radu Florian. Toward mention detection robustness with recurrent neural networks. *arXiv preprint arXiv:1602.07749*, 2016.

Tuan-Phong Nguyen, Simon Razniewski, and Gerhard Weikum. Advanced semantics for commonsense knowledge extraction. In *The Web Conference*, 2021.

Ian Niles and Adam Pease. Towards a standard upper ontology. In *International Conference on Formal Ontology in Information Systems (FOIS)*, 2001.

Tommaso Di Noia, Vito Claudio Ostuni, Paolo Tomeo, and Eugenio Di Sciascio. Sprank: Semantic path-based ranking for top-n recommendations using linked open data. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2016.

Yasumasa Onoe and Greg Durrett. Learning to denoise distantly-labeled data for entity typing. *the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

Yasumasa Onoe and Greg Durrett. Interpretable entity representations through large-scale typing. *EMNLP Findings*, 2020.

Marius Pasca. Open-domain fine-grained class extraction from web search queries. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

Marius Pasca. Finding needles in an encyclopedic haystack: Detecting classes among wikipedia articles. In *the Web Conference*, 2018.

Marius Pasca and Benjamin Van Durme. What you seek is what you get: Extraction of class attributes from query logs. In *the International Joint Conference on Artificial Intelligence (IJCAI) 2007*, 2007.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. Language models as knowledge bases? *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Simone Paolo Ponzetto and Roberto Navigli. Large-scale taxonomy mapping for restructuring and integrating Wikipedia. In *the International Joint Conference on Artificial Intelligence (IJCAI)*, 2009.

Simone Paolo Ponzetto and Michael Strube. Deriving a large scale taxonomy from Wikipedia. In *AAAI Conference on Artificial Intelligence*, 2007.

Simone Paolo Ponzetto and Michael Strube. Taxonomy induction based on a collaboratively built knowledge repository. *Artificial Intelligence*, 2011.

Jay Pujara, Hui Miao, Lise Getoor, and William W. Cohen. Using semantics and statistics to turn data into knowledge. *AI Mag.*, 2015.

Alexandra Pomares Quimbaya, Alejandro Sierra Múnera, Rafael Andrés González Rivera, Julián Camilo Daza Rodríguez, Oscar Mauricio Muñoz Velandia, Angel Alberto Garcia Peña, and Cyril Labbé. Named entity recognition over electronic health records through a combined dictionary-based approach. *Procedia Computer Science*, 2016.

Hadas Raviv, Oren Kurland, and David Carmel. Document retrieval using entity-based language models. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2016.

Marta Recasens, Marie-Catherine de Marneffe, and Christopher Potts. The life and death of discourse entities: Identifying singleton mentions. In *Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT)*, 2013.

Marek Rei, Gamal KO Crichton, and Sampo Pyysalo. Attending to characters in neural sequence labeling models. *the International Conference on Computational Linguistics (COLING)*, 2016.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Frederick Reiss, Sriram Raghavan, Rajasekar Krishnamurthy, Huaiyu Zhu, and Shivakumar Vaithyanathan. An algebraic approach to rule-based information extraction. In *IEEE International Conference on Data Engineering (ICDE)*, 2008.

Xiang Ren, Wenqi He, Meng Qu, Lifu Huang, Heng Ji, and Jiawei Han. Afet: Automatic fine-grained entity typing by hierarchical partial-label embedding. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *European Conference on Machine Learning and Data Mining (ECML-PKDD)*, 2010.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. Relation extraction with matrix factorization and universal schemas. In *the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2013.

Tim Rocktäschel, Michael Weidlich, and Ulf Leser. Chemspot: a hybrid system for chemical named entity recognition. *Bioinformatics*, 2012.

Stephen Roller and Katrin Erk. Relations such as hypernymy: Identifying and exploiting Hearst patterns in distributional vectors for lexical entailment. *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

Stephen Roller, Katrin Erk, and Gemma Boleda. Inclusive yet selective: Supervised distributional hypernymy detection. In *the International Conference on Computational Linguistics (COLING)*, 2014.

Julien Romero, Simon Razniewski, Koninika Pal, Jeff Z. Pan, Archit Sakhadeo, and Gerhard Weikum. Commonsense properties from query logs and question answering forums. In *the Conference on Information and Knowledge Management (CIKM)*, 2019.

Christopher De Sa, Alexander Ratner, Christopher Ré, Jaeho Shin, Feiran Wang, Sen Wu, and Ce Zhang. Incremental knowledge base construction using deepdive. *the International Conference on Very Large Data Bases (VLDB) J.*, 2017.

