TiFi: Taxonomy Induction for Fictional Domains*

Cuong Xuan Chu
Max Planck Institute for Informatics
Saarbrücken, Germany
cxchu@mpi-inf.mpg.de

Simon Razniewski
Max Planck Institute for Informatics
Saarbrücken, Germany
srazniew@mpi-inf.mpg.de

Gerhard Weikum
Max Planck Institute for Informatics
Saarbrücken, Germany
weikum@mpi-inf.mpg.de

ABSTRACT

Taxonomies are important building blocks of structured knowledge bases, and their construction from text sources and Wikipedia has received much attention. In this paper we focus on the construction of taxonomies for fictional domains, using noisy category systems from fan wikis or text extraction as input. Such fictional domains are archetypes of entity universes that are poorly covered by Wikipedia, such as also enterprise-specific knowledge bases or highly specialized verticals. Our fiction-targeted approach, called TiFi, consists of three phases: (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.

ACM Reference Format:
Cuong Xuan Chu, Simon Razniewski, and Gerhard Weikum. 2019. TiFi: Taxonomy Induction for Fictional Domains. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3308558.3313519

1 INTRODUCTION

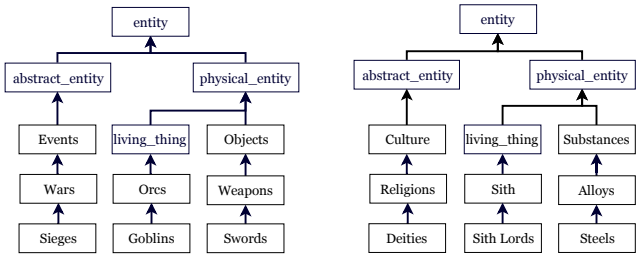
1.1 Motivation and Problem

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 [27]. Taxonomies can be hand-crafted, examples being WordNet [13], SUMO [31] or MeSH and UMLS [4], 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 [20] or bootstrapped with Hearst patterns (e.g., [47]), harvesting and learning from Wikipedia categories as a noisy seed network (e.g., [8, 14, 16, 35–37, 44, 46]), and inducing type hierarchies from query-and-click logs (e.g., [18, 32, 34]).

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 [3, 24, 43], learning story patterns [5, 6] or investigating gender bias in different cultures [1]. Thus, organizing entities and classes from fictional domains into clean taxonomies (see example in Fig. 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



(a) LoTR (b) Star Wars

Figure 1: Excerpts of LoTR and Star Wars taxonomies.

*Extended version available at https://arxiv.org/abs/1901.10263.

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313519

about the surface forms of entity names (e.g., on person names: with or without first name, starting with 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., 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 paper focuses on fiction and fantasy, though, where raw content sources are publicly available.

1.2 Approach and Contribution

In this paper 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 [16], 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 and code are available at https://www.mpi-inf.mpg.de/yago-naga/tifi.

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 [5, 6], and for extrinsic reasons such as predicting human behaviour [12] or measuring discrimination [1]. Other recurrent topics are, for instance, to discover character relationships [3, 24, 43], to model social networks [3, 9], or to describe personalities and emotions [10, 26].

Taxonomy Induction from Text. A seminal contribution towards their automated construction was the discovery of Hearst patterns [20], 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 [38, 42], devise probabilistic models and graph-flow algorithms [15], utilize distributional representations of types [30, 39, 45, 49], or aim to infer pairwise or hierarchical hypernymy [30, 49].

Taxonomy Construction using Wikipedia. A popular structured source for taxonomy construction is the Wikipedia category network (WCN) for taxonomy induction. One project, WikiTaxonomy [36, 37] 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 [22, 44] 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 [13] categories, choosing in case of ambiguity simply the meaning appearing topmost in WordNet. WiBi (Wikipedia Bitaxonomy) [14] proceeds in two steps: It first builds a taxonomy from Wikipedia pages by extracting and disambiguating lemmas from the first sentence of pages, then combines the page taxonomy and the original Wikipedia category network to induce the final taxonomy. A recent system, HEAD [16], exploits multiple lexical and structural rules towards classifying subcategory relations.

Domain-specific Taxonomies. TAXIFY is an unsupervised approach to domain-specific taxonomy construction from text [2]. 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. [28] 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.

