Automated Taxonomy Induction and its Applications

THÈSE N^{O} 8160 (2017)

PRÉSENTÉE LE 14 DÉCEMBRE 2017
À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE DE SYSTÈMES D'INFORMATION RÉPARTIS
PROGRAMME DOCTORAL EN INFORMATIQUE ET COMMUNICATIONS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Amit GUPTA

acceptée sur proposition du jury:

Prof. P. Dillenbourg, président du jury
Prof. K. Aberer, directeur de thèse
Dr D. Pighin, rapporteur
Prof. F. Suchanek, rapporteur
Prof. R. West, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2017

To my mother, the smartest doctor I know…

Acknowledgements

I would like to express my gratitude towards the people who contributed to this thesis as well as my Ph.D. journey. First and foremost, I would like to thank my supervisor, Prof. Karl Aberer, for providing me with the opportunity to do a Ph.D. at his lab. He helped in creating an ideal environment for me to perform well, by providing me with the freedom and autonomy to follow my own ideas and intuitions. Moreover, he believed in me, even at times, when I didn't believe in myself. His flexibility, constant encouragement, trust, and his unwavering faith in my abilities are the prime reasons for the success of my doctoral studies. Under his supervision, I did some of the most interesting and challenging work I have ever done in my life, and for that, I would always be grateful towards him.

I would also like to thank the rest of my thesis committee, i.e., Prof. Pierre Dillenbourg, Dr. Daniele Pighin, Prof. Fabian Suchanek, and Prof. Robert West, for putting in the efforts to evaluate the merit of my research work as well as provide important and constructive feedback. Special thanks to Prof. Fabian Suchanek for flying in all the way from Paris to Lausanne, solely for my thesis defense.

During the last year of my Ph.D., I met two amazing co-workers Rémi Lebret and Hamza Harkous, who later became my co-authors as well as amazing friends. I owe this Ph.D. to both of them because they arrived at a stage of my life where I was really struggling, and they not only helped me recover but also become super-productive and achieve my full potential. I learnt so much from them in such a short period of time. From Rémi, I learnt the importance of not compromising on the quality of your work, even during difficult times. From Hamza, I learnt that the presentation of your work is as important as the content itself. A million thanks to Hamza for asking literally 5 questions per day over the last year, and a million thanks to Rémi for eating lunch at Le Puur Innovation everyday just because of me, even though it was not your first choice :).

During the four-year stay at LSIR, I met many different people, who played an important role in making my Ph.D. journey memorable. First of all, I thank Chantal for making the administration process so smooth and stress-free. It always amazed me that no matter what I requested for, Chantal could always handle with a very positive and cheering attitude. I also thank my other co-authors Panayiotis and Michele for an interesting collaboration. I would like to thank the other LSIR team members that I interacted with during my stay at LSIR:

Alexandra, Berker, Julien, Jérémie, Julia, Tian, Thanh Tam, Rameez, Matteo, Hao, Jean-Eudes, Mehdi, Jean-Paul, Alex, Tri-Kurniawan, Hung. A special mention to Alexandra and Hung for helping me understand the philosophy of research during the early stages of my doctoral studies.

During the second year of my doctoral studies, I had the opportunity to do an internship at the HEADY research group in Google Zurich. This was one of the most amazing times I had in Switzerland from both a fun as well as work point of view. I would first like to thank my manager at Google, Daniele, for getting me started on the problem of taxonomy induction. Daniele provided me with the freedom to follow my own ideas, but also provided very helpful and constructive feedback at the right time to keep the progress on track. One of the most important things, which I learnt from Daniele, is the importance of solving problems in a more structured fashion, and converting your ideas into well-defined principles. I would like to thank the rest of the contributors from Google Zurich, Francesco Piccinno, Mikhail Kozhevnikov, and Marius Pasça. Special thanks to Marius for taking the time from his busy schedule for further discussions after the internship, which helped me form ideas that served useful for the rest of the thesis. I would also like to thank the rest of the HEADY team, Enrique Alfonseca, Katja Filippova and Aliaksei Severyn, and, a very special thanks to Massimo Nicosia, for being an amazing friend and also the fun times at the foosball and billiard tables.

Next, I thank some of the amazing friends I have made over the years in India as well as Switzerland. First, I would like to thank two very very special friends, Biswanath and Rahul. They were always there to support me and listen to each and every problem I had, no matter how small or big. I know that both of you always have my back, and it gives me a lot of comfort and confidence to have you as my friends. Since I met them in my undergraduate college IIT Bombay, I also take this as an opportunity to thank IIT Bombay, for providing me with a very strong foundation of computer science, which is still serving me until now. I would also like to thank some of the friends I made during my stay at Switzerland: Victor, Catherine, Yan, Adrian, Ashish, Saeid, Evelyne, Damian, Jonas, Valentina. A special mention to my physio and personal trainer, Luke, who not only helped me become more physically strong, but also inspired me to push my will power beyond its limits.

Finally, I would like to thank my family for always being there for me. I would like to thank my parents, especially my mother, for being so loving, caring, and supportive. She has not only always supported me unconditionally, but also encouraged me to follow my heart. A very special mention to my older brother Ankur, who stopped me from giving up so many times during the four years, and was always there standing with me, even during the lowest moments of my life. I would also like to thank my grandfather for instilling the ideas of work ethics and morality in both me and my brother.

Lausanne, 1 Dec 2017 A.G.

Abstract

Machine-readable semantic knowledge in the form of taxonomies (i.e., a collection of *is-a* edges) has proved to be beneficial in an array of Natural Language Processing (NLP) tasks including inference, textual entailment, question answering and information extraction. Such widespread utility of taxonomies has led to multiple large-scale manual efforts towards taxonomy induction such as WordNet and Cyc. However, manual construction of taxonomies is time-intensive, and usually, requires substantial annotation efforts by domain experts. Furthermore, the resulting taxonomies suffer from low coverage and are unavailable for specific domains or languages. Therefore, in recent years, there has been a growing body of work, which aims to induce taxonomies automatically, either from unstructured text or semi-structured collaborative content such as Wikipedia.

In this thesis, we focus on the task of automated taxonomy induction under a variety of different settings. We first focus on the task of inducing taxonomies from Wikipedia, which is the largest and most popular publicly-available semi-structured resource of world knowledge. More specifically, we introduce a set of novel heuristics aimed towards inducing a large-scale taxonomy from the English Wikipedia categories network. We also propose a novel comprehensive path-based evaluation framework for taxonomies. Our experiments show that the taxonomy induced using our approach significantly outperforms the state of the art across edge-based as well as path-based evaluation metrics. Moreover, our experiments also demonstrate that good performance of a taxonomy in traditional edge-based metrics does not always translate to good performance in the path-based metrics.

Subsequently, we focus on the multilingual aspect of taxonomy induction from Wikipedia. We propose a novel approach, which leverages the interlanguage links of Wikipedia to induce taxonomies in other languages. Our approach first constructs training datasets for the *is-a* relation in other languages. Off-the-shelf text classifiers are trained on the constructed datasets and used in an optimal path discovery framework to induce high-precision, wide-coverage taxonomies for all Wikipedia languages. Compared to the state of the art, our approach is simpler, more principled, and results in taxonomies that are significantly more accurate across both edge-based and path-based metrics. A key outcome of our work is the release of our taxonomies across 280 languages, which are significantly more accurate than the state of the art and provide higher coverage.

In the second part of this thesis, we focus on the task of taxonomy induction from unstructured text. We propose a novel approach towards taxonomy induction from an input vocabulary of seed terms that is extracted automatically from raw text. Unlike all previous approaches, which typically extract singular hypernym edges for terms, our approach utilizes a novel probabilistic framework to extract long-range hypernym subsequences. Taxonomy induction from the extracted subsequences is cast as an instance of the minimum-cost flow problem on a carefully designed directed graph. Through experiments, we demonstrate that our approach outperforms the state-of-the-art taxonomy induction approaches across four languages. We also show that our approach is robust to the presence of noise in the input vocabulary. Our approach facilitates the relaxation of many simplifying assumptions, which were employed by previous taxonomy induction approaches, such as clean input vocabularies of seed terms as well as pre-determined sets of roots. As a result, our work serves to automate the process of taxonomy induction from unstructured text in the true sense.

Finally, we introduce a task of discovering and generalizing lexicalized templates from the titles of Wikipedia entities. The experimental results on this task demonstrate that taxonomies, which perform better on our proposed path-based evaluation metrics, result in a more accurate set of generalizations for a given set of entities.

In summary, this thesis proposes new approaches towards automated taxonomy induction. It improves upon the state of the art in a variety of different settings. It also serves to relax many of the simplifying assumptions that limited the applicability of prior approaches.

Keywords: taxonomy induction, knowledge acquisition, natural language processing, Wikipedia, multilinguality, hypernym subsequences, minimum-cost flow optimization, generalization templates, neural networks.

Résumé

La représentation des connaissances sous forme de taxonomies (c.-à-d. une collection de liens est-une) lisible par les machines, s'est avérée bénéfique pour un ensemble de tâches du traitement automatique du langage naturel (TALN), comme par exemple l'inférence, l'implication textuelle, les systèmes questions-réponses et l'extraction d'informations. Une telle utilisation généralisée des taxonomies a conduit à de multiples efforts pour construire manuellement des taxonomies à grande échelle, telles que WordNet et Cyc. Cependant, la construction manuelle des taxonomies prend beaucoup de temps et nécessite généralement des efforts d'annotation importants de la part des experts du domaine. En outre, les taxonomies qui en résultent souffrent d'une couverture faible et ne sont pas disponibles pour des domaines ou des langues spécifiques. C'est pourquoi, au cours des dernières années, un nombre croissant de travaux ont visé à construire automatiquement des taxonomies, soit à partir de textes non structurés, soit à partir de contenus collaboratifs semi-structurés tels que Wikipédia.

Dans cette thèse, nous nous concentrons sur la construction automatique de taxonomies dans différents contextes. Dans un premier temps, nous nous penchons sur la construction de taxonomies à partir de Wikipédia, qui est une ressource semi-structurée des connaissances, la plus grande du monde, la plus populaire et disponible au public. Plus spécifiquement, nous introduisons un nouvel ensemble d'heuristiques visant à construire une taxonomie à grande échelle à partir du réseau de catégories du Wikipédia en anglais. Nous proposons également un nouveau cadre d'évaluation complet des taxonomies fondé sur les séquences. Nos expériences montrent que la taxonomie construite par notre approche surpasse de façon significative l'état de l'art sur les mesures d'évaluation au niveau des liens et des séquences. De plus, nos expériences démontrent que d'obtenir de bonnes performances au niveau des liens n'engendre pas toujours une bonne performance dans les mesures d'évaluation basées sur les séquences.

Par la suite, nous nous concentrons sur l'aspect multilingue de la construction automatique de taxonomies à partir de Wikipédia. Nous proposons une approche novatrice, qui exploite les liens interlangues de Wikipédia pour construire des taxonomies dans d'autres langues. Notre méthode commence par construire des jeux de données d'apprentissage pour la relation est-une dans les autres langues. Des classificateurs de texte standards sont entraînés sur ces jeux de données et sont ensuite utilisés pour la découverte optimale de séquences afin de construire une taxonomie de haute précision et à large couverture dans toutes les langues de

Wikipédia. Comparativement à l'état de l'art, notre approche est plus simple, plus systématique, et produit des taxonomies beaucoup plus précises sur diverses mesures d'évaluation au niveau des liens et des séquences. L'un des principaux résultats de notre travail est la publication de nos taxonomies dans 280 langues, qui sont beaucoup plus précises que l'état de l'art avec une couverture plus élevée.

Dans la deuxième partie de cette thèse, nous nous concentrons sur la tâche de construction de taxonomies à partir de textes non structurés. Nous proposons une nouvelle méthode de construction à partir d'un vocabulaire de termes initiaux, extraits automatiquement du texte brut. Contrairement à toutes les approches précédentes, qui extraient des liens hyperonymiques singuliers pour les termes, nous utilisons un nouveau cadre probabiliste pour trouver de longues sous-séquences d'hyperonymes. La construction de taxonomies à partir des sous-séquences extraites est formulée comme un exemple du problème de flot à coût minimum sur un graphe orienté soigneusement conçu. Au travers d'expériences, nous démontrons que notre méthode surpasse les approches de construction automatique de taxonomies dans quatre langues. Nous montrons également que cette technique est robuste à la présence de bruit dans le vocabulaire d'entrée. Enfin, notre approche facilite l'assouplissement de nombreuses hypothèses simplificatrices, qui ont été utilisées dans le cadre d'approches antérieures de construction de taxonomies, telles que des vocabulaires de termes initiaux non bruités ou des ensembles de racines prédéterminés. Notre travail permet ainsi d'automatiser le processus de construction de taxonomies à partir de textes non structurés au sens propre du terme.

Enfin, nous introduisons une tâche de découverte et de généralisation des modèles lexicalisés à partir des titres des entités de Wikipédia. Les résultats expérimentaux sur cette tâche montrent que les taxonomies, qui donnent les meilleurs résultats sur les mesures d'évaluation proposées au niveau des séquences, permettent d'obtenir un ensemble plus précis de généralisations pour un ensemble donné d'entités.

En résumé, cette thèse propose de nouvelles approches pour la construction automatique de taxonomies. Il améliore l'état de l'art dans une variété de contextes différents. Elle permet également d'assouplir un bon nombre des hypothèses simplificatrices qui ont limité l'applicabilité des approches antérieures.

Mots clés : Construction automatique de taxonomies, acquisition de connaissances, traitement automatique du langage naturel, Wikipédia, multilinguisme, sous-séquences d'hyperonymes, problème du flot de coût minimum, modèles de généralisation, réseaux neuronaux.

Contents

1 Introduction

"There is a set of clear-cut challenges, all centering around knowledge, that have received insufficient attention in AI, and whose solution could bring the realization of Turing's dream – the dream of a machine we can talk with just like a person, and which is therefore (at least) our intellectual equal. These challenges have to do with the representation of linguistically expressible knowledge, the role of knowledge in language understanding, the use of knowledge for several sorts of commonsense reasoning, and knowledge accumulation."

Lenhart K. Schubert [116].

1.1 Overview

The acquisition of machine-readable semantic knowledge has been a fundamental challenge in the field of Artificial Intelligence (AI). The importance of semantic knowledge in building AI, which can achieve human-level performance in complex intelligence tasks, has been continuously accentuated by a variety of different works over the past few decades [80, 81, 116]. Humans acquire and accumulate such knowledge by processing information from a variety of media such as sensory-motor interactions and verbal dialogue [70, 116]. However, the transfer of such knowledge from humans to automated intelligent systems is not straightforward by any means. Due to its inherent complexity, this transfer is usually referred to as the ***knowledge acquisition bottleneck*** [28].

Initial efforts towards loosening this knowledge acquisition bottleneck were mostly manual and involved enormous human efforts aimed towards compiling large-scale knowledge resources [86, 74, 124]. For example, CYC, a comprehensive knowledge base of everyday common sense knowledge, was constructed using a person-century of human effort that involved codifying knowledge into millions of concepts and common sense axioms [74]. However, despite such humongous efforts, the assembled knowledge resources typically suffered from low coverage over specific domains and were usually unavailable for languages other than English. Therefore, in recent years, there has been substantial interest in the acquisition of

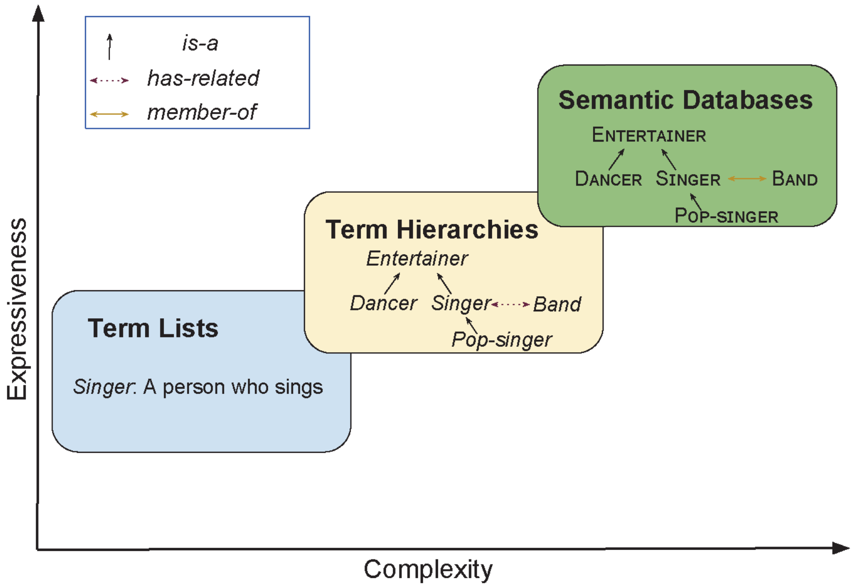


Figure 1.1 – Types of semantic knowledge resources with relative expressiveness and complexity of acquisition [82].

semantic knowledge in a semi-automated or fully-automated fashion [120, 125, 129].

Automatically acquired knowledge resources differ widely in their complexity as well as the expressiveness of the semantics that they encode. They can be loosely categorized into three different categories:

1. ***Term lists*** represent a simple collection of terms, and may possibly contain their definitions or synonyms. A ***term*** is defined as a word or string used to describe a thing or an idea (e.g., *apple*, *singing*, *johnny depp*). Some examples of knowledge resources in this category include dictionaries, vocabularies, glossaries and lexicons.

2. ***Term hierarchies*** specify groupings or classifications of terms or concepts into higher-level generic objects. These groupings typically correspond to two types of semantic relations, i.e., *is-a* and *has-related*. The *is-a* relation asserts that one term is a generalization of another term. For example, the relation *is-a*(*apple*, *fruit*) indicates that *fruit* is a generalization of *apple*. In contrast, the *has-related* relation asserts an associative link between two terms. For example, the terms *singer* and *band* can be grouped by the semantic relation *has-related*. Subject headings and taxonomies are the prime examples of term hierarchies.

3. ***Semantic databases*** are the most complex type of knowledge resources. They employ a fully-structured knowledge model, which is based on concepts rather than terms. A ***concept*** is defined as a thing or an idea, which can be represented by one or more terms. For example, the concept of a SINGER can be represented by multiple terms such as *singer* or *vocalist*.

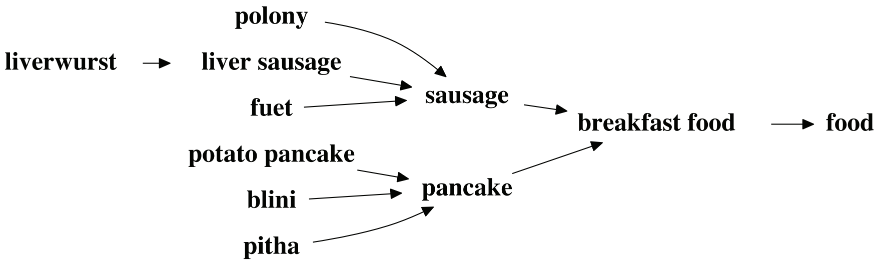


Figure 1.2 – A snippet of the taxonomy of the food domain (Chapter 7). The arrow (→) represents an *is-a* relationship between the two terms.

> Semantic databases encode knowledge in the form of facts, axioms and specific semantic relations between concepts. They also differentiate between concepts that serve as classes (e.g., *fruit*) and their instances (e.g., *apple* and *banana*). In contrast with term hierarchies, the relations in semantic databases are greater in number and more specific in their function. As a result, semantic databases are used in complex information systems such as the Semantic Web [82]. Ontologies are the most prominent examples of semantic databases.

Semantic databases offer the highest expressiveness, followed by the term hierarchies and the term lists respectively. Higher expressiveness of a knowledge resource typically results in greater complexity of its acquisition process. Figure 1.1 illustrates the relationship between the expressiveness of a knowledge resource and the complexity of its acquisition. As shown in the figure, unlike the term hierarchies, the semantic databases explicitly encode the relationship between the concepts SINGER and BAND using the specific semantic relation *member-of*. A more detailed discussion of the different types of knowledge resources can be found in Medelyan et al. [82] and Buitelaar and Magnini [17].

In this thesis, we focus on the automated acquisition of a specific type of term hierarchy, i.e., a ***taxonomy***. In the literal sense, the word “taxonomy” refers to a structure used for the purposes of classification of things or concepts in a particular domain. In the formal sense, a taxonomy is defined as a collection of *is-a* relations between terms or concepts, which represents a complete and coherent tree-like hierarchy. A taxonomy comprising of *is-a* relations on terms is referred to as a *term taxonomy* or a *lexicalized taxonomy*. A taxonomy consisting of *is-a* relations on concepts is referred to as a *concept taxonomy*.

The process of automated acquisition of a taxonomy is referred to as ***automated taxonomy induction***. The induced taxonomies can be either specific to a particular domain (e.g., Sports or Finance), or wide-scale spanning across multiple domains. Figure 1.2 shows a snippet of the term taxonomy of the food domain induced in Chapter 7. Before we proceed with the discussion on automated taxonomy induction, we first describe the *is-a* relation in more detail, as it will serve us for the remainder of this thesis.

1.2 The *Is-A* Relation

The *is-a* relation asserts that one term (or concept) is a generalization of another term (or concept). In the Natural Language Processing (NLP) community, the *is-a* relation is frequently referred to as the ***hypernymy*** relation. The inverse of *is-a* is typically referred to as the ***hyponymy*** relation or the ***specialization*** relation. For example, the semantic relation *is-a*(*apple*, *fruit*) can be equivalently expressed as follows: (1) *fruit* is a hypernym of *apple*, (2) *apple* is a hyponym of *fruit*, (3) *fruit* is a generalization of *apple*, and (4) *apple* is a specialization of *fruit*.

