ENTYFI: Entity Typing in Fictional Texts

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

Fiction and fantasy are archetypes of long-tail domains that lack comprehensive methods for automated language processing and knowledge extraction. We present ENTYFI, the first methodology 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 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.

KEYWORDS

named entity recognition; entity typing; fictional domains; knowledge acquisition; neural networks

ACM Reference Format:
Cuong Xuan Chu, Simon Razniewski, and Gerhard Weikum. 2020. ENTYFI: Entity Typing in Fictional Texts. In *The Thirteenth ACM International Conference on Web Search and Data Mining (WSDM '20), February 3–7, 2020, Houston, TX, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3336191.3371808

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 [3, 21, 26]. 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 [6, 9, 21, 22].

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. Fiction and fantasy are core parts of human culture, exemplified by an average TV consumption of 3 hours per day in 2015 [2], much of this spent on fictional TV series. 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 paper 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. In this paper, 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

WSDM '20, February 3–7, 2020, Houston, TX, USA

ACM ISBN 978-1-4503-6822-3/20/02…$15.00
https://doi.org/10.1145/3336191.3371808

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.

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 [16], 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 [29].

(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` [8, 12, 20, 28]. More recently, finer-grained entity detection and typing has received attention [6, 9, 22, 30]. These methods use much larger sets of targets, Ling and Weld for instance 112 types [22]. Similar feature based works are [27, 35, 36]. FINET [9] 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, [24] and [33], on the other hand, use an ILP model to remove noisy types in the final results. While [24] extracts type candidates based on patterns, [33] 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 [6, 10, 30, 33]. The first attempt on using neural networks is by Dong et al. [10]. 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. [30] 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 [34], or utilize various kinds of information from knowledge bases [19]. Recently, Choi et al. [6] 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 [11, 23, 32]. 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) [31], to model and predict evolving relationships [4, 5, 18], or to measure gender bias and discrimination [1].

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 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.

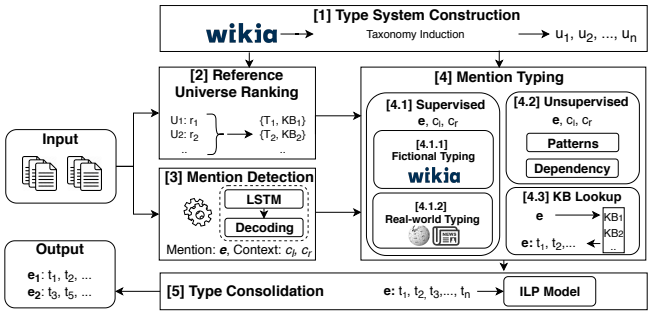

Figure 1: Overview of the architecture of ENTYFI.

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 [15], 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 [6] 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 TYPE SYSTEM CONSTRUCTION

Fiction and fantasy are a core part of human culture, spanning from traditional literature (e.g., books) to various media such as movies, TV series and video games. 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.

Table 1: Example of universes on Wikia.

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

Wikia[1]**.** 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 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 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[2] 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 [7] 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 [7]. In the final type system, the root is `entity`, with two subclasses `physical_entity` and `abstract_entity`. Resulting type systems typically contain between 700 to 10,000 types per universe.

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

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 7.1.

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 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:

$$\hat{y} = \text{argmax}\, f(X, y)$$

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

Inspired by the work in [15] 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 [38], and recurrent dropout to reduce over-fitting [13]. The final score of each label at each position is computed via a softmax layer.

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*
- 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 [28], 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.

7 MENTION TYPING

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

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 [30]. Although LSTMs can encode longer information in sequential data, this is not possible for selective encoding that focuses on local 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 [37]. The model represents the mention and its context separately, before joining them into a final logistic regression layer.

Mention Representation. Averaging of all embeddings of tokens in the mention. Where available, we use precomputed embeddings to represent mentions (300-dimensional GloVe embeddings [25]). 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 [14], 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 [30]). After that, an attention mechanism is used to compute weight factors (i.e. attentions) and integrates them to the output of BiLSTM layers. [17].