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

These Wikis, like on Wikipedia, 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 DESIGN RATIONALE AND OVERVIEW

3.1 Design Space and Choices

Input: The input of TiFi is a noisy category/tag tree or graph for a given set of entities (with textual description), from a domain of interest. Entities 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., [11, 23, 25]) and use this as an initial category system. When only text is available, we can use Hearst patterns and other text-based techniques [7, 20, 40] to generate categories and construct a subsumption-based tree.

Output: 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 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.
- 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 [13].
- 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, and to devise features that capture specific context.

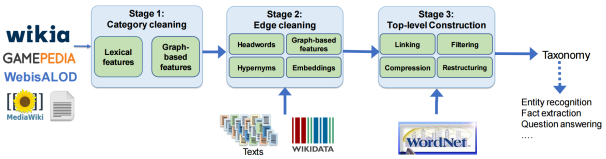


Figure 2: Architecture of TiFi.

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. 2.

The first step, *category cleaning* (Section 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. While previous work has tackled this step via syntactic and lexical rules [33, 36, 44], for generalizability, we opt for a supervised classification approach that combines rules from above with additional graph-based features.

The second step, *edge cleaning* (Section 5), identifies the edges from the original category network E that truly represent subcategory relationships. We combine ideas from previous rule-based [17, 36] and embedding-based approaches [30] with various adapted semantic features and novel graph-based features.

For the third step, *top-level construction* (Section 6), we choose to reuse the existing WordNet top-level classes. This 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).

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: (i) *Meta-categories:* Wiki platforms typically introduce metacategories related to administration and technical setup, e.g., `Meta` or `Administration`. (ii) *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. (iii) *Instances:* Editors frequently create categories that are actually instances, e.g., Arda or Mordor in *The Lord of The Rings*). (iv) *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 [33, 36, 37]. 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, administration, file, pages`, 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. 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 7, this varied feature set allows to identify in-domain classes with 83%-85% precision.

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 [8, 14, 16, 36], 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 relation 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 [16].

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.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 `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 [14] 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 [29] 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 [19], and despite the subclass relation being asymmetric, symmetric similarity measures have been found to be useful for taxonomy construction [41]. 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 first is the symmetric Wu-Palmer score compares the depth of two synsets (the headwords of the categories) with the depth of their least common subsumer,

while the second is the [48]. HyperVec score [30], which not only shows the similarity between a category and its hypernym, but is also directional.

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 captures 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 7 shows that this feature set allows to surpass the state-of-the-art in edge cleaning by 6-8% F1-score.

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 [8, 14, 16, 36, 37], 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 based on context similarity. 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. **??** (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. For instance, 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.

Table 1: Input categories from Wikia/Gamepedia.

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

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. 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 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.

As baselines we employ a rule-based approach by Ponzetto & Strube [37], to the best of our knowledge the best performing method for general category cleaning, and recent work by Marius Pasca [33] 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 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 rule-based baseline.

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, where for universes other than LoTR and GoT, having annotated only 50 samples. As one can see, F1-scores

Table 2: Step 1 - In-domain category cleaning.

Method	Universe	Precision	Recall	F1-score
Pasca 2018 [33]	LoTR	0.33	0.75	0.46
	GoT	0.57	0.85	0.68
Ponzetto & Strube 2011 [37]	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**

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.

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.

We compare TiFi with two state-of-the-art systems: (1) HEAD [16], the most recent system for Wikipedia category relationship cleaning, and (2) HyperVec [30], a recent embedding-based hypernym relationship learning system. The results for in-domain evaluation using 10-fold crossvalidation are shown in Table 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 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.

In the ablation test, all three groups of features showed importance, each leading to a 1-4% drop in F1-score when being removed. The individually most important features were *Only Plural Parent, Headword Matching, Common Children Support* and *Page Type Matching*.

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%. 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.

Final Taxonomies. Table 6 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. The taxonomies are available at https://www.mpi-inf.mpg.de/yago-naga/tifi.

Other Inputs. We also evaluated TiFi on the real-world Wikipedia dataset, and on the noisy WebIsALOD [21] hypernymy dataset. We find that TiFi performs comparable to a state-of-the-art Wikipedia baseline [37] on Wikipedia, and that it significantly outperforms [37] and [33] on the noisier WebIsALOD data. More details are in the extended version.

Table 3: Step 1 - Cross-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 4: Step 2 - In-domain edge cleaning.

Method	Universe	Precision	Recall	F1-score
HyperVec [30]	LoTR	0.82	0.8	0.81
	GoT	**0.83**	0.81	0.82
HEAD [16]	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 5: Step 2 - Cross-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 6: Taxonomies produced by TiFi.