In this thesis, we use *is-a*, hypernymy and generalization interchangeably. We now describe other semantic relations that are closely related to the *is-a* relation:

- ***SubClass-Of:*** the *subclass-of* relation defines a subsumption relationship between two classes (e.g., *subclass-of* (*pop singer*, *singer*)).
- ***Instance-Of:*** the *instance-of* relation defines the relationship between an example instance and its class concept (e.g., *instance-of*(*alicia keys*, *pop singer*)). The class concept is usually referred to as the *type* of the example instance.
- ***Part-of:*** the *part-of* relation (also referred to as the *meronymy* relation) defines the relationship between a constituent part and its whole (e.g., *part-of*(*finger*, *hand*)).
- ***Geo-containment:*** the *geo-containment* relation defines the relationship between a sub-region and its enclosing region (e.g., *geo-containment*(*vaud*, *switzerland*)).
- ***Synonymy:*** the *synonymy* relation defines the relationship between two terms that are similar in meaning (e.g., (*singer*, *vocalist*)).
- ***Co-hyponymy:*** the *co-hyponymy* relation (also referred to as the *sibling* relation) defines the relationship between two terms that generalize to the same class or concept (e.g., *apple* and *orange* are both hyponyms of *fruit*).

In this thesis, we use the definition of the *is-a* relation, which is provided by WordNet [86]. In this definition, the relations *subclass-of* and *instance-of* are considered as valid *is-a* relations. Other relations, i.e., *part-of*, *geo-containment*, *synonymy* and *co-hyponymy* are considered as invalid *is-a* (or *not-is-a*) relations. Additionally, terms or concepts that are either unrelated or do not fall under any of the above semantic relations are also considered to be in *not-is-a* relationship. Table 1.1 shows examples of these semantic relations, and helps to summarize this discussion.

1.3 Automated Taxonomy Induction

We now proceed with the discussion on the automated acquisition of taxonomies, also referred to as automated taxonomy induction. More specifically, in Section 1.3.1, we discuss the utility

is-a	***not-is-a***
iphone→smartphone (*instance-of*)	finger⇝hand (*part-of*)
rose→flower (*instance-of*)	flower⇝plant (*part-of*)
switzerland→country (*instance-of*)	switzerland⇝europe (*geo-containment*)
lausanne→city (*instance-of*)	lausanne⇝vaud (*geo-containment*)
cricket→sport (*instance-of*)	story⇝tale (*synonymy*)
european country→country (*subclass-of*)	singer⇝vocalist (*synonymy*)
singer→artist (*subclass-of*)	apple⇝orange (*co-hyponymy*)
flower→plant organ (*subclass-of*)	johnny depp⇝brad pitt (*co-hyponymy*)
fruit→food (*subclass-of*)	computer⇝yogurt (*unrelated*)
smartphone→electronic device (*subclass-of*)	gas⇝water (*unrelated*)

Table 1.1 – Examples of valid *is-a* and invalid *is-a* (i.e., *not-is-a*) relations. The arrow → represents an *is-a* relationship, whereas ⇝ represents a *not-is-a* relationship.

of taxonomies as well as motivate their automated acquisition. In Section 1.3.2, we provide a brief overview of the main approaches towards automated taxonomy induction and discuss their relative advantages and drawbacks.

1.3.1 Motivation

Intuitively, taxonomies serve to leverage added information in knowledge-intensive tasks. The hyponyms inherit information from their direct as well as ancestor hypernyms, thus eliminating the need to relearn all the relevant information. For example, the fact that "birds fly" can be learned once, and inherited by all the descendant hyponyms of *bird.* Similarly, given the query *bird*, an information retrieval system can also retrieve documents containing the descendant hyponyms of *bird*, hence resulting in a greater recall.

As a matter of fact, taxonomies have been shown to be beneficial in a variety of NLP tasks including information retrieval [21], inference and textual entailment [34, 33, 128], question answering [44, 142], information extraction [12], query understanding [88, 54] and personalized recommendations [145]. Moreover, they have supported numerous practical applications such as information management [96], biomedical systems [67] and e-commerce [1]. A popular real-world example is IBM Watson, a state-of-the-art question answering system, which employs the semantic type information present in taxonomies to restrict the set of answer candidates [29]. In the game show Jeopardy!, IBM Watson consistently outperformed its human opponents at the task of answering general knowledge questions [137].

WordNet is one of the prime examples of lexical knowledge bases that have been utilized for their taxonomic information [86]. WordNet groups English words into sets of synonyms (also referred to as *synsets*), and provides relational information about these synsets such as hypernymy, hyponymy, and meronymy. WordNet has been cited more than 10,000 times in the academic literature and has enjoyed widespread use in a variety of NLP-related and real-world tasks.

However, WordNet is compiled and maintained manually through extensive efforts by domain experts. These manual efforts are extremely time-consuming and do not scale well to the immense range of real-world knowledge. Furthermore, many knowledge domains are dynamic (e.g., Politics or Sports), where new information is produced continuously. Such domains are typically not present in WordNet because inclusion of such domains would require frequent manual updates to maintain the correctness of WordNet. As a result, despite significant efforts, WordNet is still incomplete and provides limited coverage in many domains [103, 51].

Such shortcomings hold true for most manually constructed taxonomies or knowledge resources and have led to a surge of semi-automated and fully-automated approaches towards taxonomy induction in recent years. In the next section, we provide a brief overview of these approaches and discuss their relative advantages and drawbacks.

1.3.2 Main Approaches

Depending on the type of input, approaches towards automated taxonomy induction can be broadly classified into three categories:

1. **Fully-structured resources:** the first line of work aims to automatically expand existing manually-constructed fully-structured knowledge resources such as WordNet or Cyc. These approaches typically consist of two steps: (1) discovery of relevant terms that are missing in the existing resource, (2) appropriate placement of the discovered terms. For example, Widdows [133] places missing terms in the regions of WordNet that contain the most semantically-similar neighbors. Similarly, Snow et al. [120] add missing terms to the WordNet by greedily maximizing the posterior conditional probability of a set of textual evidence.

   Advantages. These approaches typically achieve the highest accuracy, because they use highly accurate resources that are manually compiled by domain experts. Additionally, they also provide a high degree of ***ontologization***, i.e., they contain well-defined concepts and semantic relations. For example, WordNet provides sets of synonyms known as synsets, which are connected to each other in a network of well-defined semantic relations such as hypernymy and meronymy.

   Drawbacks. The main drawback of these approaches is the lack of sufficient coverage. Although multiple approaches were proposed to increase the coverage of such resources by facilitating mass collaboration among the Internet users [113, 130, 131], none of these efforts have achieved any significant advancements towards the aim of providing truly wide-coverage resources.

2. **Semi-structured resources:** the second line of work aims to extract a wide-coverage taxonomy from semi-structured resources such as Flickr [112], Wikitionary [83, 143] or Wikipedia [109, 91, 30, 31, 39, 41]. Unlike fully-structured resources, the content of semi-structured resources is only partially-structured into a fixed set of components

such as Wikipedia pages, categories, Wikitionary definitions or Flickr image tags. Since Wikipedia is by far the largest of these semi-structured resources, the most significant of these approaches are the ones that extract taxonomies from Wikipedia. These approaches benefit from the large scale as well as the semi-structured nature of Wikipedia, which enables the acquisition of highly accurate semantic knowledge through the application of lightweight heuristics. Some of these approaches also combine the taxonomies extracted from Wikipedia with WordNet, thus resulting in wide-coverage as well as high-precision taxonomies [125, 50]. Chapter 2 provides a comprehensive survey of taxonomy induction approaches from Wikipedia.

Advantages. Taxonomies induced from semi-structured resources provide significantly greater coverage than those induced from fully-structured resources, while still maintaining good accuracy [39]. Moreover, Wikipedia, the most popular semi-structured resource used for taxonomy induction, is available in more than 280 languages, thus facilitating the induction of wide-coverage taxonomies in multiple languages [31, 41]. Wikipedia is also continuously updated and maintained by a large number of users in a collaborative fashion, therefore, resulting in the induction of up-to-date taxonomies that provide good accuracy even in highly dynamic domains such as Politics.

Drawbacks. While these approaches provide much greater coverage than fully-structured resources, they still lack coverage in highly specialized or niche domains such as Law or Finance. Furthermore, the growth of Wikipedia has slowed over the recent years, thus making it unlikely that such specialized domains will be covered in the future [126].

3. **Unstructured resources:** finally, the third line of work aims to extract taxonomies from the simplest kind of resource, i.e., unstructured or raw text corpora. This line of work is relatively recent and less studied, and has only received a few publications [68, 129, 5, 102, 40]. Taxonomy induction from unstructured text typically involves two steps: (1) extraction of individual *is-a* relations between terms from unstructured text, (2) the structured organization of terms into a taxonomy using the extracted *is-a* relations. Chapter 5 provides a comprehensive survey of approaches that perform taxonomy induction from unstructured text.

   Advantages. The main advantage of taxonomy induction from unstructured text is that it can be performed on arbitrary domains because domain-specific raw text can be easily harvested on a large scale using the Web [19, 102]. As a result, these approaches typically provide greater coverage than the previous approaches. Furthermore, the temporal information present in most Web documents can be utilized effectively for inducing taxonomies that are up-to-date with the latest information trends present in highly dynamic domains [146, 77].

   Drawbacks. Taxonomy induction from unstructured text is the most challenging of the approaches mentioned above. Therefore, it suffers from multiple drawbacks. First, as expected, the accuracy of taxonomies induced from unstructured text is significantly lower than those induced from fully-structured or semi-structured resources. Second, the taxonomies induced from unstructured text provide a much lower degree of ontologization. More specifically, in contrast to other approaches that produce concept taxonomies, taxonomy induction approaches that utilize unstructured text typically produce term taxonomies. Finally, these approaches typically require a clean vocabulary of terms as input [129, 16]. This requirement is not usually satisfied for most domains, hence leading to a time-consuming step of manual cleaning of vocabularies.

Overall, these taxonomy induction approaches are complementary to each other. Approaches that use fully-structured resources provide better accuracy and a greater degree of ontologization but lower coverage, whereas approaches that use unstructured text provide lower accuracy and poor degree of ontologization but higher coverage. Approaches that use semi-structured resources provide a "sweet spot in the middle", i.e., they provide wide coverage while still maintaining good accuracy and degree of ontologization. All three approaches have been used effectively in NLP-related tasks as well as real-world intelligent applications. A more detailed discussion of these approaches and their use cases can be found in Hovy et al. [52].

1.4 Thesis Objectives

In this thesis, we focus on two of the most widely-used as well as potentially impactful approaches towards automated taxonomy induction: (1) *taxonomy induction from Wikipedia*, and (2) *taxonomy induction from unstructured text*. In each approach, our primary objective is to improve upon the state of the art, resulting in the induction of taxonomies that have higher accuracy and coverage. Furthermore, in each approach, we have specific key objectives, which are described hereafter.

Taxonomy Induction from Wikipedia. The large-scale and high quality of Wikipedia content has enabled multiple approaches towards automated taxonomy induction over the past decade [108, 109, 125, 91, 25, 50, 52, 78, 30, 31, 39, 41]. We propose the following key objectives in taxonomy induction from Wikipedia:

- **Path-level accuracy:** before we proceed with this discussion, we first define the concepts of ***edges*** and ***paths*** in the context of taxonomies. As mentioned in Section 1.1, a taxonomy is defined as a collection of *is-a* relations between terms (or concepts). However, a taxonomy can also be defined as a graph with terms (or concepts) as vertices, and *is-a* relations as *directed edges* between the vertices. A *path* sampled from this graph represents a long-range generalization, which transitively connects specific terms with increasingly more general terms. For example, a taxonomy consisting of the *is-a* edges *apple→fruit* and *fruit→food*, would provide the generalization path *apple→fruit→food*.

  An ideal taxonomy should not only provide accurate *is-a* edges, but also be a good source of accurate generalization paths. While taxonomies induced from Wikipedia have been able to achieve high edge-level accuracy (e.g., 85% from the English Wikipedia [30, 31]), it is not uncommon for their generalization paths to traverse at least one or more of the incorrect edges. Consequently, the resulting taxonomies transitively connect concepts such as *Natural language processing* to many incorrect ancestor concepts such as *Physical body* and *Mass*[1], thus limiting their utility in practice. Moreover, the evaluation of these approaches is strictly limited to edge-level measures, and completely ignores the accuracy or quality of their generalization paths.

Objectives. In light of these shortcomings, the objective of this thesis is two-fold: (1) introduce novel measures for evaluation of taxonomies that take into account the accuracy and quality of their generalization paths, and (2) induce taxonomies that are not only an accurate source of *is-a* edges but also generalization paths.

- **Multilinguality:** while many approaches that focus on the English Wikipedia have been proposed in the past [108, 109, 125, 91, 50, 30, 39], the task of inducing multilingual taxonomies from Wikipedia has received much less attention. A few systems have been proposed including MENTA [25], YAGO3 [78] and MultiWiBi [31]. However, only one of these systems, MultiWiBi, is fully-automated as well as self-contained in Wikipedia, i.e., it does not require any manual labeling or external knowledge resources such as WordNet. MultiWiBi taxonomies suffer from two major drawbacks. First, MultiWiBi taxonomies achieve low accuracy in both edge-level and path-level measures for languages other than English. Second, MultiWiBi taxonomies are generated using a complex set of heuristics that are difficult to replicate.

  Objectives. The objective of this thesis is to propose a novel approach towards inducing multilingual taxonomies from Wikipedia, which significantly improves upon the state of the art in both edge-level as well as path-level accuracy measures. Similar to MultiWiBi, it is desirable that the proposed approach be fully-automated as well as self-contained in Wikipedia. However, unlike MultiWiBi that uses complex heuristics, it is desirable that the proposed approach be simpler, more principled and easy to replicate.

Taxonomy Induction from Unstructured Text. Compared to taxonomy induction from Wikipedia, taxonomy induction from unstructured text is significantly harder. Therefore, taxonomy induction approaches that use unstructured text suffer from multiple shortcomings. We propose the following key objectives to mitigate some of these shortcomings.

- **Accurate hypernymy extraction for general terms:** as mentioned in Section 1.3.2, the first step of taxonomy induction from unstructured text involves the extraction of hypernymy (or *is-a*) relations. In the past literature, this extraction is typically performed using lexico-syntactic patterns [47, 119, 98, 69, 141, 68, 94, 89, 129, 76, 5, 6, 118]. A lexico-syntactic pattern is a generalized linguistic structure that indicates a certain semantic relationship between its placeholder terms. For example, the lexico-syntactic pattern "X

[1] These examples are taken from http://wibitaxonomy.org [30].

is a Y” indicates an *is-a* relationship between the terms X and Y (e.g., “*apple* is a *fruit* that ...”). However, *is-a* extraction based on the lexico-syntactic patterns has a major drawback, i.e., it becomes increasingly erroneous as the generality of terms increases, mainly due to the increase in term ambiguity [129]. For example, the hypernyms for the term *fruit* are likely to be less accurate and more ambiguous than the hypernyms for the term *apple*, because *fruit* is more general than *apple*.

Objectives. One of the principal objectives of this thesis is to mitigate the aforementioned drawback, by utilizing the ancestor hypernyms of specific terms (such as *apple*) to extract more accurate hypernyms for general terms (such as *fruit*).

- **Noisy input vocabulary:** past approaches aimed towards taxonomy induction from unstructured text have a constraint: they require a clean vocabulary of seed terms as input [76, 15, 16]. This constraint is severely limiting because even the most advanced automated vocabulary extraction approaches output vocabularies that contain numerous noisy terms [24]. As a result, a manual cleaning step of such automatically extracted vocabularies is usually required, before the taxonomies can be induced [129]. Although some taxonomy induction approaches do not explicitly state this constraint, they are still either evaluated only with clean vocabularies [68, 5, 102] or use a small-scale automatically-extracted vocabulary, which is unlikely to contain noisy terms [94]. Moreover, none of these approaches are specifically designed from the ground up to handle significant noise in the input vocabulary.

  Objectives. The objective of this thesis is to propose a novel approach towards taxonomy induction from unstructured text, which is robust to the presence of significant noise in the input vocabulary, thus automating the induction process in the true sense.

- **Automated root detection:** taxonomies resemble tree-like hierarchies that are rooted at higher-level terms (or concepts). For example, the taxonomy in Figure 1.2 is rooted at the term *food*. Consequently, taxonomy induction approaches that utilize unstructured text typically assume a set of one or more root terms as input [68, 76, 77, 129, 102]. If such a set is unavailable, some approaches adopt higher-level terms from existing taxonomies (such as WordNet) as input root terms [129]. Although a few approaches are capable of inducing taxonomies without a set of input roots, the final roots of their induced taxonomies are neither evaluated quantitatively nor qualitatively [5, 75].

  Objectives. The objective of this thesis is two-fold: (1) detect roots automatically during taxonomy induction from unstructured text, thus alleviating the requirement of a set of root terms as input, and (2) propose a framework for qualitative as well as quantitative evaluation of automatically-detected roots.

1.5 Thesis Contributions

We now present the main contributions of this thesis. Each of these contributions addresses one or more of the objectives mentioned in the previous section.

- **Taxonomy induction from English Wikipedia:** we propose a novel fully-automated approach towards taxonomy induction from the English Wikipedia. Wikipedia links millions of entities (e.g., JOHNNY DEPP) with thousands of inter-connected categories of different granularity (e.g., AMERICAN MALE FILM ACTORS, AMERICAN FILM PRODUCERS). Our approach exploits the syntactic evidence present in the titles of these categories to connect the Wikipedia entities with increasingly more general categories, hence resulting in a wide-coverage taxonomy.

  Furthermore, we also propose a novel, comprehensive framework for taxonomy evaluation, which focuses on the accuracy and granularity of longer generalization paths, as opposed to individual *is-a* edges. Our experiments demonstrate that our taxonomy provides generalization paths that are more than twice as accurate as the state of the art. Additionally, our taxonomy provides specializations that are more than thrice as accurate as the state of the art. The taxonomy is available at http://headstaxonomy.com.

  This work has been published in the NLP conference COLING'16 (Gupta et al. [39]).

- **Multilingual taxonomy induction from Wikipedia:** we propose a novel fully-automated approach towards inducing multilingual taxonomies from Wikipedia. Given an English taxonomy, our approach first leverages the interlanguage links of Wikipedia to construct training datasets for the *is-a* relation in the target language. Character-level classifiers are trained on the constructed datasets and used in an optimal path discovery framework to induce high-precision, wide-coverage taxonomies in other languages. Our experiments demonstrate that our approach significantly outperforms the state-of-the-art, heuristics-heavy approaches in both edge-level and path-level evaluation measures across six different languages.

  This work is presented in Gupta et al. [41], which is accepted to appear in AAAI'18.

- **Extraction of hypernym subsequences:** we propose a novel probabilistic model that extracts long-range hypernym subsequences such as *apple→tropical fruit→fruit→food* from unstructured text in a fully-unsupervised and automated fashion. Our approach utilizes the hypernyms of specific terms (such as *apple*) to choose more accurate hypernyms for general terms (such as *fruit*). We evaluate our model using both manual and automated evaluation methodologies. Our experiments demonstrate that our model performs favorably against multiple baselines. To the best of our knowledge, this is the first approach that extracts long-range hypernym subsequences from unstructured text.

- **Taxonomy induction using flow network optimization:** we propose a novel approach towards inducing a taxonomy from a collection of potentially-noisy *is-a* edges or subsequences. Our approach casts the task of taxonomy induction as an instance of the minimum-cost flow optimization problem (MCFP) on a carefully-designed flow network. Through experiments, we demonstrate that our approach outperforms state-of-the-art taxonomy induction approaches across four languages. However, more importantly, we also show that our approach is robust to the presence of significant noise in the input vocabulary. To the best of our knowledge, this noise-robustness has not been empirically proven in any previous approach.

The previous two contributions (i.e., extraction of the hypernym subsequences and the flow network framework) have been published in the Databases and Knowledge Management conference CIKM'17 (Gupta et al. [40]).

- **Extensions to the flow network framework:** we propose three extensions to the flow network framework to enhance its capabilities. First, we introduce a parameter that serves to control the *branching factor* of the output taxonomies, where the branching factor is defined as the average number of hypernyms per term. Second, we present two approaches aimed towards automated detection of appropriate roots for a given vocabulary. We evaluate the efficacy of these approaches using artificially constructed vocabularies from WordNet. To the best of our knowledge, this is the first attempt towards automated detection of roots and their evaluation in the context of taxonomy induction from unstructured text. Finally, we demonstrate that the flow network framework enables the automated discovery of new vocabulary terms given an initial seed vocabulary. An interesting outcome of this experiment is the induction of high-quality taxonomies given a single input term such as *cancer* or *fruit* (Chapter 8).

- **Generalization templates:** before we proceed, we first define a ***generalized template*** as a lexicalized linguistic template that contains placeholders, which can be replaced by suitable ***fillers*** to generate titles of entities. For example, the generalization template "Bank of X" can be used to generate the titles of entities such as BANK OF INDIA and BANK OF SCOTLAND using the fillers "India" and "Scotland" respectively.

  We introduce a novel task that aims towards selecting suitable generalizations for the placeholder slot in a generalization template. For example, in the generalization template "Bank of X", the lexical fillers "India" and "Scotland" can be generalized to the higher-order concept COUNTRIES. We propose a novel beam search-based approach that uses a Wikipedia taxonomy to select suitable generalizations for the lexical fillers. Our experiments demonstrate that the generalizations obtained using our English Wikipedia taxonomy are significantly better than those obtained using the state-of-the-art taxonomies. Although in this thesis we focus only on the generalization templates in English, our approach is inherently language-independent and can be replicated easily for other Wikipedia languages.