7.1.1 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, 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.

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

Table 2: Examples of Hearst-style 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}

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 [6], which is theoretically able to predict up to 10,331 real-world types.

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 [9, 29]. Table 2 shows sample occurrences of these 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`.

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 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.

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 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

$$T_i + T_j \leq 1 \quad \forall (t_i, t_j) \in D$$
$$T_i - T_j \leq 0 \quad \forall (t_i, t_j) \in H$$
$$\sum_i T_i \leq \delta$$

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[3]. For example, Frodo has type candidates `character` (weight 0.6,

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

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).

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 9.2), (2) crowdsourced end-to-end evaluation, on the other hand, takes the input from random texts, and evaluates the results by using crowd-sourcing (Section 9.3). We also examine the performance of each module in our system by doing an ablation study (section 9.4) and finally, testing ENTYFI in unconventional real-world domains (Section 9.5).

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 [7]. 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 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*[4], *Dead or Alive*[5], *Reindeers*[6], *Injustice Fanon*[7] and *Hawaii Five-O*[8]. 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]https://ghostrecon.fandom.com
[5]https://deadoralive.fandom.com
[6]https://reindeers.fandom.com
[7]https://injusticefanon.fandom.com
[8]https://hawaiifiveo.fandom.com

Table 3: Avg. precision, recall and F1 in automated eval.

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**

9.2 Automated End-to-End Evaluation

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

- **NFGEC-WP** [30] 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 [30]. 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** [6] 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 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 [22]. 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 [22]. 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 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 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%

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*[9], 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**[10] 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.

Table 4: Loose- macro and micro precision in crowd. eval.

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**

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.

Results. Table 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.

9.4 Component Evaluation

Type System Construction. Our type system construction uses the technique from [7], 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

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

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 [7]). Table 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.

Table 5: Examples of constructed reference type systems.

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

Mention Detection. In this experiment, we test our mention detection method on the CoNLL-2003 dataset [28], 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 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. 8), supervised fiction typing (SUPWKA - sec. 7.1), supervised real-world typing (SUPWKP - sec. 7.2), unsupervised (UNSUP - sec. 7.3) and KB lookups (sec. 7.4) are disabled. Table 7 shows how these variants perform on the test data. The supervised modules are most important, followed by unsupervised and KB lookups.

Table 6: F1-score of mention detection on CoNLL-2003.

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 7: ENTYFI ablation study – without relaxation.

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**

Anecdotal Examples. Table 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.

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[11], the Viking Age[12] and the Roman Empire[13]. We compare ENTYFI against the best performing baseline, UF-WP.

To evaluate the outputs of these methods, we conducted a crowdsourcing task, similar to Section 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 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

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

Table 8: Anecdotal examples for the outputs of ENTYFI and the baseline.

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~~

macro and micro averaging: 54.02% and 53.98%, compared to 46.47% and 43.70% of UF-WP, respectively.

10 CONCLUSION

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.