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

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. The number of answers and their precision@10 is shown in Table 7. As one can see, entity search using TiFi yields significantly higher precision than Google and Wikia text search.

Table 7: Avg. #Answers and precision of entity search.

	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%)

9 CONCLUSION

In this paper 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.

REFERENCES

[1] Apoorv Agarwal, Jiehan Zheng, Shruti Kamath, Sriramkumar Balasubramanian, and Shirin Ann Dey. 2015. Key Female Characters in Film have more to talk about besides men: Automating the Bechdel Test. In *NAACL*. 830–840.

[2] Daniele Alfarone and Jesse Davis. 2015. Unsupervised Learning of an is-a Taxonomy from a Limited Domain-specific Corpus. In *IJCAI*. 1434–1441.

[3] David Bamman, Brendan O'Connor, and Noah A Smith. 2014. Learning Latent Personas of Film Characters. In *ACL*. 352.

[4] Olivier Bodenreider. 2004. The Unified Medical Language System (UMLS): Integrating Biomedical Terminology. *Nucleic acids research* (2004), D267–D270.

[5] Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised Learning of Narrative Schemas and their Participants. In *ACL/IJCNLP*. 602–610.

[6] Snigdha Chaturvedi, Mohit Iyyer, and Hal Daumé III. 2017. Unsupervised Learning of Evolving Relationships Between Literary Characters. In *AAAI*. 3159–3165.

[7] Philipp Cimiano, Andreas Hotho, and Steffen Staab. 2005. Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. *J. Artif. Intell. Res.* 24 (2005), 305–339.

[8] Gerard de Melo and Gerhard Weikum. 2010. MENTA: Inducing Multilingual Taxonomies from Wikipedia. In *CIKM*. 1099–1108.

[9] Vinodh Krishnan Elangovan and Jacob Eisenstein. 2015. "You're Mr. Lebowski, I'm the Dude": Inducing Address Term Formality in Signed Social Networks. In *NAACL*. 1616–1626.

[10] David K Elson, Nicholas Dames, and Kathleen R McKeown. 2010. Extracting Social Networks from Literary Fiction. In *ACL*. 138–147.

[11] Quan Fang, Changsheng Xu, Jitao Sang, M. Shamim Hossain, and Ahmed Ghoneim. 2016. Folksonomy-Based Visual Ontology Construction and Its Applications. *IEEE Trans. Multimedia* 18, 4 (2016), 702–713.

[12] Ethan Fast, William McGrath, Pranav Rajpurkar, and Michael S Bernstein. 2016. Augur: Mining Human Behaviors from Fiction to Power Interactive Systems. In *CHI*. 237–247.

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

[14] Tiziano Flati, Daniele Vannella, Tommaso Pasini, and Roberto Navigli. 2014. Two Is Bigger (and Better) Than One: the Wikipedia Bitaxonomy Project.. In *ACL*. 945–955.

[15] Amit Gupta, Rémi Lebret, Hamza Harkous, and Karl Aberer. 2017. Taxonomy Induction using Hypernym Subsequences. In *CIKM*. 1329–1338.

[16] Amit Gupta, Francesco Piccinno, Mikhail Kozhevnikov, Marius Pasca, and Daniele Pighin. 2016. Revisiting Taxonomy Induction over Wikipedia. In *COLING*. 2300–2309.

[17] Amit Gupta, Francesco Piccinno, Mikhail Kozhevnikov, Marius Pasca, and Daniele Pighin. 2016. Revisiting Taxonomy Induction over Wikipedia. In *COLING 2016*. 2300–2309.

[18] Rahul Gupta, Alon Y. Halevy, Xuezhi Wang, Steven Euijong Whang, and Fei Wu. 2014. Biperpedia: An Ontology for Search Applications. *PVLDB* (2014), 505–516.

[19] Zellig S Harris. 1954. Distributional Structure. *Word* (1954), 146–162.

[20] Marti A Hearst. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In *COLING*. 539–545.

[21] Sven Hertling and Heiko Paulheim. 2017. WebIsALOD: providing hypernymy relations extracted from the web as linked open data. In *ISWC*.

[22] Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia. *Artificial Intelligence* (2013), 28–61.

[23] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. 2006. Information Retrieval in Folksonomies: Search and Ranking. In *ESWC 2006*. 411–426.

[24] Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. 2016. Feuding Families and Former Friends: Unsupervised Learning for Dynamic Fictional Relationships. In *NAACL*. 1534–1544.