  A qualitative description of this work is presented in the NLP conference COLING'16 (Gupta et al. [39]).

1.6 Thesis Outline

This thesis is divided into three parts. In the first part of the thesis, we focus on taxonomy induction from Wikipedia. It is structured as follows:

- **Chapter 2, Background and related work:** in this chapter, we describe the various components of Wikipedia. We also provide a comprehensive survey of the state-of-the-art approaches used for taxonomy induction from Wikipedia.
- **Chapter 3, Taxonomy induction from English Wikipedia:** in this chapter, we present our approach towards inducing a large-scale taxonomy from the English version of Wikipedia. We also present our path-based framework for the evaluation of taxonomies.
- **Chapter 4, Multilingual taxonomy induction from Wikipedia:** in this chapter, we present our approach for inducing large-scale taxonomies from Wikipedia in languages other than English.

In the second part of the thesis, we focus on taxonomy induction from unstructured text. It is structured as follows:

- **Chapter 5, Background and related work:** in this chapter, we provide an overview of the state-of-the-art approaches towards taxonomy induction from unstructured text.
- **Chapter 6, Extraction of hypernym subsequences:** in this chapter, we present our approach that extracts hypernym subsequences from unstructured text in a fully-unsupervised fashion.
- **Chapter 7, Taxonomy induction using flow network optimization:** in this chapter, we present our approach that employs flow network optimization to induce a taxonomy from the extracted hypernym subsequences.
- **Chapter 8, Extensions to the flow network framework:** in this chapter, we extend the taxonomy induction approach based on flow network optimization to support the following capabilities: (1) user-defined branching factor for seed terms, (2) automated root detection, and (3) automated expansion of taxonomies by discovery of new vocabulary terms. We also show some examples of the taxonomies, which are induced using our approaches in a variety of settings.

The third part of this thesis, i.e., **Chapter 9, Applications of taxonomies**, focuses on the applications of the induced taxonomies. It provides a brief survey of the state of the art and introduces the task of generalization templates. The last chapter, i.e., **Chapter 10, Conclusion**, concludes the thesis and proposes directions for further research.

Taxonomy Induction from Wikipedia Part I

2 Background and Related Work

2.1 Overview

The large scale as well as the high quality of Wikipedia content has enabled a wide-variety of knowledge acquisition approaches over the recent years, including thesauri extraction [11, 55], taxonomy induction [108, 109, 30, 31, 39, 41] and ontology acquisition [140, 125, 50, 8, 91]. The extracted knowledge has been utilized in many NLP-related tasks including named entity recognition [84, 97], word sense disambiguation [106, 92], computation of semantic similarity between words [123, 107], document clustering [53], question answering [29, 3] and information retrieval [57, 26, 27]. Acquisition of taxonomies from Wikipedia started with the pioneering work of Ponzetto and Strube [108], who demonstrated that a large-scale and high-quality taxonomy could be extracted from Wikipedia using simple heuristics in a fully-automated fashion. The extracted taxonomy achieved performance similar to manually-constructed ontologies (such as WordNet) at the task of computing semantic similarity between words. Since then, a steady body of research has focused on this direction, and a wide of variety of approaches have been proposed.

In this chapter, we provide an overview of these approaches towards taxonomy induction from Wikipedia. However, we first provide a brief introduction of Wikipedia and describe its main components. We also discuss the key advantages offered by Wikipedia, which render it as a particularly favorable candidate for large-scale knowledge acquisition.

2.2 Components of Wikipedia

Wikipedia is a publicly-available online repository of encyclopedic entries, which are commonly referred to as Wikipedia articles. Wikipedia is built and maintained in a collaborative editing framework, which allows any user to edit any article. The collaborative editing framework along with a large number of contributing users has resulted in Wikipedia becoming the largest and the most popular source of reference knowledge on the internet [138]. Wikipedia content is semi-structured, i.e., it is partially structured into a variety of components such as articles, categories, and infoboxes that interact with each other. In this section, we describe a few of these components in detail, which are relevant for the task of taxonomy induction. A larger list of components can be found in Wikipedia [135].

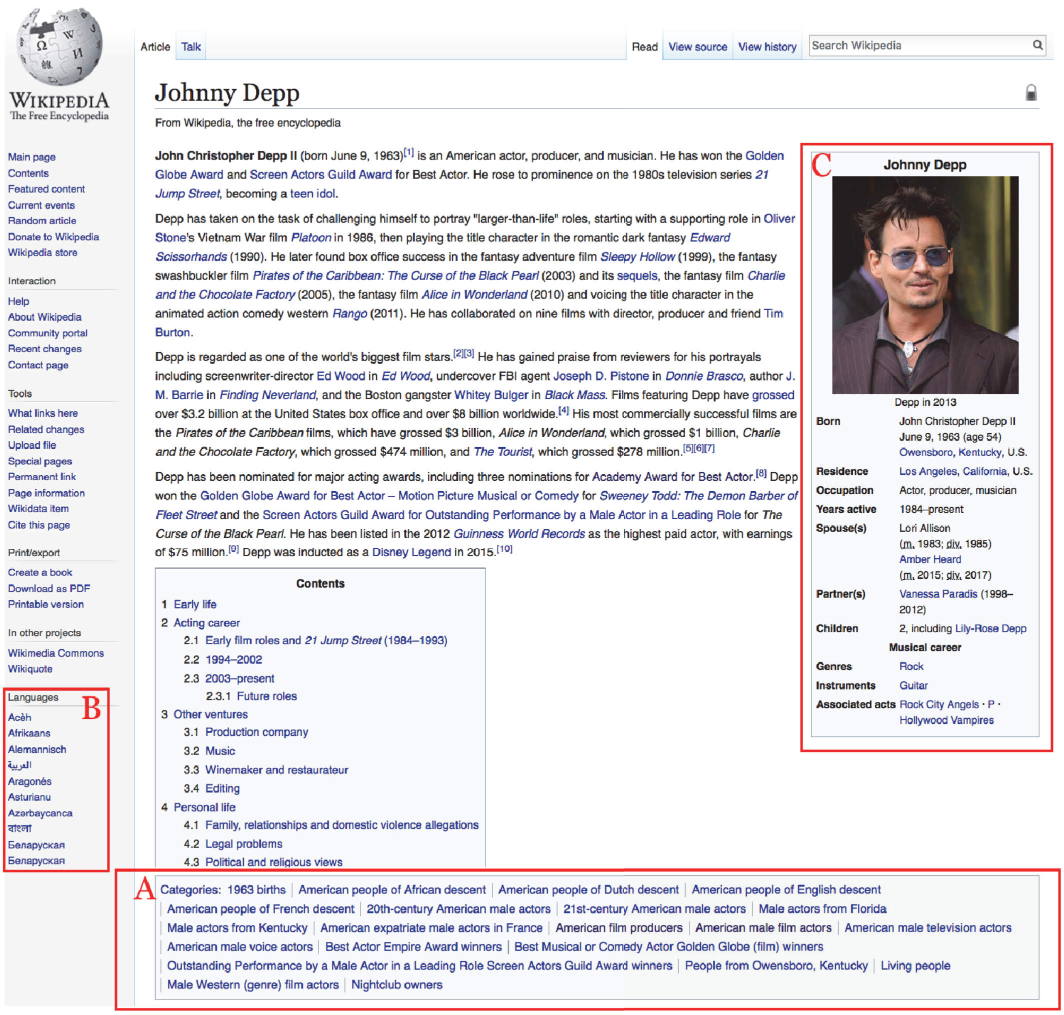
Wikipedia Articles. A Wikipedia article (or page) is an encyclopedic entry about a single concept, where the concept represents a specific sense of a nominal string. For example, the Wikipedia page SWITZERLAND refers to the country sense of the string "Switzerland", whereas SWITZERLAND (SOFTWARE) refers to the software sense. Wikipedia pages frequently refer to entities (e.g., JOHNNY DEPP, LAUSANNE) or real-world concepts (e.g., ACTING, FRUIT). Entities that are homonymous are associated with different pages, which are further disambiguated by a disambiguation string (e.g., BRAD PITT vs. BRAD PITT (boxer)). Since Wikipedia pages frequently refer to entities, the terms Wikipedia entities, Wikipedia pages and Wikipedia articles are used interchangeably in the academic literature.

Wikipedia pages form the largest and the most important component of the knowledge present in Wikipedia. More than 44 million Wikipedia pages are available across 280 different languages [139]. Figure 2.1 shows a condensed version of the English Wikipedia page for the entity JOHNNY DEPP.

Wikipedia Categories. A Wikipedia category groups related pages and categories into broader categories. For example, the Wikipedia page JOHNNY DEPP is categorized into categories such as CATEGORY:AMERICAN MALE FILM ACTORS and CATEGORY:AMERICAN FILM PRODUCERS, whereas CATEGORY:AMERICAN FILM PRODUCERS is further categorized into CATEGORY:FILM PRODUCERS BY NATIONALITY. Categories for the English Wikipedia page JOHNNY DEPP are shown at the bottom in Figure 2.1. Similar to the Wikipedia articles, Wikipedia categories are also collaboratively created and maintained by a large number of contributing users.

Interlanguage Links. Interlanguage links are hyperlinks that connect corresponding pages (or categories) across Wikipedias in different languages. For example, the English Wikipedia page for JOHNNY DEPP is linked to its equivalent versions in 49 different languages including French (JOHNNY DEPP) and Greek (Τζόνι Ντεπ). Two nodes (i.e., pages or categories) linked by an interlanguage link are referred to as ***equivalent*** to each other. Interlanguage links for the English Wikipedia page JOHNNY DEPP are shown at the left side in Figure 2.1.

Internal Hyperlinks. Internal hyperlinks are links embedded in the text of Wikipedia pages, which link to other Wikipedia pages. For example, the English Wikipedia page JOHNNY DEPP links to the Wikipedia pages ACADEMY AWARD FOR BEST ACTOR and HOLLYWOOD. Some examples of these links can be seen in the article text in Figure 2.1.



Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction

Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools

What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export

Create a book
Download as PDF
Printable version

In other projects

Wikimedia Commons
Wikiquote

Languages

Acèh
Afrikaans
Alemannisch
العربية
Aragonés
Asturianu
Azərbaycanca
বাংলা
Беларуская
Беларуская

Article Talk Read View source View history Search Wikipedia

Johnny Depp

From Wikipedia, the free encyclopedia

John Christopher Depp II (born June 9, 1963)[1] is an American actor, producer, and musician. He has won the Golden Globe Award and Screen Actors Guild Award for Best Actor. He rose to prominence on the 1980s television series *21 Jump Street*, becoming a teen idol.

Depp has taken on the task of challenging himself to portray "larger-than-life" roles, starting with a supporting role in Oliver Stone's Vietnam War film *Platoon* in 1986, then playing the title character in the romantic dark fantasy *Edward Scissorhands* (1990). He later found box office success in the fantasy adventure film *Sleepy Hollow* (1999), the fantasy swashbuckler film *Pirates of the Caribbean: The Curse of the Black Pearl* (2003) and its sequels, the fantasy film *Charlie and the Chocolate Factory* (2005), the fantasy film *Alice in Wonderland* (2010) and voicing the title character in the animated action comedy western *Rango* (2011). He has collaborated on nine films with director, producer and friend Tim Burton.

Depp is regarded as one of the world's biggest film stars.[2][3] He has gained praise from reviewers for his portrayals including screenwriter-director Ed Wood in *Ed Wood*, undercover FBI agent Joseph D. Pistone in *Donnie Brasco*, author J. M. Barrie in *Finding Neverland*, and the Boston gangster Whitey Bulger in *Black Mass*. Films featuring Depp have grossed over $3.2 billion at the United States box office and over $8 billion worldwide.[4] His most commercially successful films are the *Pirates of the Caribbean* films, which have grossed $3 billion, *Alice in Wonderland*, which grossed $1 billion, *Charlie and the Chocolate Factory*, which grossed $474 million, and *The Tourist*, which grossed $278 million.[5][6][7]

Depp has been nominated for major acting awards, including three nominations for Academy Award for Best Actor.[8] Depp won the Golden Globe Award for Best Actor – Motion Picture Musical or Comedy for *Sweeney Todd: The Demon Barber of Fleet Street* and the Screen Actors Guild Award for Outstanding Performance by a Male Actor in a Leading Role for *The Curse of the Black Pearl*. He has been listed in the 2012 *Guinness World Records* as the highest paid actor, with earnings of $75 million.[9] Depp was inducted as a Disney Legend in 2015.[10]

Contents

1 Early life
2 Acting career
  2.1 Early film roles and *21 Jump Street* (1984–1993)
  2.2 1994–2002
  2.3 2003–present
    2.3.1 Future roles
3 Other ventures
  3.1 Production company
  3.2 Music
  3.3 Winemaker and restaurateur
  3.4 Editing
4 Personal life
  4.1 Family, relationships and domestic violence allegations
  4.2 Legal problems
  4.3 Political and religious views

Johnny Depp

Depp in 2013

Born	John Christopher Depp II June 9, 1963 (age 54) Owensboro, Kentucky, U.S.
Residence	Los Angeles, California, U.S.
Occupation	Actor, producer, musician
Years active	1984–present
Spouse(s)	Lori Allison (m. 1983; div. 1985) Amber Heard (m. 2015; div. 2017)
Partner(s)	Vanessa Paradis (1998–2012)
Children	2, including Lily-Rose Depp
Musical career	
Genres	Rock
Instruments	Guitar
Associated acts	Rock City Angels · P · Hollywood Vampires

Categories: 1963 births | American people of African descent | American people of Dutch descent | American people of English descent | American people of French descent | 20th-century American male actors | 21st-century American male actors | Male actors from Florida | Male actors from Kentucky | American expatriate male actors in France | American film producers | American male film actors | American male television actors | American male voice actors | Best Actor Empire Award winners | Best Musical or Comedy Actor Golden Globe (film) winners | Outstanding Performance by a Male Actor in a Leading Role Screen Actors Guild Award winners | People from Owensboro, Kentucky | Living people | Male Western (genre) film actors | Nightclub owners

Figure 2.1 – A condensed version of the Wikipedia page for JOHNNY DEPP. The Wikipedia categories are shown at the bottom (A). The interlanguage links are shown at the left (B). The infobox is shown at the right (C).

Infoboxes. Infoboxes are tables that summarize important attributes of the entity referred to by the Wikipedia page. For example, the infobox for JOHNNY DEPP, which is shown on the right side in Figure 2.1, mentions important attributes about JOHNNY DEPP such as date of birth or occupation.

2.3 Wikipedia for Knowledge Acquisition

Wikipedia is the largest and the most popular collaboratively-built repository of encyclopedic knowledge. The collaborative editing framework of Wikipedia and the semi-structured nature of its content enables Wikipedia to alleviate many of the problems faced by the knowledge acquisition approaches that use fully-structured or unstructured resources. Compared to such resources as well as other semi-structured resources, Wikipedia offers many unique advantages, which are described hereafter.

- **Knowledge acquisition bottleneck:** one of the major reasons the knowledge acquisition bottleneck exists (see Section 1.1 for definition) is because a large amount of world knowledge is not explicitly mentioned in textual language. For example, the fact that "birds fly" is not frequently expressed in the unstructured text. However, Wikipedia explicitly encodes such knowledge, because the first lines of Wikipedia pages are usually textual definitions [136, 92]. As a result, Wikipedia aids in mitigating the knowledge acquisition bottleneck. In fact, this fact has already been exploited for knowledge extraction by previous approaches [30].
- **Ontologization:** Wikipedia already provides a high degree of ontologization, because its pages and categories refer to specific, unambiguous concepts or named entities.
- **Semi-structured content:** as discussed in the previous section, Wikipedia content is partially structured into well-defined components such as pages, categories, and infoboxes. This partial structure facilitates the acquisition of semantic knowledge through simple rule-based approaches such as heuristics that exploit the regularities in the structure of Wikipedia content. In fact, the surge of such heuristics-based approaches, chiefly enabled by Wikipedia, has been referred to as "the heuristic renaissance" [52].
- **High quality and large scale:** due to a large number of contributing users, Wikipedia content is generally accurate, large-scale and covers most domains. In fact, it has been estimated that Wikipedia content is the result of a cumulative human effort of approximately 100 million hours, spread across millions of users [134].
- **Dynamism:** Wikipedia content is continuously updated and maintained by the contributing users in a collaborative fashion. Consequently, Wikipedia serves as an up-to-date and accurate source of knowledge even for highly dynamic domains such as Politics or Sports.
- **Multilinguality:** Wikipedia is one of the largest multilingual knowledge repositories ever constructed. Wikipedia is available in more than 280 languages, with at least 13 languages offering more than 1 million articles [138]. Furthermore, many of the Wikipedia pages across different languages are connected by interlanguage links, which can be utilized effectively for tasks such as multilingual taxonomy induction [31, 78, 41] and construction of parallel corpora [2].

Overall, Wikipedia offers some unique features that enable the acquisition of high-quality, large-scale, and multilingual knowledge using relatively simple heuristics-based methods[1]. In the next section, we describe some of these methods that aim towards induction of taxonomies.

[1] A more detailed discussion of this topic can be found in Hovy et al. [52].

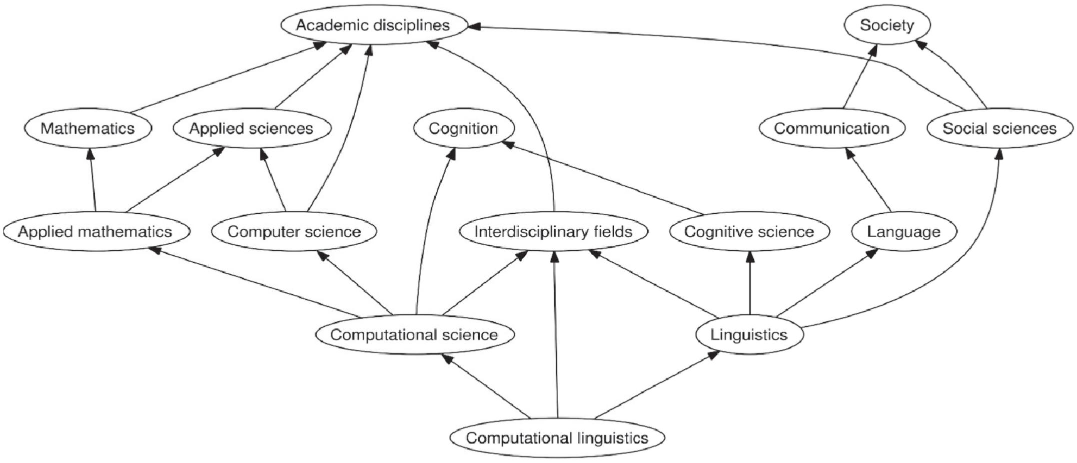


Figure 2.2 – A snippet of the English WCN [52].

2.4 State-of-the-Art Approaches

As discussed in Section 2.2, Wikipedia provides categories, which serve as groupings of Wikipedia pages as well as other related categories. This system of categorization can be converted into a directed graph, which consists of pages and categories as vertices, and the groupings as directed edges. This graph, which represents a semantic network between the pages and the categories, is often referred to as the ***Wikpedia Categories Network*** (hereafter referred to as the ***WCN***). A different WCN exists for each of the languages of Wikipedia. Figure 2.2 shows a snippet of the English WCN.

WCN edges are usually noisy, containing a mix of *is-a* edges (e.g., *Johnny Depp*→*American actors*) and *not-is-a* edges that mainly indicate topic relatedness (e.g., *Johnny Depp*⇝*Hollywood*). Although the WCN can also be used directly in an 'as-is' fashion [107, 52], Ponzetto and Strube [108] demonstrated that the removal of *not-is-a* edges from the WCN results in significantly better performance in tasks such as computing the semantic relatedness between words. The main reason behind this improvement is that the original WCN is overly-connected and noisy. In fact, depending on the language, only 80±5% of the WCN edges indicate correct *is-a* relationships [41]. The remaining edges are *not-is-a* edges, which lead to accumulation of errors during the traversal of WCN, and hence, result in erroneous conclusions such as *Johnny Depp* is a hyponym descendant of *Government*.

Therefore, in the past decade, there has been significant interest in the removal of *not-is-a* edges from WCN. However, selectively discarding the *not-is-a* edges from the WCN, while retaining the *is-a* edges is not trivial by any means, and has been the object of a steady body of research. This process is referred to as taxonomy induction from Wikipedia, and a variety of approaches towards this have been proposed. Some of these approaches are self-contained in Wikipedia, i.e., they rely solely on Wikipedia for taxonomy induction. In contrast, other

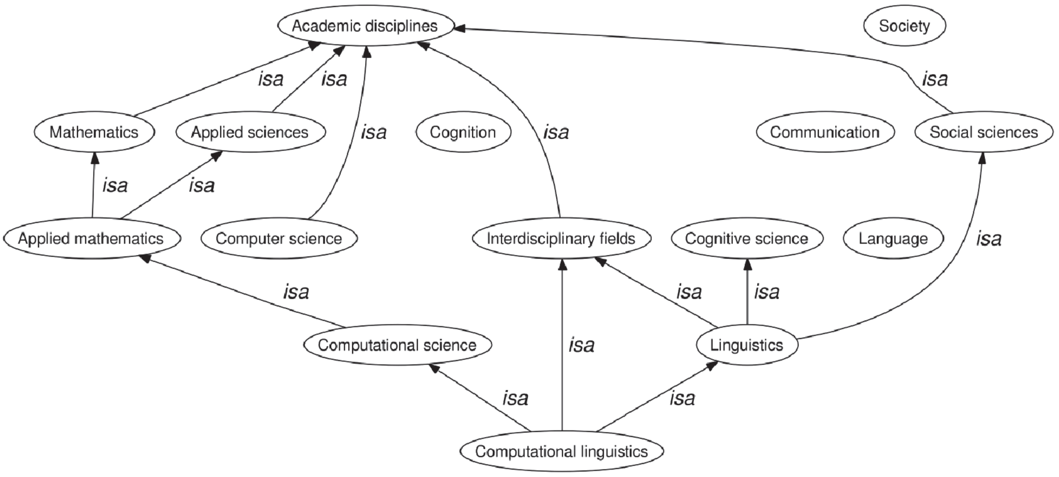


Figure 2.3 – A snippet of the WikiTaxonomy induced from the English WCN [52]. The original WCN corresponding to this snippet is shown in Figure 2.2.

approaches use external resources such as WordNet or other manually-constructed ontologies. We now describe some of these approaches in detail.