Mark Sanderson and W. Bruce Croft. Deriving concept hierarchies from text. In *the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999.

Erik F Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *Proceedings of CoNLL-2003*, 2003.

Karin Kipper Schuler. *VerbNet: A broad-coverage, comprehensive verb lexicon.* 2005.

Julian Seitner, Christian Bizer, Kai Eckert, Stefano Faralli, Robert Meusel, Heiko Paulheim, and Simone Paolo Ponzetto. A large database of hypernymy relations extracted from the web. In *the International Conference on Language Resources and Evaluation (LREC)*, 2016.

Satoshi Sekine and Chikashi Nobata. Definition, dictionaries and tagger for extended named entity hierarchy. In *the International Conference on Language Resources and Evaluation (LREC)*, 2004.

Jingbo Shang, Liyuan Liu, Xiang Ren, Xiaotao Gu, Teng Ren, and Jiawei Han. Learning named entity tagger using domain-specific dictionary. *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Wei Shen, Jianyong Wang, and Jiawei Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2014.

Peng Shi and Jimmy Lin. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*, 2019.

Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. Neural architectures for fine-grained entity type classification. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2017.

Jaeho Shin, Sen Wu, Feiran Wang, Christopher De Sa, Ce Zhang, and Christopher Ré. Incremental knowledge base construction using deepdive. In *the International Conference on Very Large Data Bases (VLDB)*, 2015.

Vered Shwartz, Enrico Santus, and Dominik Schlechtweg. Hypernyms under siege: Linguistically-motivated artillery for hypernymy detection. *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2016.

Rion Snow, Daniel Jurafsky, and Andrew Y Ng. Learning syntactic patterns for automatic hypernym discovery. In *the Annual Conference on Neural Information Processing Systems (NIPS)*, 2005.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. Matching the blanks: Distributional similarity for relation learning. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Stephen Soderland, David Fisher, Jonathan Aseltine, and Wendy Lehnert. Crystal: Inducing a conceptual dictionary. *the International Joint Conference on Artificial Intelligence (IJCAI)*, 1995.

Shashank Srivastava, Snigdha Chaturvedi, and Tom Mitchell. Inferring interpersonal relations in narrative summaries. In *the AAAI Conference on Artificial Intelligence*, 2016a.

Shashank Srivastava, Snigdha Chaturvedi, and Tom M Mitchell. Inferring interpersonal relations in narrative summaries. In *AAAI Conference on Artificial Intelligence*, 2016b.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. Supervised open information extraction. In *the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018.

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. Fast and accurate entity recognition with iterated dilated convolutions. *the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. YAGO: A core of semantic knowledge. In *the Web Conference*, 2007.

Fabian M Suchanek, Mauro Sozio, and Gerhard Weikum. Sofie: a self-organizing framework for information extraction. In *the Web Conference*, 2009.

György Szarvas, Richárd Farkas, and András Kocsor. A multilingual named entity recognition system using boosting and c4. 5 decision tree learning algorithms. In *International Conference on Discovery Science*, 2006.

Niket Tandon, Gerard De Melo, Fabian Suchanek, and Gerhard Weikum. Webchild: Harvesting and organizing commonsense knowledge from the web. In *ACM International WSDM Conference*, 2014.

Makarand Tapaswi, Martin Bauml, and Rainer Stiefelhagen. Book2movie: Aligning video scenes with book chapters. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Kentaro Torisawa et al. Exploiting wikipedia as external knowledge for named entity recognition. In *Conference on Empirical Methods on Natural Language Processing and Computational Natural Language Learning (EMNLP-CNLP)*, 2007.

Bayu Distiawan Trisedya, Gerhard Weikum, Jianzhong Qi, and Rui Zhang. Neural relation extraction for knowledge base enrichment. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Hardik Vala, David Jurgens, Andrew Piper, and Derek Ruths. Mr. bennet, his coachman, and the archbishop walk into a bar but only one of them gets recognized: On the difficulty of detecting characters in literary texts. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *the Annual Conference on Neural Information Processing Systems (NIPS)*, 2017.

Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 2014.

Tu Vu and Vered Shwartz. Integrating multiplicative features into supervised distributional methods for lexical entailment. *SEM Conference*, 2018.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. Entity, relation, and event extraction with contextualized span representations. *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Difeng Wang, Wei Hu, Ermei Cao, and Weijian Sun. Global-to-local neural networks for document-level relation extraction. *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Hong Wang, Christfried Focke, Rob Sylvester, Nilesh Mishra, and William Wang. Fine-tune bert for docred with two-step process. *arXiv preprint arXiv:1909.11898*, 2019.