[25] Robert Jäschke, Leandro Balby Marinho, Andreas Hotho, Lars Schmidt-Thieme, and Gerd Stumme. 2007. Tag Recommendations in Folksonomies. In *PKDD 2007*. 506–514.

[26] Harshita Jhavar and Paramita Mirza. 2018. EMOFIEL: Mapping Emotions of Relationships in a Story. In *The Web Conference*. 243–246.

[27] Aditya Kalyanpur, J. William Murdock, James Fan, and Christopher A. Welty. 2011. Leveraging Community-Built Knowledge for Type Coercion in Question Answering. In *ISWC 2011*. 144–156.

[28] Xueqing Liu, Yangqiu Song, Shixia Liu, and Haixun Wang. 2012. Automatic Taxonomy Construction from Keywords. In *KDD*. 1433–1441.

[29] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *ACL*. 55–60.

[30] Kim Anh Nguyen, Maximilian Köper, Sabine Schulte im Walde, and Ngoc Thang Vu. 2017. Hierarchical Embeddings for Hypernymy Detection and Directionality. *arXiv preprint arXiv:1707.07273* (2017).

[31] Ian Niles and Adam Pease. 2001. Towards a Standard Upper Ontology. In *FOIS*. 2–9.

[32] Marius Pasca. 2013. Open-Domain Fine-Grained Class Extraction from Web Search Queries. In *EMNLP 2013*. 403–414.

[33] Marius Pasca. 2018. Finding Needles in an Encyclopedic Haystack: Detecting Classes Among Wikipedia Articles. In *WWW*. 1267–1276.

[34] Marius Pasca and Benjamin Van Durme. 2007. What You Seek Is What You Get: Extraction of Class Attributes from Query Logs. In *IJCAI 2007*. 2832–2837.

[35] Simone Paolo Ponzetto and Roberto Navigli. 2009. Large-Scale Taxonomy Mapping for Restructuring and Integrating Wikipedia. In *IJCAI*. 2083–2088.

[36] Simone Paolo Ponzetto and Michael Strube. 2007. Deriving a Large Scale Taxonomy from Wikipedia. In *AAAI*. 1440–1445.

[37] Simone Paolo Ponzetto and Michael Strube. 2011. Taxonomy Induction Based on a Collaboratively Built Knowledge Repository. *Artificial Intelligence* (2011), 1737–1756.

[38] Stephen Roller and Katrin Erk. 2016. Relations such as Hypernymy: Identifying and Exploiting Hearst Patterns in Distributional Vectors for Lexical Entailment. *arXiv preprint arXiv:1605.05433* (2016).

[39] Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet Selective: Supervised Distributional Hypernymy Detection. In *COLING*. 1025–1036.

[40] Mark Sanderson and W. Bruce Croft. 1999. Deriving Concept Hierarchies from Text. In *SIGIR 1999*. 206–213.

[41] Vered Shwartz, Enrico Santus, and Dominik Schlechtweg. 2016. Hypernyms under Siege: Linguistically-motivated Artillery for Hypernymy Detection. *arXiv preprint arXiv:1612.04460* (2016).

[42] Rion Snow, Daniel Jurafsky, and Andrew Y Ng. 2005. Learning Syntactic Patterns for Automatic Hypernym Discovery. In *NIPS*. 1297–1304.

[43] Shashank Srivastava, Snigdha Chaturvedi, and Tom M Mitchell. 2016. Inferring Interpersonal Relations in Narrative Summaries. In *AAAI*. 2807–2813.

[44] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. YAGO: A Core of Semantic Knowledge. In *WWW*. 697–706.

[45] Tu Vu and Vered Shwartz. 2018. Integrating Multiplicative Features into Supervised Distributional Methods for Lexical Entailment. *arXiv preprint arXiv:1804.08845* (2018).

[46] Fei Wu, Raphael Hoffmann, and Daniel S. Weld. 2008. Information Extraction from Wikipedia: Moving Down the Long Tail. In *KDD 2008*. 731–739.

[47] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Qili Zhu. 2012. Probase: A Probabilistic Taxonomy for Text Understanding. In *SIGMOD 2012*. 481–492.

[48] Zhibiao Wu and Martha Palmer. 1994. Verb Semantics and Lexical Selection. In *ACL*. 133–138.

[49] Zheng Yu, Haixun Wang, Xuemin Lin, and Min Wang. 2015. Learning Term Embeddings for Hypernymy Identification. In *IJCAI*. 1390–1397.