2.4.1 WikiTaxonomy

WikiTaxonomy, presented by Ponzetto and Strube [105, 108, 109], is one of the first attempts towards taxonomy induction from Wikipedia. WikiTaxonomy labels the edges of WCN as *is-a* and *not-is-a* using a cascade of heuristics, which utilize the syntactic structure of the category labels, the topology of WCN and lexico-syntactic patterns. For example, the *is-a* edge CATEGORY:AMERICAN FILM ACTORS→CATEGORY:ACTORS BY NATIONALITY is induced by a syntactic analysis of the category names, i.e., by matching the syntactic heads of these categories (i.e., *actor*).

WikiTaxonomy contains more than 100,000 *is-a* relations between pages and categories. It provides better coverage than manually created taxonomies such as WordNet, especially for entity-centric and specialized domains such as Arts and Business. Furthermore, since WikiTaxonomy is extracted using an approach that is self-contained in Wikipedia, it can be easily adapted for other languages such as German [59]. Figure 2.3 show a snippet of the WikiTaxonomy, which is extracted from the snippet of the WCN shown in Figure 2.2.

2.4.2 WikiNet

In contrast with WikiTaxonomy that only extracts a taxonomy, WikiNet aims to extract a full ontology from the WCN [91]. To this end, WikiNet expands the *not-is-a* relations of the WCN into more fine-grained relations such as *part-of* and *located-in*, through a variety of heuristics that exploit the shallow structure of Wikipedia components such as categories and infoboxes. More specifically, the categories-based heuristics of WikiNet utilize the syntactic structure of the category labels, and the topology of the WCN to extract a variety of semantic relations. The infoboxes-based heuristics extract attributes (e.g., place of birth) of entities from their corresponding infoboxes in the Wikipedia pages. The extracted relations are mapped to other languages using the interlanguage links, thus resulting in a multilingual semantic network. Similar to WikiTaxonomy, WikiNet is also self-contained in Wikipedia. WikiNet achieves an accuracy of 76.4% for the *is-a* relation, and up to 95.56% for other relations such as *member-of*.

2.4.3 YAGO

YAGO, an acronym for Yet Another Great Ontology, is a large-scale full-fledged ontology that is derived through the unification of Wikipedia and WordNet [125]. Similar to WikiTaxonomy and WikiNet, YAGO also employs heuristics that exploit the shallow structure of Wikipedia categories and infoboxes to extract semantic relations. However, in contrast with WikiTaxonomy and WikiNet that are self-contained in Wikipedia, YAGO uses the taxonomic hierarchy from WordNet as the source of higher-level hypernyms. More specifically, YAGO connects Wikipedia pages with the synsets in WordNet using two simple heuristics:

- Assign the label *instance-of* (i.e., a valid *is-a* relation, see Section 1.2) to the WCN edges between Wikipedia pages and their parent categories that have a plural lexical head. For example, the edge JOHNNY DEPP→CATEGORY:AMERICAN MALE FILM ACTORS is labeled as *instance-of*, because the lexical head of the string "American male film actors" is "actors", which is plural. The intuition behind this heuristic is that categories that have a plural lexical head are more likely to be genuine classes or collections (e.g., CATEGORY:COUNTRIES) as opposed to entities or instances (e.g., CATEGORY:FRANCE) [136].
- map a WCN category to the WordNet synset, which denotes the most frequent sense of the lexical head of the WCN category. For example, the Wikipedia Category CATEGORY:AMERICAN MALE FILM ACTORS is mapped to the most frequent WordNet synset for the word "actor". The frequencies for the senses are computed using a sense-tagged corpus (i.e., SemCor [87]).

In addition to the above heuristics, YAGO also uses other heuristics to extract implicit relations from the labels of WCN categories. For example, the category CATEGORY:1980 BIRTHS can be used to extract the relation that its descendants were born in 1980. A more detailed discussion of these heuristics can be found in Suchanek et al. [125]. YAGO integrates all extracted relations into a unified knowledge base, which follows the semantics of a formal Semantic Web language (i.e., RDF), and can be accessed by query languages such as SPARQL. YAGO contains more than 1.7 million entities and more than 15 million facts about these entities [140]. YAGO achieves a high accuracy of >95% and has been employed in a wide variety of intelligent applications including IBM Watson [29].

YAGO2. YAGO is further extended to include spatial and temporal knowledge in Hoffart et al. [50]. The resulting knowledge base is referred to as YAGO2. It contains 447 million facts about 9.8 million entities and achieves a high accuracy of 95%.

YAGO3. YAGO3 extends YAGO by combining the information present in WordNet as well as Wikipedias in multiple languages into one coherent knowledge base [78]. Similar to YAGO and YAGO2, YAGO3 achieve a high accuracy of >95% across ten different languages.

2.4.4 DBpedia

Similar to YAGO, DBpedia also aims to extract a fully-structured knowledge base from the semi-structured content of Wikipedia [9]. DBpedia maps Wikipedia entities[2] to a manually constructed coarse-grained ontology of approximately 300 classes. This ontology is collaboratively maintained and contains classes that corresponding to popular entity types such as PERSON and ORGANIZATION. DBpedia employs a cascade of parsers to extract information from different structured components of the Wikipedia pages such as redirects, interlanguage links, categories, and infoboxes. Furthermore, the extracted knowledge is linked with existing knowledge resources such as YAGO, Freebase and Cyc. Similar to YAGO, DBpedia knowledge is also represented using the formal Semantic Web language RDF and can be accessed by SPARQL. Multiple versions of DBpedia have been released over the years [9, 14, 73]. The latest version of DBpedia consists of 1.46 billion facts about 13.7 million entities that are extracted from Wikipedia editions of 111 different languages [73].

2.4.5 MENTA

MENTA is one of the first projects that aimed towards exploiting the multilingual nature of Wikipedia [25]. MENTA integrates Wikipedia pages in multiple languages with WordNet into a single coherent taxonomic hierarchy. To this end, MENTA uses a linker, which links the Wikipedia categories with their equivalent WordNet synsets. The linker uses the Ridge Regression model [13] trained over a small set of manually-labeled examples. The features of the regression model are computed using a variety of information such as the term overlap between Wikipedia categories and WordNet synsets, cosine similarity between the vectors of descriptions of Wikipedia categories and WordNet synsets, and WordNet synsets picked by the most frequent sense heuristic of YAGO. The application of the linker results in a unified graph of Wikipedia categories and WordNet synsets, which is further partitioned to form equivalence classes of entities. A Markov chain-based ranking approach is employed to construct the final taxonomy. At the time of its creation, MENTA was presumably one of the largest multilingual lexical knowledge bases and described 5.4 million entities in more than 270 languages.

[2] As mentioned in Section 2.2, we use Wikipedia pages, articles and entities interchangeably.

2.4.6 MultiWiBi

The Multilingual Wikipedia Bitaxonomy Project (also referred to as MultiWiBi), is the most recent approach towards taxonomy induction from Wikipedia [30, 31]. Similar to WikiTaxonomy and WikiNet, but unlike YAGO, DBpedia and MENTA, MultiWiBi is self-contained in Wikipedia, i.e., it does not require external resources such as WordNet or manually labeled training examples for taxonomy induction. MultiWiBi proceeds in three steps:

- **English bitaxonomy induction**: in the first step, a bitaxonomy, i.e., a separate taxonomy for Wikipedia pages and categories, is induced from the English WCN.
- **Bitaxonomy projection**: in the second step, a cascade of heuristics, which utilize the interlanguage links and the topology of the WCN, is employed to map the taxonomic relations from the English page taxonomy to the pages in a target language such as French or German.
- **Target language bitaxonomy induction**: in the final step, starting from the mapped page taxonomy, a full-fledged large-scale bitaxonomy is induced in the target language.

These steps are fully-automated and language-independent. Consequently, the execution of these steps results in large-scale bitaxonomies for each of the Wikipedia languages. We now describe the three steps in more detail.

English Bitaxonomy Induction. In the first step, MultiWiBi aims to identify lemmas that are good candidate hypernyms for Wikipedia English entities. To this end, it syntactically parses the first line of the Wikipedia pages, because the first line is usually considered to be a textual definition [94]. For example, *actor*, *producer* and *musician* are extracted as candidate hypernymy lemmas for JOHNNY DEPP from the first line of its Wikipedia page (see Figure 2.1). The candidate hypernym lemmas are further disambiguated to Wikipedia entities using a cascade of heuristically-motivated hypernym linkers. For example, the candidate hypernym lemma *actor* is disambiguated to the Wikipedia entity ACTOR. This process results in the extraction of a large number of hypernym edges that connect two Wikipedia entities (e.g., JOHNNY DEPP→ACTOR). These hypernym edges form an initial taxonomy between the Wikipedia pages. It is important to note that due to the requirement of syntactic parsing, this step is language-specific and hence only performed for the English Wikipedia.

In the second step, MultiWiBi utilizes the English page taxonomy to induce a taxonomy over the categories in the English WCN. To this end, it assumes that the generalization information present in the page taxonomy is beneficial for taxonomizing the categories, and vice-versa. More specifically, it assumes that a hypernymy relation is likely between two categories (or pages), if hypernymy relations exists between their corresponding pages (or categories) in the WCN. This idea is presented as ***the bitaxonomy algorithm***, which aims to update the

English lemma	Translations
plane	piano_cartesiano:0.20 piano:0.15 pialla:0.04 aeroplano:0.03 aereo:0.023 piano_astrale:0.02 …
car	automobile:0.33 autovettura:0.11 automobili:0.05 auto:0.02 autovetture:0.01 vettura:0.01 …
key	chiave:0.37 chiavi:0.03 chiave_crittografica:0.001 chiave_segreta:0.0005 …

Figure 2.4 – The English-Italian probabilistic translation table for lemmas extracted from Wikipedia as published by Flati et al. [31]. The numbers indicate the translation probabilities.

category (or page) taxonomy by exploiting the page (or category taxonomy) iteratively. The page taxonomy is initially set to the taxonomy induced in the first step, and the algorithm is run until convergence. As a consequence, MultiWiBi outputs a bitaxonomy, i.e., a separate page taxonomy and a category taxonomy.

Bitaxonomy Projection. This step of MultiWiBi aims to exploit the interlanguage links in Wikipedia to induce a taxonomy in an arbitrary target language (such as French). To this end, it employs a simple rule (hereafter referred to as the ***projection rule***): add a hypernymy edge between two nodes (page or category) in the target language, if a hypernymy edge exists between their English equivalents. For example, the French hypernymy edge Auguste→Empereur romain is induced from the English hypernymy edge Augustus→Roman emperor, and the interlanguage links Augustus↔Auguste, Roman emperor↔Empereur romain.

The application of the projection rule results in the creation of an initial bitaxonomy in the target language. However, this initial bitaxonomy only consists of hypernyms for pages (or categories) that have an English equivalent in the Wikipedia. As a result, they suffer from low coverage. Theoretically, similar to English bitaxonomy induction, a syntactic parser could be used in the target language as well for extracting candidate hypernym lemmas by parsing the first lines of the Wikipedia pages. However, high-quality syntactic parsers are only available for a few languages. Furthermore, their accuracy varies significantly across different languages and is usually lower for non-English languages [71, 72].

Therefore, MultiWiBi compensates for the lack of syntactic parsers in other languages by constructing a probabilistic translation table of lemmas contained in the texts of Wikipedia pages. To this end, it exploits the anchor texts of the internal hyperlinks of Wikipedia. Figure 2.4 shows an excerpt of the English-Italian translation table. This probabilistic translation is constructed for every language and further utilized by heuristics that pick candidate hypernym lemmas for Wikipedia entities in the target language. The exact details of the construction of the probabilistic translation table, as well as the heuristics, are fairly complex and beyond the scope of this thesis. For a full description, we would like to point the readers to the original publication, i.e., Flati et al. [31].

Target Language Bitaxonomy Induction. The application of the above step results in the induction of an initial bitaxonomy as well as a set of translated hypernym lemmas in the target language. Subsequently, the bitaxonomy algorithm is reapplied to produce the final bitaxonomy in the target language.

Comparative Evaluation. A detailed comparative evaluation of MultiWiBi against the state of the art can be found in Flati et al. [31]. We briefly summarize the results as follows. For English, the bitaxonomies induced by MultiWiBi performs favorably compared to the state of the art, achieving higher precision and coverage than most previous approaches. MultiWiBi achieves 90.76% precision over pages and 90.65% precision over categories. MultiWiBi also achieves high coverage, resulting in at least one hypernym for 94.78% of the pages and 98.26% of the categories. While the precision of YAGO is higher than MultiWiBi for English categories (93.58% vs. 90.65%), its coverage is significantly lower (56.74% vs. 98.26%).

MultiWiBi also reports the evaluation results for three other languages, i.e., French, Italian and Spanish. For all three languages, MultiWiBi achieves 80%-85% precision, and 93%-96% coverage. Similar to English, the precision of MultiWiBi taxonomies is slightly lower than YAGO as well as DBpedia, but its coverage is significantly higher.

Despite, achieving slightly lower precision than YAGO and DBpedia, MultiWiBi has its advantages: (1) MultiWiBi achieves significantly higher coverage over both pages and categories than other approaches, thus resulting in a more useful resource. (2) MultiWiBi is the only approach that is language-independent as well as self-contained in Wikipedia. A positive consequence of the language-independence is that MultiWiBi taxonomies are available for all Wikipedia languages.

2.5 Summary

In this chapter, we provided a brief overview of the state of the art of taxonomy induction from Wikipedia. We described the main components of Wikipedia, and also discussed the key advantages that Wikipedia offers over other resources. Finally, we discussed a few of the past approaches aimed towards taxonomy induction from Wikipedia. These approaches differ from each other in a variety of aspects. Some of these approaches aim towards the extraction of taxonomies from Wikipedia (WikiTaxonomy, MENTA, MultiWiBi), whereas others aim towards the extraction of a full-fledged ontology (WikiNet, YAGO, DBpedia). While WikiTaxonomy, WikiNet, and MultiWiBi rely solely on Wikipedia, other approaches use external knowledge resources such as WordNet. However, despite such significant efforts, the taxonomies induced from these approaches still suffer from multiple shortcomings. In the next two chapters, we describe some of these shortcomings and propose yet another approach towards taxonomy induction from Wikipedia, which aims to address these shortcomings.

3 Taxonomy Induction from English Wikipedia

3.1 Overview

In the previous chapter, we described multiple past approaches that have been proposed towards the induction of large-scale taxonomies from Wikipedia. However, despite substantial progress, recent methods still produce taxonomies with glaring gaps in precision and coverage. More importantly, even if the approaches correctly identify individual *is-a* edges with an accuracy as high as 85% (i.e., MultiWiBi [31]), it is not uncommon for long-range generalization paths to traverse at least some incorrect edges. Consequently, the resulting taxonomies transitively connect entities (such as *Natural language processing*) to many ancestor categories (such as *Physical body, Mass*)[1] that are incorrect generalizations, thus limiting the utility of such taxonomies in practice.

In this chapter, we propose a novel approach towards taxonomy induction from the English WCN. Our approach exploits syntactic evidence present in the titles of Wikipedia categories to connect entities (i.e., pages) with increasingly more general categories. Our approach draws inspiration from many of the previous approaches including WikiTaxonomy, WikiNet, YAGO and MultiWiBi (see Chapter 2). However, our approach is the most similar to WikiTaxonomy and MultiWiBi due to two reasons: (1) similar to WikiTaxonomy and MultiWiBi, our approach also aims towards the extraction of a taxonomy rather than a full ontology. (2) similar to these approaches, our approach is also self-contained in Wikipedia, i.e., it does not require additional knowledge resources such as WordNet.

Furthermore, we also propose a novel, comprehensive framework for taxonomy evaluation, which focuses on the accuracy and quality of long-range generalization paths. We perform an in-depth comparison of the taxonomy induced using our approach against the state of the art (i.e., MultiWiBi), and show that our approach results in significant improvements in both edge-level and path-level accuracy measures while maintaining similar coverage.

[1] Examples taken from MultiWiBi (http://wibitaxonomy.org).

Child Node	Parent Categories
Academy Awards	**Category:American film awards** **Category:Awards established in 1929** Category:1929 establishments in California Category:Cinema of Southern California Category:Hollywood history and culture Category:Academy Awards
Category:Film award winners	**Category:Award winners by subject** **Category:Arts award winners** Category:Film people Category:Film awards

Table 3.1 – Examples of parent categories from the English WCN. Categories that are selected as candidate generalizations are shown in bold. Other categories are discarded.

3.2 Our Approach

In this section, we present our approach towards taxonomy induction from the English Wikipedia. Our approach aims to induce a unified taxonomy of pages and categories from the English WCN. To this end, it employs a cascade of linguistically-motivated heuristics. Each of these heuristics exploits the lexical information present in Wikipedia categories to generate a set of candidate generalizations for the WCN nodes (i.e., pages and categories). As an example, Table 3.1 shows two WCN nodes along with their parent categories from the English WCN. Categories that would be selected as the candidate generalizations are shown in bold.

Our heuristics can be grouped into two categories based on the node type: ***category heuristics*** pick candidate generalizations for categories, whereas ***page heuristics*** pick candidate generalizations for pages. Before we present our heuristics, we first specify some concepts and notations that will serve us for the remainder of this section:

- **E**: the set of all English WCN edges.
- $\mathbf{h_c}$: lexical head of the title string of category c. For example, *actors* is the lexical head for the category Category:American male film actors.
- $\mathbf{C_a(n)}$: set of all direct parent categories of a node n (page or category) in WCN, i.e., $C_a(n) = \{c \mid (n, c) \in E\}$. This does not include Wikipedia maintenance categories (e.g., Category:Sports award stubs), which are removed using a handful of blacklisted keywords such as “articles”, “stubs”, “templates”, etc.
- $\mathbf{C_{pl}(n)}$: subset of parent categories ($C_a(n)$), whose titles have a plural lexical head, such as Category:Administrative divisions. As discussed in Section 2.4.3, categories with plural heads have played an important role in prior work on taxonomy induction from Wikipedia because they are more likely to be genuine classes (e.g., Category:Countries) as opposed to individual entities (e.g., Category:France). As a

matter of fact, the Wikipedia guidelines for naming categories also specify that categories that indicate sets of entities should have a plural lexical head [136].

- **$\mathbf{L_p}$**: set of *defining* lemmas attached to the root copular verb in the first sentence of the text of the Wikipedia page p. For example, L_p for JOHNNY DEPP is {*actor, producer, musician*}, as described by its first line (cf. Figure 2.1): *"John Christopher Depp II (born June 9, 1963)[1] is an American actor, producer, and musician."*. As discussed in Section 2.4.6, this construct was first introduced by MultiWiBi [31], who showed that first line of page text can be used for generating candidate hypernym lemmas for the Wikipedia entities.

3.2.1 Category Heuristics

We now describe the category heuristics in detail. For a Wikipedia category c, each category heuristic aims to select zero or more categories that are suitable generalizations of c.

Same Head. Similar to the head-matching heuristic in previous work (i.e., WikiTaxonomy [108]), for a category c, ***same head heuristic*** picks all categories $c' \in C_a(c)$ as the candidate generalizations, which have the same lexical head as c. For example, CATEGORY:AMERICAN ACTORS is picked as candidate generalization for CATEGORY:AMERICAN CHILD ACTORS because they have the same lexical head "actors".

Global Head Support. Most previous approaches, such as WikiTaxonomy and MultiWiBi, augment the same head heuristic with other heuristics that exploit the topology of the WCN. However, we propose a novel high-precision heuristic ***global head support***, which further employs the lexical heads of categories to yield highly-accurate generalization edges between Wikipedia categories.

We first define the ***global support*** ($\sup(h_1, h_2)$) between a pair of lexical heads (h_1, h_2) as the number of edges in E (i.e., the set of all English WCN edges), from a category with lexical head h_1 to a category with lexical head h_2. A higher value of $\sup(h_1, h_2)$ indicates that a category with lexical head h_2 is likely to be a correct generalization for a category with lexical head h_1. Table 3.2 shows a sample of pairs of lexical heads and their global support values.

Given these definitions, for a category c, the global head support heuristic picks the category $c' \in C_{pl}(c)$ with the highest global support $\sup(h_c, h_{c'})$ as the candidate generalization, if $\sup(h_c, h_{c'})$ is above a fixed threshold T_{sup}. In our experiments, $T_{\text{sup}} = 5$ achieved the best results, providing wide coverage while maintaining precision.

We now illustrate this heuristic with an example. Assume that the child category (c) is CATEGORY:ACTORS, which has three direct parents in the original WCN: CATEGORY:ACTING, CATEGORY:ENTERTAINERS and CATEGORY:THEATRICAL OCCUPATIONS. The global head support

Lexical Head (h_1)	**Lexical Head (h_2)**	**Global Support (sup(h_1, h_2))**
actors	actors	8798
	people	1238
	men	142
	entertainers	96
	singers	96
biologists	biologists	199
	scientists	101
	people	11
	oceanographers	11
	scholars	2

Table 3.2 – Pairs of lexical heads and their global supports. Lexical heads with the highest global support for *actors* and *biologists* are shown.

heuristic picks CATEGORY:ENTERTAINERS as the candidate generalization for CATEGORY:ACTORS, because sup(“actors”, “entertainers”) is highest among candidate heads {“acting”, “entertainers”, “occupations”} (as shown in Table 3.2).