Linlin Wang, Zhu Cao, Gerard De Melo, and Zhiyuan Liu. Relation classification via multi-level attention cnns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.

Mengqiu Wang. A re-examination of dependency path kernels for relation extraction. In *International Joint Conference on Natural Language Processing (IJCNLP)*, 2008.

Xiaozhi Wang, Xu Han, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Adversarial multi-lingual neural relation extraction. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, 2018.

Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. Connecting language and knowledge bases with embedding models for relation extraction. *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

Fei Wu, Raphael Hoffmann, and Daniel S. Weld. Information extraction from Wikipedia: Moving down the long tail. In *ACM SIGKDD Conference on Knowledge Discovery Data Mining*, 2008.

Peiyun Wu, Xiaowang Zhang, and Zhiyong Feng. A survey of question answering over knowledge base. In *China Conference on Knowledge Graph and Semantic Computing*, 2019.

Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. Probase: A probabilistic taxonomy for text understanding. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, 2012a.

Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Qili Zhu. Probase: A probabilistic taxonomy for text understanding. In *the ACM SIGMOD Conference*, 2012b.

Yonghui Wu, Jun Xu, Min Jiang, Yaoyun Zhang, and Hua Xu. A study of neural word embeddings for named entity recognition in clinical text. In *AMIA Annual Symposium Proceedings*, 2015.

Zhibiao Wu and Martha Palmer. Verb semantics and lexical selection. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, 1994.

Wenhan Xiong, Jiawei Wu, Deren Lei, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. Imposing label-relational inductive bias for extremely fine-grained entity typing. *the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

Bo Xu, Zheng Luo, Luyang Huang, Bin Liang, Yanghua Xiao, Deqing Yang, and Wei Wang. Metic: Multi-instance entity typing from corpus. In *the Conference on Information and Knowledge Management (CIKM)*, 2018.

Peng Xu and Denilson Barbosa. Neural fine-grained entity type classification with hierarchy-aware loss. In *Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT)*, 2018.

Yan Xu, Ran Jia, Lili Mou, Ge Li, Yunchuan Chen, Yangyang Lu, and Zhi Jin. Improved relation classification by deep recurrent neural networks with data augmentation. In *International Conference on Computational Linguistics (COLING)*, 2016.

Lin Yao, Hong Liu, Yi Liu, Xinxin Li, and Muhammad Waqas Anwar. Biomedical named entity recognition based on deep neutral network. *Int. J. Hybrid Inf. Technol*, 2015.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. Docred: A large-scale document-level relation extraction dataset. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Dani Yogatama, Daniel Gillick, and Nevena Lazic. Embedding methods for fine grained entity type classification. In *Annual Meeting of the Association for Computational Linguistics International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, 2015.

Mohamed Amir Yosef, Sandro Bauer, Johannes Hoffart, Marc Spaniol, and Gerhard Weikum. Hyena: Hierarchical type classification for entity names. In *the International Conference on Computational Linguistics (COLING)*, 2012.

Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. In *IEEE CIM*, 2018.

Dian Yu, Kai Sun, Claire Cardie, and Dong Yu. Dialogue-based relation extraction. *the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Zheng Yu, Haixun Wang, Xuemin Lin, and Min Wang. Learning term embeddings for hypernymy identification. In *the International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via convolutional deep neural network. In *International Conference on Computational Linguistics (COLING)*, 2014.

Shaodian Zhang and Noémie Elhadad. Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts. *Journal of biomedical informatics*, 2013.

Shu Zhang, Dequan Zheng, Xinchen Hu, and Ming Yang. Bidirectional long short-term memory networks for relation classification. In *Proceedings of the 29th Pacific Asia conference on language, information and computation*, 2015.

Yu Zhang, Guoguo Chen, Dong Yu, Kaisheng Yaco, Sanjeev Khudanpur, and James Glass. Highway long short-term memory rnns for distant speech recognition. In *the International Conference on Acoustics, Speech, Signal Processing*, 2016.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. Position-aware attention and supervised data improve slot filling. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017a.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. Position-aware attention and supervised data improve slot filling. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017b.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. Joint extraction of entities and relations based on a novel tagging scheme. *the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017.

GuoDong Zhou and Jian Su. Named entity recognition using an hmm-based chunk tagger. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002.

GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. Exploring various knowledge in relation extraction. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2005.

Wenxuan Zhou, Kevin Huang, Tengyu Ma, and Jing Huang. Document-level relation extraction with adaptive thresholding and localized context pooling. *the AAAI Conference on Artificial Intelligence*, 2021.