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*.
[2] Anne Austin, Jonathan Barnard, Nicola Hutcheon, and David Parry. 2015. Media consumption forecasts 2015. In *Zenith*.
[3] Andrew Carlson, Justin Betteridge, Richard C Wang, Estevam R Hruschka Jr, and Tom M Mitchell. 2010. Coupled semi-supervised learning for information extraction. In *WSDM*.
[4] Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised Learning of Narrative Schemas and their Participants. In *ACL/IJCNLP*.
[5] Snigdha Chaturvedi, Mohit Iyyer, and Hal Daumé III. 2017. Unsupervised Learning of Evolving Relationships Between Literary Characters. In *AAAI*.
[6] Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. In *ACL*.
[7] Cuong Xuan Chu, Simon Razniewski, and Gerhard Weikum. 2019. TiFi: Taxonomy Induction for Fictional Domains. In *The Web Conference*.
[8] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. In *JMLR*.
[9] Luciano del Corro, Abdalghani Abujabal, Rainer Gemulla, and Gerhard Weikum. 2015. Finet: Context-aware fine-grained named entity typing. In *ACL*.
[10] Li Dong, Furu Wei, Hong Sun, Ming Zhou, and Ke Xu. 2015. A hybrid neural model for type classification of entity mentions. In *IJCAI*.
[11] Xishuang Dong, Lijun Qian, Yi Guan, Lei Huang, Qiubin Yu, and Jinfeng Yang. 2016. A multiclass classification method based on deep learning for named entity recognition in electronic medical records. In *NYSDS*.
[12] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *ACL*.
[13] Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *NIPS*.
[14] Alex Graves. 2012. Supervised sequence labelling. In *Supervised sequence labelling with recurrent neural networks*.
[15] Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *ACL*.
[16] Marti A Hearst. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In *COLING*.
[17] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NIPS*.
[18] 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*.
[19] Hailong Jin, Lei Hou, Juanzi Li, and Tiansi Dong. 2018. Attributed and Predictive Entity Embedding for Fine-Grained Entity Typing in Knowledge Bases. In *COLING*.
[20] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *NAACL HLT*.
[21] 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. 2006. Fine-grained named entity recognition using conditional random fields for question answering. In *Asia Information Retrieval Symposium*.
[22] Xiao Ling and Daniel S Weld. 2012. Fine-Grained Entity Recognition.. In *AAAI*.
[23] Zengjian Liu, Ming Yang, Xiaolong Wang, Qingcai Chen, Buzhou Tang, Zhe Wang, and Hua Xu. 2017. Entity recognition from clinical texts via recurrent neural network. In *BMC medical informatics and decision making*.
[24] Ndapandula Nakashole, Tomasz Tylenda, and Gerhard Weikum. 2013. Fine-grained semantic typing of emerging entities. In *ACL*.
[25] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.
[26] Marta Recasens, Marie-Catherine de Marneffe, and Christopher Potts. 2013. The life and death of discourse entities: Identifying singleton mentions. In *NAACL HLT*.
[27] Xiang Ren, Wenqi He, Meng Qu, Lifu Huang, Heng Ji, and Jiawei Han. 2016. Afet: Automatic fine-grained entity typing by hierarchical partial-label embedding. In *EMNLP*.
[28] Erik F Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050* (2003).
[29] Julian Seitner, Christian Bizer, Kai Eckert, Stefano Faralli, Robert Meusel, Heiko Paulheim, and Simone Paolo Ponzetto. 2016. A Large DataBase of Hypernymy Relations Extracted from the Web.. In *LREC*.
[30] Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2017. Neural architectures for fine-grained entity type classification. In *EACL*.
[31] Makarand Tapaswi, Martin Bauml, and Rainer Stiefelhagen. 2015. Book2movie: Aligning video scenes with book chapters. In *CVPR*.
[32] Yonghui Wu, Jun Xu, Min Jiang, Yaoyun Zhang, and Hua Xu. 2015. A study of neural word embeddings for named entity recognition in clinical text. In *AMIA Annual Symposium Proceedings*.
[33] Bo Xu, Zheng Luo, Luyang Huang, Bin Liang, Yanghua Xiao, Deqing Yang, and Wei Wang. 2018. METIC: Multi-Instance Entity Typing from Corpus. In *CIKM*.
[34] Peng Xu and Denilson Barbosa. 2018. Neural fine-grained entity type classification with hierarchy-aware loss. In *NAACL HLT*.
[35] Dani Yogatama, Daniel Gillick, and Nevena Lazic. 2015. Embedding methods for fine grained entity type classification. In *ACL-IJCNLP*.
[36] Mohamed Amir Yosef, Sandro Bauer, Johannes Hoffart, Marc Spaniol, and Gerhard Weikum. 2012. Hyena: Hierarchical type classification for entity names. In *COLING*.
[37] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. 2018. Recent trends in deep learning based natural language processing. In *IEEE CIM*.
[38] Yu Zhang, Guoguo Chen, Dong Yu, Kaisheng Yaco, Sanjeev Khudanpur, and James Glass. 2016. Highway long short-term memory rnns for distant speech recognition. In *ICASSP*.