Type Similarity. Before we present this heuristic, we first compute vector representations for all the plural lexical heads in the WCN. More specifically, we compute the dimensions of the vector representation for a lexical plural head h as the co-occurrence counts of plural head h with every plural head h' in WCN. The co-occurrence count between two plural heads is defined as the number of pairs of categories with heads h and h' which have at least one common child (page or category). In other words, the co-occurrence count between two plural lexical heads is defined as the number of instances, where categories with these heads are co-parents of a WCN node. The vector representation of the plural lexical head h is referred to as $\mathbf{v_h}$. Using these vector representations, we compute the ***type similarity*** (tsim(h_1, h_2)) between two plural heads h_1 and h_2 as the cosine similarity between $\overrightarrow{v_{h_1}}$ and $\overrightarrow{v_{h_2}}$. Table 3.3 shows the lexical heads with the highest type-similarity for the lexical head *artists*.

Given these definitions, for a category c, the type similarity heuristic picks the category $c' \in C_{pl}(c)$ as the candidate generalization, which has the lexical head h' with the highest type similarity tsim(h, h'), if the similarity is above a fixed threshold T_{tsim}. In our experiments, $T_{\text{tsim}} = 0.2$ achieved the best results.

The global head support and the type similarity heuristics are similar to each other, and only differ in the ranking function used (sup(h_1, h_2) vs. tsim(h_1, h_2)). The global head support heuristic is more precise, whereas the type similarity heuristic has higher coverage, because tsim(h_1, h_2) can be computed even between lexical heads that never co-occur in the WCN.

Lexical Head (h_2)	**Type-similarity (tsim**(h_1, h_2)**)**
watercolorists	0.903
songwriters	0.896
bluesman	0.895
etchers	0.889
animators	0.880
printmakers	0.874
muralists	0.873
parsons	0.865
. . .	

Table 3.3 – Lexical heads with the highest type-similarity for the lexical head (h_1) *artists*.

Only Plural Parent. For a category c, if $C_{pl}(c)$ contains only one category, the ***only plural parent heuristic*** picks it as the candidate generalization. This heuristic follows from the fact that categories that have a plural lexical head typically tend to be set categories [136].

Only Singular Parent. For a category c with a non-plural head h_c, if $C_a(c)$ contains only one category, the ***only singular parent heuristic*** picks it as the candidate generalization. A similar heuristic has been used by MultiWiBi [31].

The previous two heuristics result in the exclusion of the cases when a category with a non-plural head is the only parent of a category with a plural head. The intuition behind this exclusion is that such edges typically tend to be *not-is-a* edges because set categories can be only generalized to other set categories.

Grouping Child Category. Categories with titles matching the pattern **X by Y** (e.g., CATEGORY:ACTORS BY NATIONALITY) usually indicate groupings of instances of *class* X by *attribute* Y [90]. Following this observation, for a category c whose title matches the pattern **X by Y**, the ***grouping child category heuristic*** picks the category with title **X** as the candidate generalization, if it exists in the WCN. For example, using this heuristic, CATEGORY:ACTORS is picked as the candidate generalization for CATEGORY:ACTORS BY NATIONALITY.

Grouping Parent Category. For a category c, the ***grouping parent category heuristic*** picks those categories in $C_{pl}(c)$ as candidate generalizations, whose titles match the pattern **X by Y**. For example, CATEGORY:OCCUPATIONS BY TYPE is picked as the candidate generalization for CATEGORY:LEGAL PROFESSIONS, because CATEGORY:OCCUPATIONS BY TYPE is the direct parent of CATEGORY:LEGAL PROFESSIONS, and the title of CATEGORY:OCCUPATIONS BY TYPE matches the pattern **X by Y**.

Suffix Head. For a category c, the ***suffix head heuristic*** picks all categories $c' \in C_{pl}(c)$ as the candidate generalizations, if their lexical heads $h_{c'}$ are suffixes of h_c. For example, CATEGORY:PEOPLE is picked as the candidate generalization for CATEGORY:SPORTSPEOPLE, because "people" is suffix of "sportspeople".

Lookahead Candidates. For a category c, the ***lookahead candidates heuristic*** picks its grandparents (second-level ancestor categories) as the candidate generalizations, if they satisfy the conditions in the SAME HEAD, GROUPING PARENT CATEGORY or SUFFIX HEAD heuristics. Higher-level ancestors are ignored as they tend to be noisy and introduce semantic drift from the original category.

Title Head. For a category c, the ***title head heuristic*** picks the category with the title h_c as the candidate generalization, if the lemma of h_c is in top T_l% most frequent lemmas among the defining lemmas L_p of the child pages of c. For example, CATEGORY:WRITERS is picked as a candidate generalization for CATEGORY:LEGAL WRITERS, because many child pages of CATEGORY:LEGAL WRITERS have "writer" as a defining lemma. In our experiments, $T_l = 10$ achieved the best results.

3.2.2 Page Heuristics

We now describe the page heuristics in detail. For a Wikipedia page p, each page heuristic aims to pick zero or more suitable generalization categories from its direct parents in the WCN (i.e., $C_a(p)$).

Exact Defining Lemma. For a page p, the ***exact defining lemma heuristic*** picks the category $c \in C_{pl}(p)$ as a candidate generalization, if the lemma of the lexical head of c is present in L_p. For example, all parent categories of page JOHNNY DEPP with the lexical head "actors" are picked as candidate generalizations, because "actor" is present in $L_{\text{JOHNNY DEPP}}$.

Type-similar Lemma. For a page p, the ***type-similar lemma heuristic*** picks a category $c \in C_{pl}(p)$ as the candidate generalization, if the type similarity between the lemmatized lexical head of the category (h_c) and at least one of the defining lemmas in L_p is greater than the fixed threshold T_{tsim}. For example, all parent categories of the page *Johnny Depp* with the lexical head *people* are picked as the candidate generalizations because *actor* is present in $L_{\text{JOHNNY DEPP}}$ and $\text{tsim}\left(\textit{actors, people}\right) > T_{\text{tsim}}$. Similar to the previous section, T_{tsim} is set to 0.2.

Plural Head. Similar to YAGO [125], for a page p, ***plural head heuristic*** picks all categories in $C_{pl}(p)$ as the candidate generalizations.

Child Node	Heuristic	Number of edges	Percent Contribution
Categories	Same head	1,666,049	87.5
	Global head support	153,667	8.1
	Type similarity	10,973	0.5
	Only plural parent	27,916	1.47
	Only singular parent	24,595	1.29
	Lookahead	21,763	1.14
	Grouping child category	< 1000	< 0.05
	Grouping parent category	< 1000	< 0.05
	Suffix head	< 1000	< 0.05
	Title head	< 1000	< 0.05
Pages	Exact defining lemma	5,691,931	51.3
	Type-similar lemma	3,584,712	32.3
	Plural head	1,819,344	16.4

Table 3.4 – Relative contribution of page and category heuristics.

3.2.3 Taxonomy Construction

Up till now, we described the heuristics that are used to pick candidate generalizations for pages and categories. We now describe our approach towards taxonomy construction, which runs in three steps:

1. **Application of heuristics.** The heuristics, which are described in the previous section, are applied to individual pages or categories in the order of decreasing edge-level precision, where precision of the heuristics is computed using a manually-annotated development set. The order of the heuristics is the same, in which they have been presented in Section 3.2.1 & 3.2.2. For each node (i.e., page or category), the process stops when one of the heuristics produces at least one generalization. Subsequently, the remaining heuristics for that node are ignored.

   Table 3.4 shows the relative contributions of each heuristic after this step. For pages, the exact defining lemma heuristic generates the most number of edges followed by the type-similar lemma and plural head heuristics. For categories, the same head heuristic provides the highest number of edges. This result is expected because a large number of lower-level[2] WCN edges have the same lexical heads for both child and parent categories (e.g., CATEGORY:AMERICAN MALE FILM ACTORS→CATEGORY:AMERICAN FILM ACTORS). Furthermore, the WCN is a bottom-heavy graph, i.e., the number of lower-level categories in WCN is significantly higher than the number of higher-level categories, thus justifying the significantly greater contribution by the same head heuristic.

   However, relying solely on the same head heuristic would result in a significantly lower-quality taxonomy, due to the poor coverage at higher-level categories. For example, the node CATEGORY:ACTORS cannot be further generalized using the same head heuristic. In such cases, the other category heuristics play an important role. For example, the

[2] In this context, lower-level means closer to the leaves of the WCN.

global head support heuristic picks CATEGORY:ENTERTAINERS as the generalization for CATEGORY:ACTORS.

2. **Transfer.** Many concepts or entities have both page and category nodes in Wikipedia (e.g., SWITZERLAND and CATEGORY:SWITZERLAND). For such concepts, the generalizations discovered using the category heuristics can be transferred to the pages and vice-versa. To realize this intuition, we first define the concept of equivalency between a page and a category. A page and category are considered to be ***equivalent*** to each other, if they have the same title after the lemmatization of each token. For example, the category CATEGORY:AMERICAN ACTORS and the page AMERICAN ACTOR are considered to be equivalent to each other. If a disambiguation string is specified in the title (e.g., *biology* in FAMILY (BIOLOGY)), it should also match post-lemmatization. For example, the category CATEGORY:FAMILIES (BIOLOGY) is equivalent to the page FAMILY (BIOLOGY), but not to the page FAMILY.

   Given this definition, the pairs of categories and pages that are equivalent are discovered, and the candidate generalizations generated by page (category) heuristics are transferred to the equivalent category (page). This step adds 272,485 generalization edges to the output taxonomy.

3. **Simplification.** Certain Wikipedia categories encode information that is orthogonal to types. For example, CATEGORY:20TH-CENTURY ACTORS refers to time, because it groups actors born in the 20th century. Similarly, CATEGORY:ACTORS FROM SINGAPORE refers to the location and CATEGORY:ACTORS BY NATIONALITY refers to group-by attributes. Such categories are usually redundant, as they represent extra information related to the spatial or temporal domain that is orthogonal to type-based categorization. Therefore, in this step, such categories are detected using a set of hand-crafted regular expressions and eliminated, i.e., their children are linked directly to their parents, and the redundant categories are removed. In total, 65% of the parent categories from the original WCN are identified as redundant and removed. This step is hereafter referred to as ***simplification***.

Figure 3.1 illustrates the taxonomy construction process with an artificial example. Figure 3.1(a) shows the candidate generalizations for the page TOM CRUISE and its parent categories. Different page heuristics (e.g., α_p, β_p and γ_p) are used to propose the candidate generalizations for the page TOM CRUISE. Assuming α_p is ranked higher than β_p and γ_p, generalizations proposed by α_p are retained, whereas others are ignored. Fig. 3.1(b) shows the taxonomy after the application of heuristics, which contains redundant category nodes such as CATEGORY:PEOPLE BY STATUS and CATEGORY:MALE ACTORS FROM NY. Such redundant categories are removed in the process of simplification, resulting in a more compact final taxonomy (Fig. 3.1(c)).

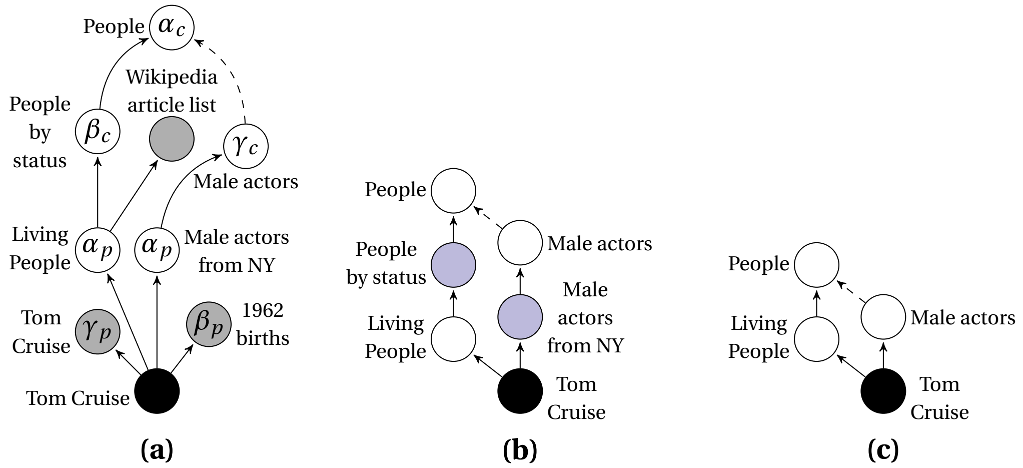


Figure 3.1 – Taxonomy induction phases. Black circles denote entities. White circles denote categories. Dashed lines denote paths including possibly multiple edges. **(a)** Step 1: Page heuristics (α_p, β_p and γ_p) and category heuristics (α_c, β_c and γ_c) are applied sequentially to select candidate generalizations for each node (page or category), until one produces at least one candidate (white circles). Gray nodes show candidates that would have been produced by remaining heuristics. These nodes are ignored. **(b)** Step 2: Initial taxonomy after the application of heuristics. Nodes that encode redundant information are detected (shown in blue). **(c)** Final taxonomy after the removal of the redundant nodes.

3.3 Evaluation and Results

The taxonomy induced after the application of the steps mentioned in the previous section is referred to as the HEADS taxonomy. In this section, we evaluate the HEADS taxonomy against the state-of-the-art taxonomies induced from the English WCN. More specifically, we first present edge-level evaluation using standard metrics such as precision and recall [110, 30]. Further, we demonstrate that, as popular as they might be, such metrics do not reflect the real quality of a taxonomy. We propose a more comprehensive evaluation framework, which takes into account the correctness of multi-edge generalization paths. Our experiments show that performance along these newly-proposed dimensions is not necessarily correlated with the edge-level metrics and cannot be estimated directly from them.

We compare the HEADS taxonomy against the taxonomies released by MultiWiBi, because of two reasons: (1) unlike most other approaches, MultiWiBi and ours is self-contained in Wikipedia. They do not require manually-labeled training examples or external resources, such as WordNet or Wikitionary. (2) MultiWiBi is already shown to outperform most other approaches (see Section 2.4.6).

Experimental Setup. HEADS taxonomy is constructed using a November 2015 snapshot of the English Wikipedia. However, the taxonomies released by MultiWiBi are generated using the October 2012 snapshot. Therefore, to perform a uniform comparison, we initially attempted

Taxonomy	**WiBi$_E$**	**WiBi$_C$**	**Heads**
Nodes	3,414,512	597,179	4,580,662
Entities (E)	3,414,512	-	4,239,486
Categories (C)	-	597,179	341,176
Leaves	3,308,755	465,682	4,359,178
Edges (total)	3,859,717	594,917	11,648,975
$E \rightarrow E$	3,859,717	-	-
$E \rightarrow C$	-	-	11,077,992
$C \rightarrow C$	-	594,917	570,983
Branching factor	1.13	0.996	2.54
WCCs	6,448	2,301	3,195
		Largest WCC	
Nodes	3,386,995 (99.2%)	469,453 (78.6%)	4,563,949 (99.6%)
Edges	3,838,286 (99.4%)	469,453 (78.9%)	11,634,161 (99.9%)

Table 3.5 – Topological properties of the Heads and the MultiWiBi taxonomies. (WCC: weakly connected component).

to re-implement the taxonomy induction approach of MultiWiBi. However, we were unable to replicate the reported results. Moreover, the source code for MultiWiBi was not available publicly and was not shared upon request. Therefore, we instead compared the Heads taxonomy directly against the entity and category taxonomies released by MultiWiBi [31]. These MultiWiBi taxonomies are hereafter referred to as WiBi$_E$ (for entities) and WiBi$_C$ (for categories).

It is important to stress that MultiWiBi taxonomies are generated using an older snapshot of Wikipedia. However, to the best of our knowledge, there is no evidence to suggest that taxonomy induction is easier or harder on more recent vs. older snapshots. In fact, noisy edges between categories such as Japan↝660 BC can be found in both snapshots. Meanwhile, the WCN has grown significantly, with more than twice as many categories (1.37M vs. 619K) and 20% more entities (4.7M vs. 3.8M), therefore, possibly adding to the complexity of the task.

3.3.1 Topological Properties

The main topological properties of the Heads and the MultiWiBi taxonomies are shown in Table 3.5. Heads contains fewer categories and category→category edges than WiBi$_C$, due to the simplification step (see Section 3.2.3), which removes approximately 65% of parent categories from the WCN. Heads covers a larger number of entities than MultiWiBi taxonomies, but a direct comparison of absolute sizes is not necessarily meaningful, since the three taxonomies are defined in different spaces (i.e., WiBi$_E$ has entity→entity edges, WiBi$_C$ has category→category edges, while Heads has entity→category and category→category

***is-a* Edges**	***not-is-a* Edges**
Nidaan→Indian films	Chambezon⇝Geography
Psychiatrists→People	Writing⇝Language
Catte Adams→Singer-songwriter	Jan Ellis⇝Rugby union
SLR cameras→Cameras by type	Visitor attractions in Bonn⇝Bonn

Table 3.6 – Examples of *is-a* and *not-is-a* edges from the gold standard.

edges). In addition, as already mentioned, MultiWiBi taxonomies are generated using an older snapshot of Wikipedia.

As shown in Table 3.5, the largest weakly connected component in HEADS and WIBI$_E$ covers over 99% of the nodes. HEADS has 50% fewer connected components than WIBI$_E$, which is desirable, as each component is an enclave of isolated entities, which cannot be further generalized. WIBI$_C$, which is an order of magnitude smaller than WIBI$_E$ and HEADS, has even fewer connected components, but is overall less connected, with the largest connected component containing only 78% of the nodes.

Finally, Table 3.5 also reports the branching factors of the three taxonomies, where the branching factor is computed as the average out-degree of a node in the taxonomy. The branching factor of HEADS is significantly higher than the branching factor of MultiWiBi taxonomies, which allows it to better account for multiple aspects of a concept or entity, e.g., JOHNNY DEPP is both an *actor* and a *film producer*.

3.3.2 Edge-level Evaluation

We first compare HEADS and MultiWiBi taxonomies using the methodology introduced and consistently followed in prior literature, namely computing the edge-level precision and recall scores against a gold standard [110, 31]. For the construction of gold standard, 500 entities and 500 categories are randomly selected, and their parents in the WCN are annotated by three human judges as *is-a* or *not-is-a* generalizations[3]. Table 3.6 shows some examples of these edges along with their annotations. Precision and recall with respect to the gold standard edges are computed for each sampled node, and then averaged over all the nodes in the gold standard. Table 3.7 shows the *precision* and *recall* scores for HEADS and MultiWiBi taxonomies.

Compared to the MultiWiBi taxonomies, HEADS shows significantly lower precision and recall scores in this evaluation. However, the losses can be largely attributed to two reasons. First, many heuristics in HEADS taxonomy pick candidate generalizations that are not direct parents of the child node in the WCN (e.g., grouping child category heuristic in Section 3.2.1). Such generalizations are missing from the gold standard, and hence considered a loss of precision and recall irrespective of their correctness. Similarly, the simplification step removes many correct but redundant generalizations from the HEADS taxonomy, and replaces them with

[3] The inter-annotator agreement (i.e., Fleiss' Kappa) was 0.52. Annotations were harmonized by majority voting.

Taxonomy	Edge type	P	R	C	A
WCN	$E \rightarrow C$	78.5	100	100	90.2
	$C \rightarrow C$	80.7	100	97.0	84.0
Heads	$E \rightarrow C$	39.4	24.9	89.8	**95.6**
	$C \rightarrow C$	40.5	34.4	24.9	**93.1**
WiBi$_E$	$E \rightarrow E$	**84.1**	79.4	92.6	78.9
WiBi$_C$	$C \rightarrow C$	**85.2**	82.9	97.3	84.0

Table 3.7 – Edge-level evaluation. $E \rightarrow C$ represents entity→category edges, $E \rightarrow E$ represents entity→entity edges, and $C \rightarrow C$ represents category→category edges. MultiWiBi results are as reported by Flati et al. [30]. P: precision, R: recall, C: coverage, A: accuracy.

more compact generalizations. For example, in Figure 3.1, the simplification step replaces the edge Tom Cruise→Male actors from NY with Tom Cruise→Male actors. Such cases result in a loss of both precision and recall, because correct gold standard edges are removed from the Heads taxonomy, and replaced with other correct edges that are not present in the gold standard.

Therefore, due to such issues, Table 3.7 also reports an additional edge-level metric, i.e., *accuracy*. In contrast to precision and recall, which are computed using a gold standard, accuracy is computed by directly annotating the correctness of a random sample of 450 edges from each taxonomy. Formally, accuracy is defined as the ratio of edges annotated as *is-a* over the total number of edges sampled from a taxonomy. As shown in Table 3.7, Heads is more accurate than WiBi$_E$ for entities, though a direct comparison is not meaningful, as WiBi$_E$ contains entity→entity edges and Heads contains entity→category edges. For category→category edges, Heads achieves a fairly significant > 10% improvement in accuracy compared to WiBi$_C$ taxonomy.

Finally, Table 3.7 also reports *coverage*, which is defined as the fraction of entities and categories in a taxonomy with at least one generalization, independent of its correctness. Heads shows lower coverage on categories because 65% of categories in the WCN are filtered out due to the simplification procedure.

3.3.3 Path-accuracy Evaluation

Motivation. Good performance at edge-level, though widely used as an indicator of quality of a taxonomy [108, 90, 30], does not automatically translate into good performance at path level. For example, the generalization path *apples→fruits⇝vegetarians→people→organisms* is 75% edge-accurate (i.e., 3/4 edges are correct as indicated by the symbol →), but it can lead to the wrong inference that *apples* are *vegetarians* and, in turn, *people* and *organisms*. In fact, a single incorrect edge, i.e., *fruits⇝vegetarians*, causes a cascade of generalization errors for *fruits* and all its descendants, and a cascade of specialization errors for *vegetarians* and all its ancestors. Moreover, the addition of another correct edge *organisms→things* would increase

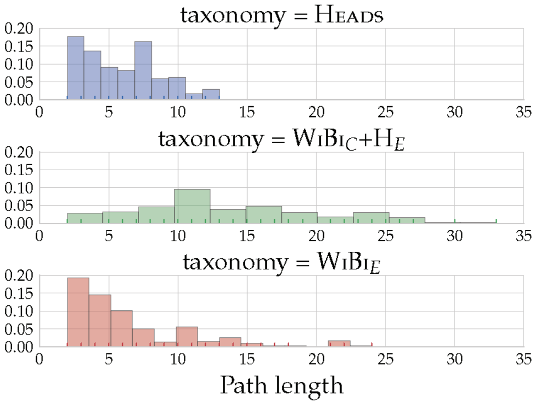


Figure 3.2 – Length distribution of the generalization paths sampled from HEADS and the MultiWiBi taxonomies.

the edge-level accuracy, but would not affect the path-level accuracy.

The above example suggests that edge-level performance of a taxonomy may not always correlate with path-level performance. Therefore, as an alternative to the standard edge-level evaluation, we propose a structured path-based framework for taxonomy evaluation. More specifically, our framework seeks to answer the following questions about a taxonomy:

1. What is the accuracy of multi-edge generalization paths?
2. Are individual generalizations at the right level of granularity?
3. What is the accuracy of the specializations of a node?

Evaluation Framework. The previous example (i.e., *apples⇢organisms*) demonstrates that during traversal of an upward generalization path, the correctness of individual edges is inconsequential to finding a good generalization for the starting node (i.e., *apples*) once the first wrong edge (*fruits⇝vegetarians*) is encountered. Therefore, an ideal taxonomy should not only provide a large proportion of correct edges, but also provide correct generalization paths, i.e., paths which are correct in their entirety. However, in practice, it is common for relatively deep taxonomies to provide long generalization paths, which pick at least one wrong generalization edge. In such cases, it is still desirable to have a long *correct path prefix*, i.e., the maximal prefix of a path which is correct in its entirety.

In this section, we evaluate HEADS and the MultiWiBi taxonomies on their ability to provide paths with longer correct path prefixes. To avoid bias, it is desirable that paths sampled from different taxonomies start from the same starting entities. Therefore, WIBI$_C$, which lacks the notion of entities, is first augmented with Entity→Category edges from HEADS, resulting in a new hybrid taxonomy. This hybrid taxonomy is hereafter referred to as WIBI$_C$+H$_E$.

WiBi$_E$	WiBi$_C$+H$_E$	Heads
Structure	Government	
↑Algebraic structure	... 23 more categories ...	**Apes**
↑Category (mathematics)	↑Cinema by region	↑**Humans**
↑Sequence	↑Cinema by continent	↑**People**
↑Process (science)	↑North American cinema	↑**Producers**
↑Filmmaking	↑Cinema of the United States	↑**American producers**
↑**Film producer**	↑**American film producers**	↑**American film producers**
Johnny Depp	**Johnny Depp**	**Johnny Depp**

Table 3.8 – Upward generalization paths for Johnny Depp in three taxonomies. Correct path prefixes are shown in bold.

WiBi$_E$	WiBi$_C$+H$_E$	Heads
Law	Concepts by field	
↑Principle	↑... 9 more categories ...	
↑Process (philosophy)	↑Computer programming	↑Systems
↑**Abstraction (computer science)**	↑Debugging	↑Operating systems
↑**Software framework**	↑**Debuggers**	↑**Linux kernel features**
DTrace	**DTrace**	↑**DTrace**

Table 3.9 – Upward generalization paths for DTrace in three taxonomies. Correct path prefixes are shown in bold.

For evaluation, we first sample a set of 250 entities that are present in all three taxonomies (i.e., Heads, WiBi$_E$ and WiBi$_C$+H$_E$). Further, we sample one generalization path for each (entity, taxonomy) pair, thus resulting in a total of 750 paths. Figure 3.2 shows the length distribution of the generalization paths sampled from each taxonomy. As expected, paths sampled from the Heads taxonomy are shorter than MultiWiBi taxonomies due to the simplification step (cf. Section 3.2.3). To compare the quality of these generalization paths, three human annotators inspect each path starting from the entity and annotate[4] the first incorrect generalization, thus marking their correct path prefixes. Table 3.8 and 3.9 shows some examples of these sampled paths along with their correct path prefixes.

We report two path-accuracy metrics: (1) the average length of CPP, which is hereafter referred to as ***ACPP***, and (2) the average ratio of lengths of CPPs to the full paths, which is referred to as ***ARCPP***. As an example, for the generalization path *apple→fruit⇝farmer→human→animal* with the *not-is-a* edge *fruit⇝farmer*, the path length is 5, length of CPP is 2, and ratio of length of CPP to total path is 0.4 (i.e., $\frac{2}{5}$).

Intuitively, ACPP indicates the average number of upward generalization edges that can be traversed in a generalization path sampled until the first wrong generalization edge is encountered. Similarly, ARCPP indicates the average fraction of a generalization path that

[4] At least two annotators agreed for 93% of paths. All three annotators agreed for 53% of paths. Annotations are harmonized using majority voting.

Method	AL	ACPP	ARCPP
WiBi$_E$	6.37	2.47	0.53
WiBi$_C$+H$_E$	13.57	3.59	0.34
Heads	6.00	**4.99**	**0.87**

Table 3.10 – Comparison of average path length (AL), average length of correct path prefix (ACPP), and average ratio of CPP to path lengths (ARCPP) for Heads and the MultiWiBi taxonomies.

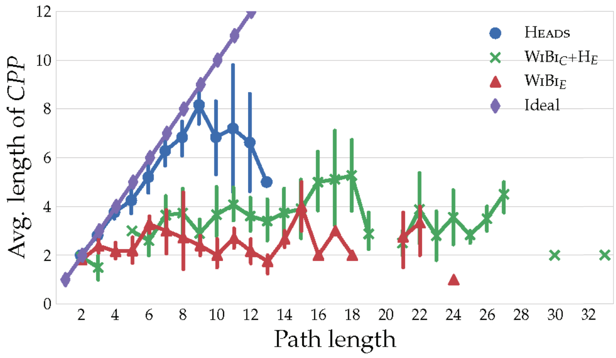


Figure 3.3 – Average length of correct path prefix (i.e., ACPP) in different taxonomies (computed using the set of 750 annotated paths).

can be traversed until the first wrong generalization edge is encountered. ARCPP is useful for comparing taxonomies that have significantly different average path lengths.

Table 3.10 shows the ACPP and ARCPP values for Heads and the MultiWiBi taxonomies. Heads significantly outperforms both WiBi$_E$ as well as WiBi$_C$+H$_E$, achieving both higher ACPP as well as ARCPP. Figure 3.3 plots ACPP against total path lengths for Heads and MultiWiBi taxonomies, along with their 95% confidence interval bars[5]. For a correct generalization path, the length of the correct path prefix is the same as the path length. Therefore, an ideal taxonomy with only correct generalization paths would show up as the line ACPP = Path length in Figure 3.3. The behavior of Heads taxonomy is very close to an ideal taxonomy for the majority of path lengths, and outperforms WiBi$_E$ or WiBi$_C$+H$_E$ at all lengths.

It is interesting to note that this difference does not translate into similar differences in the edge-level evaluation, where all taxonomies consistently show relatively high accuracy (cf. Section 3.3.2). The superior performance of Heads is further confirmed by the results in Figure 3.4, which plots the probability of obtaining a correct generalization path of length up to k against k. Heads paths are more than twice as accurate as WiBi$_E$, and thrice as accurate as WiBi$_C$+H$_E$. Furthermore, in contrast with MultiWiBi taxonomies, Heads generalization paths maintain high probability of correctness (> 0.7) at all lengths.

[5] The confidence intervals reflect the distribution of the paths being sampled. A larger confidence bar indicates lower probability that a path of that length is present in the annotated samples.

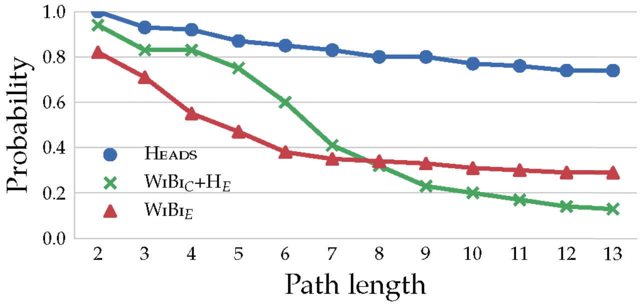


Figure 3.4 – Probability of correct generalization paths vs. length (computed using the set of 750 annotated paths). The probability at length k is the ratio of correct paths of length $\leq k$ to the total number of paths of length $\leq k$. Paths with length > 13 are omitted, as they are not present in HEADS samples, and always incorrect in WIBI$_E$ and WIBI$_C$+H$_E$ samples.

WIBI$_C$+H$_E$	**Label**		**HEADS**
Physical systems	2		
Technology systems ↑	1		
Technology by type ↑	2		
Transport ↑	1	2	Physical systems
Transport by mode ↑	2	2	↑Technology systems
Water transport ↑	0	2	↑Transport systems
Watercraft ↑	2	2	↑Vehicles
Ships ↑	2	2	↑Ships
USS Calhoun (1851) ↑	2	2	↑USS Calhoun (1851)

Table 3.11 – Example of the generalization paths sampled from HEADS and the MultiWiBi taxonomies, along with their respective annotations. The source entity is *USS Calhoun (1851)* and the destination category is *Physical systems*.

3.3.4 Path-granularity Evaluation

A good taxonomy should not only provide correct generalization paths but also ensure that each edge in the path provides generalization at the right level of granularity, i.e., neither too specific nor too general. To evaluate this aspect, we sample 100 generalization paths originating from the same starting entities from each taxonomy. For each path, each individual edge is annotated by three human annotators with one of the following labels: 0 for wrong generalization (e.g., *fruits⇝vegetarians*); 1 for under-generalization (e.g., *fruits by country→fruits*); 2 for good-generalization (e.g., *edible fruits→fruits*); 3 for over-generalization (e.g., *edible fruits→physical bodies*). An edge under-generalizes if it adds or removes little information relative to the source node (e.g. *cricketers by team→cricketers*) or if it is a synonym or rephrasing of the original category (e.g. *coaches by sport→sport coaches*). An edge over-generalizes if it removes too much information (e.g., *edible fruits→physical bodies*).

To ensure that the paths, which are compared, are similar in length and complexity across different taxonomies, we only consider pairs of shortest paths $\langle p_1, p_2 \rangle$ with the same starting

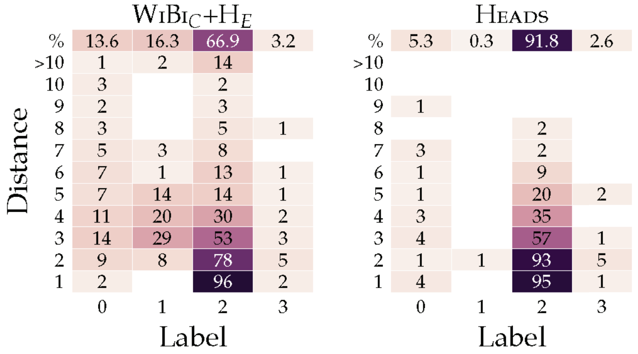


Figure 3.5 – Generalization granularity evaluation for HEADS and WIBI$_C$+H$_E$ using a set of 100 generalization paths. Labels on the horizontal axis indicate generalization granularity: 0 (wrong generalization), 1 (under-generalization), 2 (good-generalization) and 3 (over-generalization). Top row shows overall distribution of labels. Other rows represent number of sampled paths, which have an edge with the corresponding label at the given distance from starting node.

and ending nodes in the taxonomies. Furthermore, the paths are selected such that the difference in the length of the shortest paths, i.e., $||p_1|-|p_2||$, is minimal[6]. An example of such pair of paths along with their annotations is shown in Table 3.11.

WIBI$_E$ is excluded from this experiment, because in contrast to HEADS and WIBI$_C$+H$_E$, WIBI$_E$ does not contain categories. Therefore, for WIBI$_E$, the condition of the same final node cannot be satisfied. Figure 3.5 graphically summarizes the results of this experiment. In general, HEADS outperforms WIBI$_C$+H$_E$, achieving significantly higher percentages of good-generalizations (91.8% vs. 66.9%). HEADS also has fewer under-generalizations than WIBI$_C$+H$_E$ (0.3% vs 16.3%), which can be largely attributed to the simplification step that removes redundant categories (cf. Section 3.2.3). However, it is interesting to note that despite the removal of 65% of categories through simplification, HEADS still does not suffer significantly from over-generalizations.

3.3.5 Evaluation of Specializations

A good taxonomy should not only provide accurate generalizations going upwards in the taxonomy but also provide accurate specializations going downwards. Therefore, in this section, we evaluate HEADS and MultiWiBi taxonomies on the quality of their specializations. To this end, three human annotators annotate the correctness of a sample of descendants of higher-level nodes in the taxonomies WIBI$_E$, WIBI$_C$ and HEADS. To avoid bias, the higher-level nodes (entities for WIBI$_E$; categories for WIBI$_C$, HEADS) are sorted in decreasing order of the number of descendants in the respective taxonomies. Ten nodes at fixed ranks (5, 10, .., 50) from each list are selected for evaluation. To enable a comparison of WIBI$_E$ with WIBI$_C$ and

[6] It is ensured that the paths are not identical (i.e., $p_1 \neq p_2$).

Ancestor	Descendant	Annotation
CATEGORY:PLACES	CATEGORY:EDUCATION IN MANCHESTER, CONNECTICUT	0
CATEGORY:PLACES	CATEGORY:FLORENCE CATHEDRAL	1
CATEGORY:PLACES	CATEGORY:FLUVANNA COUNTY, VIRGINIA	1
CATEGORY:POLITICIANS	CATEGORY:DEFENCE MINISTRIES	0
CATEGORY:POLITICIANS	CATEGORY:DISTRICT ATTORNEYS	1

Table 3.12 – Examples of (ancestor, descendant) pairs and their annotations. 0 indicates an incorrect specialization, whereas 1 indicates a correct specialization.

Taxonomy	Overall accuracy	Per-node accuracy
WIBI$_E$	24.3	23.0
HEADS (entity)	**70.3**	**72.7**
WIBI$_C$	38.1	40.8
HEADS (category)	**67.0**	**72.5**

Table 3.13 – Accuracy of specializations. Results for entity and category descendants of HEADS are reported separately.

HEADS, category nodes are manually mapped to equivalent entity nodes and vice-versa (e.g., *Category:Places* is mapped to the entity *Place*). The annotators judge the correctness of 10 randomly sampled descendants for each selected node in each of the three taxonomies.

Table 3.12 shows some examples of the sampled pairs along with their annotations. Table 3.13 shows the results of this experiment. Both overall and per-node accuracy are reported. Overall accuracy is defined as the fraction of sampled (node, descendants) pairs that are correct, whereas per-node accuracy is defined as the average ratio of correct descendants per node. Results for entity and category descendants of HEADS are reported separately. The results demonstrate that the descendants provided by HEADS are almost three times as accurate as WIBI$_E$, and almost twice as accurate as WIBI$_C$.

3.4 Discussion and Related Work

In the previous chapter (Section 2.3), we discussed the unique advantages offered by Wikipedia that enable the acquisition of high-quality semantic knowledge on a large scale using relatively simple rule-based methods. Our approach serves to demonstrate some of these advantages in a practical manner. For instance, the heuristics employed by our approach are effective, chiefly because Wikipedia content is already meaningfully-structured into pages and categories. Furthermore, due to the Wikipedia guidelines [136], the titles of the categories follow regular syntactic patterns, which allow our heuristics to make simplifying assumptions that hold true in most of the cases. Such factors enable our approach to acquire a wide-coverage high-quality taxonomy using simple rule-based heuristics.

Our approach draws inspiration from many of the previous approaches towards taxonomy induction from Wikipedia (see Section 2.4 for a survey). Similar to most of the prior work, our approach also aims to discard the WCN edges that are least likely to represent *is-a* relations. Similarly, like most previous approaches, our approach is also heuristic-driven. Furthermore, similar to WikiNet and YAGO, our approach also aims to exploit the syntactic structure of the categories to choose suitable generalizations for WCN Nodes.

The first main contribution of our work is the introduction of novel heuristics. While a few of our heuristics are adapted from previous work (as cited in Sections 3.2.1& 3.2.2), the rest of our heuristics are introduced for the first time. The most important of these novel heuristics is the global support heuristic, which enables the extraction of a large number of high-quality *is-a* edges (Table 3.4). Our second contribution is the path-based framework, which proposes the measures ACPP and ARCPP (Section 3.3.3) for evaluating the quality of a taxonomy. Experimental results using the path-based framework demonstrate that performance of a taxonomy on edge-level may not be correlated with the performance on the path-level. Indeed, HEADS taxonomy achieves seemingly similar performance to MultiWiBi on edge-level metrics, but significantly outperforms the MultiWiBi taxonomies on path-level metrics (Section 3.3.3 & 3.3.4). Furthermore, HEADS taxonomy is a also significantly more accurate source of specializations (Section 3.3.5). A key outcome of our work is the release of HEADS taxonomy, which is publicly available at http://www.headstaxonomy.com. Figure 3.6 (page 49) shows a snippet of the HEADS taxonomy.

During the course of this work, we also experimented with a few variants of our approach that did not produce optimal results. For example, we trained a SVM classifier using the outputs of the category (or page) heuristics as features, and a small manually-annotated set of edges as training data. While the edge-level accuracy of this approach was similar to HEADS, it suffered from poor path-level performance. The primary reason for this effect was the difficulty to choose an appropriate classification threshold that would result in an appropriate level of generality of the taxonomy roots. We also experimented with ancestor-level versions of many of the heuristics presented in Section 3.2.1. However, we discarded them as they typically produced noisy generalizations. Finally, we also experimented with the taxonomy prior to the simplification step (Section 3.2.3). While the pre-simplification taxonomy produced good results as well, we introduced the simplification step for mainly two reasons: (1) produce a more compact and type-oriented taxonomy, (2) reduce the effort of annotations.

3.5 Summary

Whether built from scratch or derived by filtering existing data, automatically-constructed taxonomies are accurate and useful only to the extent that they correctly assert not only short-range but also longer-range generalizations or specializations among concepts or entities. In this chapter, we presented a novel approach towards taxonomy induction from the English Wikipedia categories network. Similar to previous approaches, our approach also employs a set of heuristics to distill a unified taxonomy of pages and categories. However, our experiments show that our high-precision heuristics result in a taxonomy, which is significantly better than the state of the art in edge-level accuracy as well as a variety of path-level evaluation metrics.

Implications. The work done in this chapter has multiple implications on the rest of this thesis. First, the HEADS taxonomy is used in the next chapter to induce high-quality, large-scale taxonomies for other Wikipedia languages such as French. Second, some of the ideas formulated during this work are used towards taxonomy induction from unstructured text (Chapter 6). Finally, in Chapter 9, the task of generalizing linguistic templates demonstrates that the higher path-level accuracy of HEADS taxonomy leads to significantly better generalizations than the MultiWiBi taxonomies.

Limitations and Future Work. A key limitation of our work is that it is largely heuristic-driven. As discussed in the previous section, our efforts to utilize a classifier with the heuristics as features failed due to the difficulty of choosing an appropriate classification threshold. An interesting future work would be to use such a classifier in conjunction with another classifier that is specifically aimed towards detecting appropriate roots of the taxonomy.

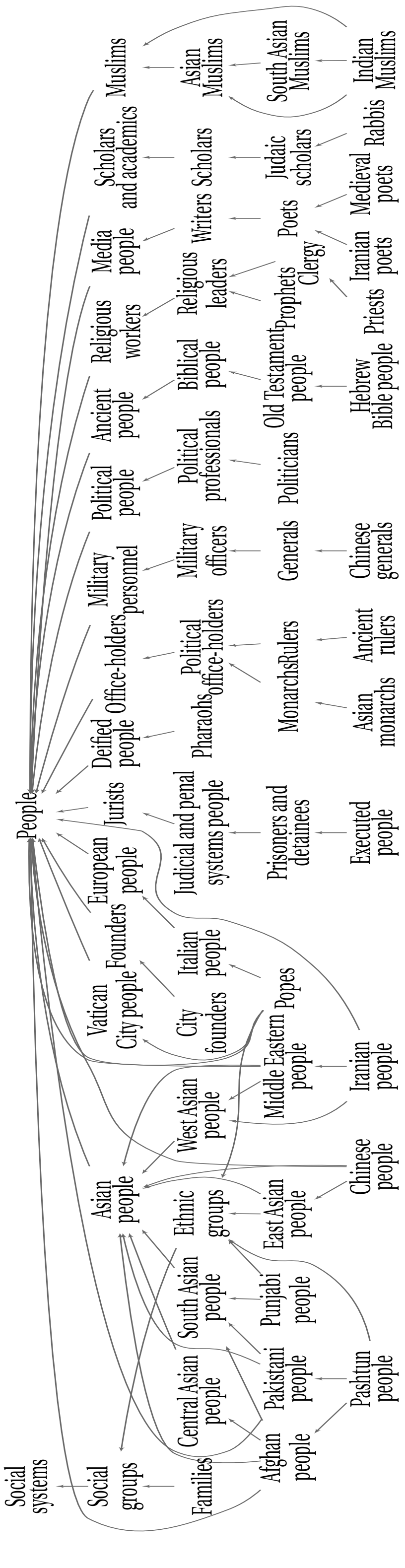


Figure 3.6 – A snippet of the HEADS taxonomy.

4 Multilingual Taxonomy Induction from Wikipedia

4.1 Overview

In the previous chapter, we proposed our approach that employs a novel set of heuristics to induce a large-scale taxonomy from the English WCN (i.e., Wikipedia categories network). We also demonstrated that the taxonomy induced using our approach (referred to as HEADS taxonomy) significantly outperforms the state of the art in edge-level accuracy and path-level evaluation measures. However, our approach has a severely-limiting constraint: it depends heavily on the syntactic structure of Wikipedia categories. As a result, it is not easily extensible to most other languages, which lack the availability of an accurate syntactic parser [71, 72]. MultiWiBi (described in Section 2.4.6) mitigated this constraint by constructing a probabilistic translation table from the anchor texts of internal hyperlinks of Wikipedia. A set of complex heuristics, which used the probabilistic translation table, was further employed to extract candidate hypernym lemmas in languages other than English.

However, in this chapter, we propose a different and completely novel approach to compensate for the lack of accurate syntactic parsers in other languages. Our approach is fully-automated, language-independent, and self-contained in Wikipedia. Similar to MultiWiBi, it also starts with a taxonomy induced from the English WCN. However, instead of relying on a set of complex heuristics for transferring this taxonomy to a target language (such as French), our approach first leverages the interlanguage links of Wikipedia to construct training datasets automatically for the *is-a* relation in the target language. Off-the-shelf text classifiers are trained on the constructed datasets and used in an optimal path discovery framework to induce high-precision, wide-coverage taxonomy in the target language.

Our approach provides a significant advancement over the state of the art in multilingual taxonomy induction from Wikipedia because of the following reasons:

- Most previous approaches such as MENTA or MultiWiBi rely on a set of complex heuristics that utilize custom hand-crafted features. In contrast, our approach is simpler, more principled and easily replicable.

- Our approach significantly outperforms the state-of-the-art approaches across multiple languages in both (1) standard edge-based precision/recall measures and (2) path-quality measures. Furthermore, our taxonomies have significantly higher branching factor than the state-of-the-art taxonomies without incurring any loss of precision.
- As a consequence of our work, we release presumably the largest and the most accurate multilingual taxonomic resource spanning over 280 languages. We also release edge-based gold standards for three different languages (i.e., French, Italian, Spanish) and annotated path datasets for six different languages (i.e., French, Italian, Spanish, Chinese, Hindi, Arabic) for further comparisons and benchmarking purposes.

4.2 Our Approach

We now describe our approach for inducing multilingual taxonomies from Wikipedia. Our approach takes three inputs: (1) the HEADS taxonomy, which is a unified taxonomy of English Wikipedia pages and categories induced in the previous chapter, (2) the interlanguage links (described in Section 2.2), and (3) the WCN in the target language (such as French). Given these inputs, our approach aims to induce a unified taxonomy of pages and categories for the target language. It runs in three phases:

1. **Projection phase**: in the first phase, the interlanguage links are used to create a high-precision, low-coverage taxonomy for the target language by simply projecting the *is-a* edges from the HEADS taxonomy.
2. **Training phase**: in the second phase, the high-precision taxonomy is leveraged to train classifiers that classify edges into *is-a* or *not-is-a* in the target language.
3. **Induction Phase**: in the final phase, a high-precision, high-coverage taxonomy is induced in the target language by running optimal path search over the target WCN. The probability of a WCN edge being *is-a* is computed using the trained classifiers and used as edge weights during the optimal path search.

It is noteworthy that although we use the HEADS taxonomy in the projection phase, our approach is compatible with any English taxonomy that consists of WCN nodes (i.e., pages or categories). We now describe the three phases of our approach in more detail.

4.2.1 Projection Phase

Let T_e be the given English taxonomy, which is the HEADS taxonomy in our case. Let G_f be the WCN and T_f be the (initially empty) output taxonomy in the target language f (such as French). For a node (i.e., page or category) $n_f \in G_f$, which has the English equivalent[1] n_e, and for which no hypernym exists yet in T_f, we perform the following steps:

[1] Two nodes are considered *equivalent*, if they are linked by an interlanguage link (Section 2.2).

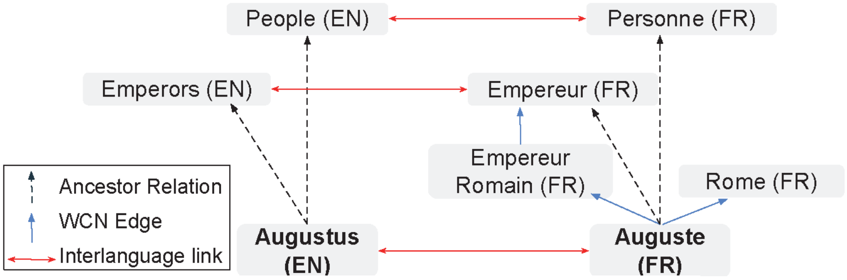


Figure 4.1 – Example of projection phase.

1. Collect the set A_e of all ancestor nodes of n_e in T_e up to a fixed height k_1.

2. Fetch the set A_f of equivalents for nodes in A_e in the target language f.

3. Find the shortest path in G_f between n_f and any node in A_f up to a fixed height k_2.

4. Add all the edges in the shortest path to the output taxonomy T_f.

If no English equivalent n_e exists, then the node n_f is ignored. In our experiments, $k_1 = 14$ sufficed as HEADS taxonomy had a maximum height of 14, and no cycles. k_2 is set to 3 to maintain high precision.

Figure 4.1 shows an example of the projection phase with French as the target language. For the French node *Auguste*, its English equivalent (i.e., *Augustus*) is fetched via the interlanguage link. The ancestors of *Augustus* in English taxonomy (i.e., *Emperors*, *People*) are collected, and mapped to their French equivalents (i.e., *Empereur*, *Personne*). Finally, the WCN edges in the shortest path from *Auguste* to *Empereur* (i.e., *Auguste*→*Empereur Romain*, *Empereur Romain*→*Empereur*) are added to the French taxonomy.

4.2.2 Training Phase

Up till now, we constructed an initial taxonomy for the target language by simply projecting the English taxonomy using the interlanguage links. However, the resulting taxonomy suffers from low coverage, because nodes that do not have an English equivalent are ignored. For example, only 44.8% of the entities and 40.5% of the categories from the French WCN have a hypernym in the projected taxonomy.

Therefore, to increase coverage, we train two different binary classifiers for classifying remaining target WCN edges into *is-a* (positive) or *not-is-a* (negative). The first classifier is for Entity→Category edges and the other for Category→Category edges[2]. We construct the training data for edge classification as follows:

[2] Entity→Entity and Category→Entity edges are not present in the WCN.

1. Assign an *is-a* label to the edges in T_f (i.e., the projected taxonomy in the target language).

2. Assign a *not-is-a* label to all the edges in G_f (i.e., the target WCN) that are not in T_f but originate from a node covered in T_f.

For example, in Figure 4.1, the edge *Auguste→Empereur Romain* is assigned the *is-a* label, and other WCN edges starting from *Auguste* (e.g., *Auguste→Rome*) are assigned the *not-is-a* label. We note that the *not-is-a* labels, which are assigned during this phase, are not final; they are only temporarily assigned for training the edge classifiers. The final labels are assigned in the next phase (i.e., the induction phase). While, some edges that are assigned temporary *not-is-a* labels may actually be correct *is-a* edges, this design ensures that most of the edges with the assigned *is-a* label, are correct *is-a* edges, thus leading to training of classifiers that achieve high accuracy.

Classifiers. To classify edges into *is-a* or *not-is-a*, we train classifiers using the constructed training sets. We experiment with the following off-the-shelf text classifiers:

1. **Bag-of-words TFIDF**: given edge *A→B*, concatenate the features vectors for *A* and *B* computed using TFIDF over the bag of words of their titles (e.g., “Empereur Romain” is the title of category *Empereur Romain*) and train a linear Support Vector Machine over the concatenated features. This method is hereafter referred to as **Word TFIDF**.

2. **Bag-of-character-*n*-grams TFIDF:** same as Word TFIDF, except TFIDF is computed over bag of character *n*-grams[3] (hereafter referred to as **Char TFIDF**).

3. **fastText:** a simple yet efficient baseline for text classification based on a linear model with a rank constraint and a fast loss approximation. Experiments show that fastText typically produces results on par with sophisticated deep learning classifiers [35].

4. **Convolutional Neural Network (CNN):** we use a single-layer CNN model trained on top of word vectors as proposed by Kim [60]. We also experiment with a character version of this model, in which instead of words, vectors are computed using characters and fed into the CNN. These models are referred to as **Word CNN** and **Char CNN** respectively. Finally, we experiment with a two-layer version of the character-level CNN proposed by Zhang et al. [144], which is referred to as **Char CNN-2l**.

5. **Long Short-term Memory Network (LSTM):** we experiment with both word-level and character-level versions of LSTM [49]. These models are hereafter referred to as **Word LSTM** and **Char LSTM** respectively.

[3] n-values={2,3,4,5,6} worked best in our experiments.

4.2.3 Induction Phase

In the last step of our approach, we discover taxonomic edges for nodes not yet covered in the projected taxonomy (T_f). To this end, we first set the weights of Entity→Category and Category→Category edges in the target WCN as the probability of being *is-a* (as computed using the corresponding classifiers). Further, for each node n_f that does not have a hypernym in T_f, we find the top k paths[4] with the highest probabilities originating from n_f to any node in T_f. The probability of a path is defined as the product of probabilities of individual edges. If multiple paths with the same probabilities are found, the shortest paths are chosen. The individual edges of the most probable paths are added to the T_f, resulting in the final taxonomy in the target language.

4.3 Evaluation and Results

We now evaluate the taxonomies induced using the approach described in the previous section. Similar to the evaluation of the English taxonomy in the previous chapter (Section 3.3), we evaluate our multilingual taxonomies against the state of the art using both edge-based and path-based evaluation methods. More specifically, in Section 4.3.1, we compute standard edge-level precision, recall, and coverage measures against a gold standard for three different languages (i.e., French, Italian and Spanish). In Section 4.3.2, we perform a comprehensive path-level comparative evaluation across six languages.

Analogous to the evaluation in the previous chapter, we compare our taxonomies against the MultiWiBi taxonomies [31], because there are multiple similarities between MultiWiBi and our approach:

- Only MENTA, MultiWiBi, and our taxonomies are constructed in a fully language-independent fashion. Hence, they are available for all 280 Wikipedia languages.
- Unlike YAGO3, MENTA and most other approaches, MultiWiBi and ours are self-contained in Wikipedia. They do not require manually labeled training examples or external resources, such as WordNet or Wikitionary.
- MultiWiBi is already shown to outperform most previous approaches across multiple languages [31].

4.3.1 Edge-level Evaluation

Experimental Setup. We create gold standards for three languages (French, Spanish and Italian) by selecting 200 entities and 200 categories randomly from the 2015 WCN and annotat-

[4] k is set to 1 unless specified otherwise.

ing their correctness[5]. Table 4.1 (page 57) shows a sample of annotated edges from the French gold standard. In total, 4045 edges were annotated across the three languages.

For evaluation, we reuse the same metrics, which are used in Section 3.3.2 as well as MultiWiBi [31]: (1) Macro-precision (P) defined as the average ratio of correct hypernyms to the total number of hypernyms returned (per node), (2) Recall (R) as the ratio of nodes for which at least one correct hypernym is returned, and (3) Coverage (C) as the ratio of nodes with at least one hypernym returned irrespective of its correctness.

Training Details. All neural network models are trained on Titan X (Pascal) GPU using the Adam optimizer [61]. A grid search is performed to determine the optimal values of hyperparameters. For CNN models, we use an embedding of 50 dimensions. The number of filters is set to 1024 for word-level models and 512 for character-level models. For Char CNN-2l model, we use the same parameters used in Zhang et al. [144]. For LSTM models, we use an embedding of 128 dimensions, and 512 units in the LSTM cell. We also experimented with more complex architectures, such as stacked LSTM layers and bidirectional LSTMs. However, these architectures failed to provide any significant improvements over the simpler ones.

Results. Table 4.2 shows the results for different methods including the state-of-the-art approaches (i.e., MENTA and MultiWiBi) and multiple versions of our three-phase approach with different classifiers. It also includes two baselines, i.e., **WCN** and **UNIFORM**. The WCN baseline outputs the original WCN as the induced taxonomy without performing any filtering of edges. UNIFORM is a uniformly-random baseline, in which all the edge weights are set to 1 in the induction phase (cf. Section 4.2.3).

Table 4.2 shows that all classifiers-based models achieve significantly higher precision than UNIFORM and WCN baselines, thus showing the utility of weighing with classification probabilities in the Induction phase. Interestingly, UNIFORM achieves significantly higher precision than WCN for both entities and categories across all three languages, hence, demonstrating that optimal path search in the Induction phase also contributes towards hypernym selection. All classifier-based approaches (except Word TFIDF) significantly outperform MultiWiBi for entities across all languages as well as for French and Spanish categories. Although MultiWiBi performs better for Italian categories, Char TFIDF achieves similar performance (89.2% vs. 89.7%) [6].

Coverage is 100% for all the baselines and the classifiers-based approaches because at least one path is discovered for each node in the induction phase, thus resulting in at least one (possibly

[5] Two annotators annotated each edge independently. Inter-annotator agreement (Cohen's Kappa) varied between 0.71 to 0.93 for different datasets.

[6] We note that entity edges are qualitatively different for MultiWiBi and other methods, i.e., MultiWiBi has Entity→Entity edges whereas other methods have Entity→Category edges. Given that fact and the unavailability of the gold standards from MultiWiBi, we further support the efficacy of our approach with a direct path-level comparison in the next section.

is-a	*not-is-a*
Naissance à Omsk→Naissance en Russie par ville	Naissance à Omsk⇝Omsk
Port d'Amérique du Sud→Port par continent	Port d'Amérique du Sud⇝Géographie de l'Amérique du Sud

Table 4.1 – Examples of Annotated Edges (French).

		Entity			**Category**		
Language	**Method**	**P**	**R**	**C**	**P**	**R**	**C**
French	Original WCN	72.0	100	100	78.8	100	100
	MENTA	81.4	48.8	59.8	82.6	55.0	65.7
	MultiWiBi	84.5	80.9	94.1	80.7	80.7	100
	UNIFORM	80.6	83.2	100	85.7	86.7	100
	Word TFIDF	86.5	90.1	100	82.1	83.1	100
	Char TFIDF	**88.0**	**91.7**	100	92.3	93.4	100
	fastText	86.5	90.1	100	90.5	91.6	100
	Word LSTM	**87.8**	**91.5**	100	91.6	92.7	100
	Char LSTM	86.2	89.8	100	**93.9**	**95.1**	100
	Word CNN	86.3	90.0	100	**92.8**	**93.9**	100
	Char CNN	86.2	89.9	100	**93.3**	**94.4**	100
	Char CNN-2l	**87.7**	**91.0**	100	92.2	93.3	100
Italian	Original WCN	74.5	100	100	76.2	100	100
	MENTA	79.7	53.2	66.7	77.1	25.4	32.8
	MultiWiBi	80.1	79.4	96.3	**89.7**	**89.0**	99.2
	UNIFORM	77.7	81.6	100	86.6	88.3	100
	Word TFIDF	**90.0**	**94.4**	100	84.1	85.7	100
	Char TFIDF	88.4	92.8	100	**89.2**	**90.9**	100
	fastText	86.8	91.1	100	**87.3**	**89.0**	100
	Word LSTM	**90.9**	**95.4**	100	83.1	84.8	100
	Char LSTM	89.8	94.4	100	83.3	83.8	100
	Word CNN	89.6	94.3	100	83.1	84.8	100
	Char CNN	**92.6**	**97.2**	100	86.9	88.7	100
	Char CNN-2l	87.7	92.1	100	86.1	87.8	100
Spanish	Original WCN	81.4	100	100	80.9	100	100
	MENTA	81.0	42.9	52.7	80.5	54.2	66.4
	MultiWiBi	87.0	82.0	93.7	84.8	84.4	100
	UNIFORM	88.0	90.7	100	83.0	85.0	100
	Word TFIDF	89.9	92.7	100	78.9	80.8	100
	Char TFIDF	92.5	95.4	100	88.3	90.4	100
	fastText	**93.0**	**95.9**	100	**88.9**	**91.0**	100
	Word LSTM	**93.4**	**96.3**	100	88.2	90.3	100
	Char LSTM	92.3	95.3	100	88.8	90.3	100
	Word CNN	92.9	95.8	100	87.6	89.7	100
	Char CNN	92.9	95.8	100	**92.9**	**95.1**	100
	Char CNN-2l	**93.3**	**96.3**	100	**89.9**	**92.1**	100

Table 4.2 – Edge-level precision (P), recall (R) and Coverage (C) scores for different methods. MENTA and MultiWiBi results are as reported by Flati et al. [31]. The top 3 results are shown in bold, and the best is also underlined.

incorrect) hypernym for each node in the final taxonomy. These results also demonstrate that the initial projected taxonomy (Section 4.2.1) is reachable from every node in the target WCN.

Word vs. Character Models. In general, character-level models outperform their word-level counterparts. Char TFIDF significantly outperforms Word TFIDF for both entities and categories across all languages. Similarly, Char CNN outperforms Word CNN. Char LSTM outperforms Word LSTM for categories, but performs slightly worse for entities. We hypothesize that this is due to the difficulty in training character LSTM models over larger training sets. Entity training sets are much larger, as the number of Entity→Category edges are significantly greater than the number of Category→Category edges (usually by a factor of 10).

Neural Models vs. TFIDF. CNN-based models perform slightly better on average, followed closely by LSTM and TFIDF respectively. However, the training time for neural networks-based models is significantly higher than TFIDF models. For example, it takes approximately 25 hours to train the Char CNN model for French entities using a dedicated GPU. In contrast, the Char TFIDF model for the same data is trained in less than 5 minutes.

Therefore, for the sake of efficiency, as well as to ensure simplicity and reproducibility across all languages, we choose Char TFIDF taxonomies as our final taxonomies for the rest of the evaluations. However, it is important to note that more accurate taxonomies can be induced by using our approach with neural-based models, especially if the accuracy of taxonomies is critical for the application at hand.

4.3.2 Path-level Evaluation

In the previous chapter, we demonstrated that high edge-level precision may not always translate to high path-level precision for taxonomies. We introduced the notion of length of *correct path prefix (CPP)*, i.e., the maximal correct prefix of a generalization path, as an alternative measure of the quality of a taxonomy (see Section 3.3.3). We computed two metrics based on the lengths of CPPs: (1) the average length of CPP (ACPP), and (2) the average ratio of lengths of CPPs to the full paths (ARCPP). Following the same evaluation methodology, we first randomly sample paths originating from 25 entities and 25 categories using the MultiWiBi and Char TFIDF taxonomies[7] for six different languages (i.e., French, Italian, Spanish, Arabic, Hindi, and Chinese). For each path, we annotate the first wrong hypernym edge in the upward direction. In total, we annotated 600 such paths across the six languages for the two approaches (i.e., MultiWiBi and Char TFIDF).

Table 4.3 shows some examples of these sampled paths, along with their CPPs. Table 4.4 shows the comparative results. Char TFIDF taxonomies significantly outperform MultiWiBi taxonomies, achieving higher ACPP for all languages and higher ARCPP for most languages.

[7] Same starting entities and categories are used for all taxonomies per language.

MultiWiBi
Patrimoine mondial en Équateur ⇝ Conservation de la nature → Écologie → Biologie → Sciences naturelles → Subdivisions par discipline → Sciences → Discipline académique → Académie → Concept philosophique

Char TFIDF
Patrimoine mondial en Équateur → Patrimoine mondial en Amérique → Patrimoine mondial par continent → Patrimoine mondial → Infrastructure touristique → Lieu ⇝ Géographie → Discipline des sciences humaines et sociales → Sciences humaines et sociales → Subdivisions par discipline

Table 4.3 – Samples of generalization paths for French categories. Correct path prefix (CPP) for each path is shown in bold.

			Entity			**Category**	
Language	**Method**	**AL**	**ACPP**	**ARCPP**	**AL**	**ACPP**	**ARCPP**
French	MultiWiBi	8.24	2.96	0.49	8.92	3.6	**0.56**
	Char TFIDF	11.08	**5.08**	0.49	8.36	**3.76**	0.49
Italian	MultiWiBi	7.36	2.68	0.45	14.84	3.72	0.27
	Char TFIDF	8.32	**4.88**	**0.61**	8.32	**4.52**	**0.57**
Spanish	MultiWiBi	7.04	3.08	**0.55**	12.08	4.08	0.36
	Char TFIDF	12.8	**5.0**	0.48	12.76	**5.28**	**0.48**
Arabic	MultiWiBi	8.96	2.12	0.31	14.64	4.12	0.31
	Char TFIDF	7.48	**5.88**	**0.81**	6.96	**5.04**	**0.74**
Hindi	MultiWiBi	7.72	1.88	0.27	7.4	1.8	0.36
	Char TFIDF	10.28	**4.92**	**0.47**	8.0	**2.44**	**0.38**
Chinese	MultiWiBi	7.4	2.56	0.47	8.0	4.43	0.63
	Char TFIDF	6.32	**3.92**	**0.68**	6.95	**4.48**	**0.68**

Table 4.4 – Comparison of average path length (AL), average length of correct path prefix (ACPP), and average ratio of CPP to path lengths (ARCPP) for the MultiWiBi and Char TFIDF taxonomies.

Therefore, compared to the state-of-the-art MultiWiBi taxonomies, Char TFIDF taxonomies are a significantly better source of generalization paths across multiple languages.

However, the overall performance of the Char TFIDF taxonomies is still significantly worse than HEADS taxonomy, which achieved an ARCPP of 0.87 (see Table 3.10). This effect is expected, because Char TFIDF taxonomies are created through the projection of the HEADS taxonomy. As a result, the errors in HEADS taxonomy would be propagated to the Char TFIDF taxonomies, thus, resulting in accuracy of HEADS being an upper bound for Char TFIDF taxonomies. This also suggests that hand-crafted language-specific features in conjunction with an accurate syntactic parser, as used for the induction of HEADS taxonomy, could possibly result in the induction of more accurate taxonomies for other languages as well.

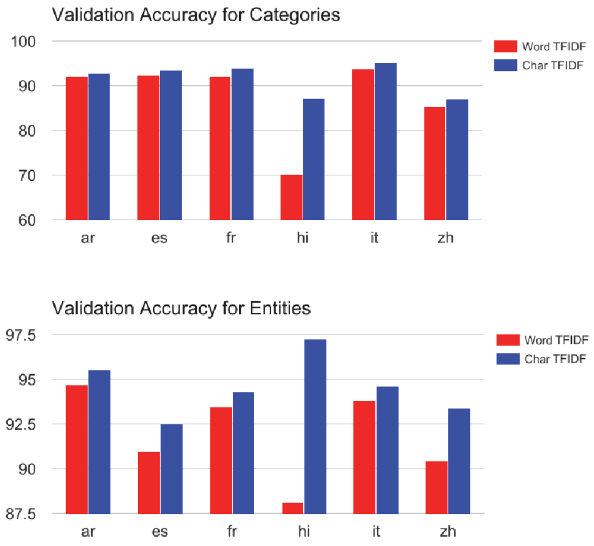
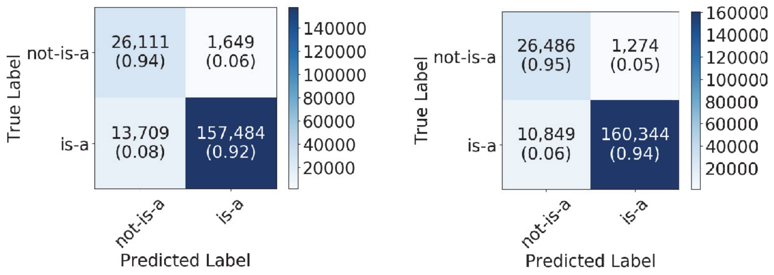


Figure 4.2 – Validation accuracies for Word TFIDF vs. Char TFIDF models.



(a) Word TFIDF

(b) Char TFIDF

Figure 4.3 – Confusion matrices for Word TFIDF vs. Char TFIDF models for French categories. Each cell shows the total number of edges along with the ratios in brackets.

4.4 Analysis

In this section, we perform additional analyses to gain further insights into our approach. More specifically, in Section 4.4.1 & 4.4.2, we perform an in-depth comparison of the Word TFIDF and Char TFIDF models. In section 4.4.3, we show the effect of the parameter k, i.e., the number of paths discovered during optimal path search (see Induction Phase in Section 4.2.3), on the branching factor and the precision of the induced taxonomies.

Word TFIDF	Char TFIDF
dolphins, dolphins, miami	s dol, s dolp, es dol
miami, entraîneur, des	hins, dolph, hins d

Table 4.5 – Top features for the *not-is-a* edge Entraîneur des Dolphins de Miami⇝Dolphins de Miami.

Word TFIDF	Char TFIDF
dolphins, américain, miami	ur spo, r spor, eur sp
entraîneur, sportif, entraîneur	tif am, if am, if amé

Table 4.6 – Top features for the *is-a* edge Entraîneur des Dolphins de Miami→Entraîneur sportif américain.

4.4.1 Word vs. Character Models

To compare word and character-level models, we first report the validation accuracies[8] for Word TFIDF and Char TFIDF models in Figure 4.2, as obtained during the training phase (cf. Section 4.2.2). Char TFIDF models significantly outperform Word TFIDF models, achieving higher validation accuracies across six different languages. The improvements are usually higher for languages with non-Latin scripts. This effect can be partly attributed to the error-prone nature of whitespace-based tokenization for such languages. For example, the word tokenizer for Hindi splits words at many accented characters in addition to word boundaries, thus leading to erroneous features and poor performance. In contrast, character-level models are better equipped to handle languages with arbitrary scripts, because they do not need to perform text tokenization.

4.4.2 False Positives vs. False Negatives

To further compare word and character models, we focus on the specific case of French categories. In Figure 4.3, we show the confusion matrices of Word TFIDF and Char TFIDF model computed using the validation set for French categories. While, in general, both models perform well, Char TFIDF outperforms Word TFIDF, producing fewer false positives as well as false negatives. In fact, we noticed similar patterns across most languages for both entities and categories.

We hypothesize that the superior performance of Char TFIDF is because character n-gram features incorporate the morphological properties computed at the sub-word level as well as word boundaries, which are ignored by the word-based features. To demonstrate this, we show in Tables 4.5 & 4.6, the top Word TFIDF and Char TFIDF features of a *not-is-a* and an *is-a* edge. These edges are misclassified by Word TFIDF but correctly classified by Char TFIDF.

[8] Validation set is constructed by randomly selecting 25% of the edges with each label (i.e., *is-a* and *not-is-a*) as discovered during the projection phase.

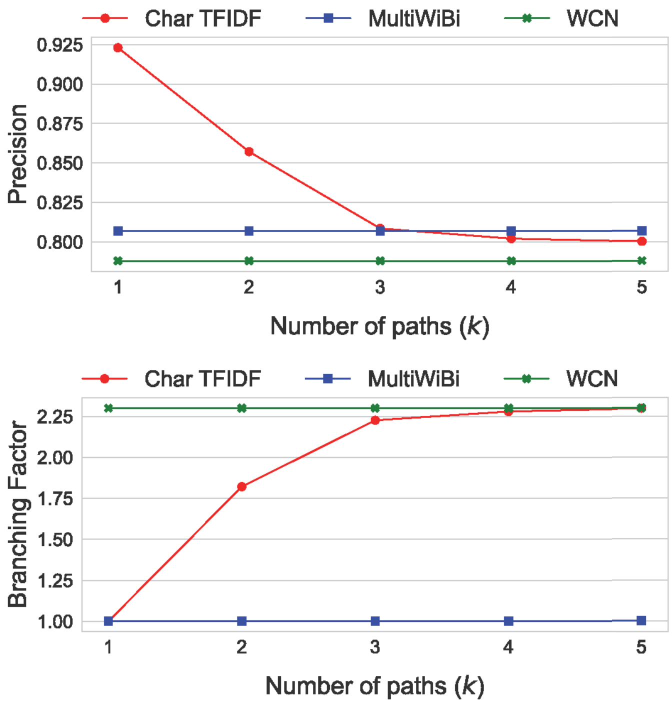


Figure 4.4 – Precision vs. branching factor for different number of paths (k) in the Induction phase (cf. Section 4.2.3).

While Word TFIDF features are restricted to individual words, Char TFIDF features can capture patterns across word boundaries. For example the 6-gram feature "*ur spor*" occurs in multiple hypernyms with different words: e.g., *Commentateur sportif américain*, *Entraîneur sportif américain* and *Entraîneur sportif russe.* Such features incorporate morphological information such as plurality and affixes, which can be important for the detection of an *is-a* relationship. Furthermore, Char TFIDF features are more robust to morphological variations, and hence, more suitable for handling inflected languages. This is also evidenced by our heuristics in Chapter 3 as well as prior work by Suchanek et al. [125] that utilize multiple hand-crafted features based on such morphological information. Therefore, character-level models equipped with such features perform better at the task of WCN edge classification than their word-level counterparts.

4.4.3 Precision vs. Branching Factor

Along with standard precision/recall measures, structural evaluation also plays an important role in assessing the quality of a taxonomy. One of the important structural properties of a taxonomy is the *branching factor*, which is defined as the average out-degree of the nodes in the taxonomy. Taxonomies with higher branching factors are desirable because they are better equipped to account for multiple facets of a concept or entity (e.g., BILL GATES is both a philanthropist and an entrepreneur).

However, there is usually a trade-off between branching factor and precision in automatically induced taxonomies [129]. Higher branching factor typically results in lowering of precision due to erroneous edges with lower scores being added to the taxonomy. Prioritizing the precision over the branching factor or vice-versa is usually determined by the specific use case at hand. Therefore, it is desirable for a taxonomy induction method to provide a control mechanism over this trade-off.

In our approach, the number of paths discovered (k) during the optimal path search in the induction phase (Section 4.2.3), serves as the parameter for controlling this trade-off. As k increases, the branching factor of the induced taxonomy increases because more paths per term are discovered. To demonstrate this effect, we plot the values of precision and branching factor of Char TFIDF taxonomies for varying values of k for French categories[9] in Figure 4.4. Precision and branching factors for the MultiWiBi taxonomy and the original WCN are also shown for comparison purposes.

Char TFIDF significantly outperforms MultiWiBi, either achieving higher precision ($k \leq 2$) or higher branching factor ($k \geq 2$). At $k=2$, Char TFIDF presents a sweet spot, outperforming MultiWiBi in both precision and branching factor. For $k \geq 3$, Char TFIDF taxonomies start to resemble the original WCN because most of the WCN edges are selected by optimal path discovery. This experiment demonstrates that in contrast to MultiWiBi's fixed set of heuristics, our approach provides better control over the branching factor of the induced taxonomies.

4.5 Discussion and Related Work

The large-scale and high quality of Wikipedia content has enabled multiple approaches towards knowledge acquisition and taxonomy induction over the past decade. The earlier attempts at taxonomy induction from Wikipedia focused on the English language. These include WikiTaxonomy, WikiNet, YAGO and the first versions of DBpedia and MultiWiBi. Later attempts aimed to extend the taxonomy induction process to other languages by exploiting the multilingual nature of Wikipedia content. These include MENTA, YAGO3, and the later versions of DBpedia and MultiWiBi. Chapter 2 provides a survey of these approaches.

In the previous chapter (Chapter 3), we proposed an approach that induces a unified taxonomy of entities and categories from the English WCN using a novel set of high-precision heuristics. In contrast, our approach proposed in this chapter is language-independent and results in taxonomies for all Wikipedia languages. Our approach borrows inspiration from many of the past approaches. First, similar to most previous approaches, it also classifies WCN edges into *is-a* or *not-is-a*. Second, similar to MultiWiBi, our approach also projects an English taxonomy into other languages using the interlanguage links.

However, unlike the previous approaches, our approach does not employ any linguistic heuristics or hand-crafted features. Instead, it uses standard text classifiers trained on an auto-

[9] Similar effects are observed for both entities and categories for all languages.

matically constructed dataset to assign edge weights to WCN edges. Taxonomic edges are discovered by running optimal path search over the WCN in a fully-automated and language-independent fashion. The principled design of our approach leads to two advantages: (1) our approach achieves 100% coverage, because unlike most heuristics, the text classifiers are applicable for all nodes, and (2) the parameter k in the optimal path search framework helps to regulate the precision vs. branching factor tradeoff, thus providing better control over the structural properties of the induced taxonomies.

Although the approach presented in this chapter uses the HEADS taxonomy as the input English taxonomy, theoretically it is general and replicable with any English taxonomy that consists of WCN nodes. However, since our approach collects ancestors of WCN nodes in the English taxonomy (Section 4.2.1), the high path-level accuracy of the HEADS taxonomy is a major advantage as it ensures more accurate sets of ancestors.

Our experiments show that taxonomies derived using our approach significantly outperform the state-of-the-art taxonomies, derived by MultiWiBi using more complex heuristics. We hypothesize that it is because our model primarily uses categories as hypernyms, whereas MultiWiBi first discovers hypernym lemmas for entities using potentially noisy textual features derived from unstructured text. Categories have redundant patterns, which can be effectively exploited using simpler models. This has also been shown in Chapter 3, where we employed simple high-precision heuristics based on the lexical head of categories to achieve significant improvements over MultiWiBi for English.

Additionally, for taxonomy induction in other languages, MultiWiBi uses a probabilistic translation table, which is likely to introduce further noise. However, the high-precision heuristics described in Chapter 3 are not easily extensible to languages other than English, due to the requirement of a syntactic parser for lexical head detection. Therefore, we present this approach that learns such features from automatically generated training data, hence resulting in high-precision, high-coverage taxonomies for all Wikipedia languages. Figure 4.5 shows some examples of generalization paths sampled from these taxonomies for ten different languages. Our taxonomies contain more than 1 million *is-a* edges for 10 languages, and more than 100,000 *is-a* edges for 46 languages. For rest of the languages, taxonomies are smaller (i.e., less than 50,000 *is-a* edges), mainly due to the smaller sizes of their corresponding WCNs. Nonetheless, our approach is still effective as it achieves 100% coverage over the WCNs by design.

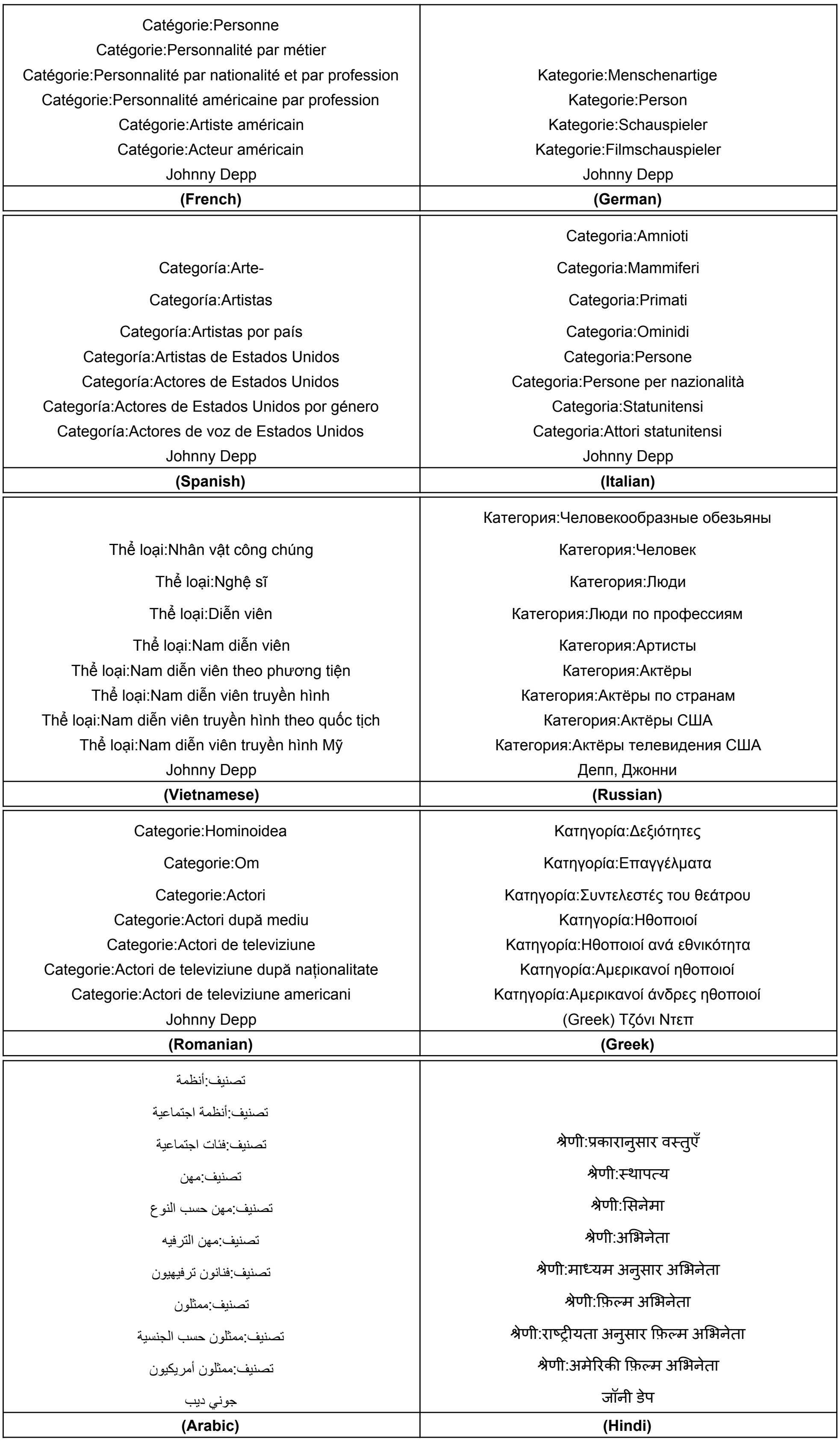
4.6 Summary

In this chapter, we presented a novel fully-automated approach towards multilingual taxonomy induction from Wikipedia. Unlike previous state-of-the-art approaches, which are complex and heuristic-heavy, our approach is simpler, principled and easy to replicate. Our approach runs in three phases. In the first phase, our approach leverages an English Wikipedia taxonomy and the interlanguage links of Wikipedia to project an initial taxonomy in the target language.

In the second phase, it constructs a training dataset automatically for the *is-a* relation in the target language. In the final phase, off-the-shelf text classifiers are trained on the constructed datasets and used in an optimal path discovery framework to induce a high-precision as well as wide-coverage taxonomy in the target language.

Taxonomies induced using our approach outperform the state of the art on both edge-level and path-level metrics across multiple languages. Our approach also provides a parameter for controlling the trade-off between precision and branching factor of the induced taxonomies. Additionally, our experiments demonstrate that character-level models perform better than their word-level counterparts at the task of classifying WCN edges because they are equipped with features related to word boundaries and morphological information. A key outcome of this work is the release of our taxonomies across 280 languages, which are significantly more accurate than the state of the art and provide higher coverage.

Limitations and Future Work. The first key limitation of our approach is that it uses an English taxonomy as the the source taxonomy for projection. This design could possibly introduce a bias in the taxonomies induced in other languages. For example, this design would favor the target language categories that have an English equivalent over categories without an English equivalent. An interesting future work would be analyze the relative distributions of interlanguage links across pages and categories in different languages, and use that information to identify potentially-beneficial source languages other than English. Another interesting approach could be to run taxonomy induction for all languages in iterative fashion in a unified framework, such that the taxonomies induced in each language aids the taxonomy induction all other languages. The second key limitation of our approach is that only one specific approach for the construction of the negative examples are presented in the training phase (Section 4.2.2). Experiments can be performed with other equivalent approaches. For example, negative training examples can be generated by projecting *not-is-a* edges from the source taxonomy. Finally, we only experimented with a few models for the classification of edges. Many other classification models could be tried and may potentially lead to improved results.



Catégorie:Personne Catégorie:Personnalité par métier Catégorie:Personnalité par nationalité et par profession Catégorie:Personnalité américaine par profession Catégorie:Artiste américain Catégorie:Acteur américain Johnny Depp	Kategorie:Menschenartige Kategorie:Person Kategorie:Schauspieler Kategorie:Filmschauspieler Johnny Depp
(French)	**(German)**

Categoría:Arte- Categoría:Artistas Categoría:Artistas por país Categoría:Artistas de Estados Unidos Categoría:Actores de Estados Unidos Categoría:Actores de Estados Unidos por género Categoría:Actores de voz de Estados Unidos Johnny Depp	Categoria:Amnioti Categoria:Mammiferi Categoria:Primati Categoria:Ominidi Categoria:Persone Categoria:Persone per nazionalità Categoria:Statunitensi Categoria:Attori statunitensi Johnny Depp
(Spanish)	**(Italian)**

Thể loại:Nhân vật công chúng Thể loại:Nghệ sĩ Thể loại:Diễn viên Thể loại:Nam diễn viên Thể loại:Nam diễn viên theo phương tiện Thể loại:Nam diễn viên truyền hình Thể loại:Nam diễn viên truyền hình theo quốc tịch Thể loại:Nam diễn viên truyền hình Mỹ Johnny Depp	Категория:Человекообразные обезьяны Категория:Человек Категория:Люди Категория:Люди по профессиям Категория:Артисты Категория:Актёры Категория:Актёры по странам Категория:Актёры США Категория:Актёры телевидения США Депп, Джонни
(Vietnamese)	**(Russian)**

Categorie:Hominoidea Categorie:Om Categorie:Actori Categorie:Actori după mediu Categorie:Actori de televiziune Categorie:Actori de televiziune după naționalitate Categorie:Actori de televiziune americani Johnny Depp	Κατηγορία:Δεξιότητες Κατηγορία:Επαγγέλματα Κατηγορία:Συντελεστές του θεάτρου Κατηγορία:Ηθοποιοί Κατηγορία:Ηθοποιοί ανά εθνικότητα Κατηγορία:Αμερικανοί ηθοποιοί Κατηγορία:Αμερικανοί άνδρες ηθοποιοί (Greek) Τζόνι Ντεπ
(Romanian)	**(Greek)**

تصنيف:أنظمة تصنيف:أنظمة اجتماعية تصنيف:فئات اجتماعية تصنيف:مهن تصنيف:مهن حسب النوع تصنيف:مهن الترفيه تصنيف:فنانون ترفيهيون تصنيف:ممثلون تصنيف:ممثلون حسب الجنسية تصنيف:ممثلون أمريكيون جوني ديب	श्रेणी:प्रकारानुसार वस्तुएँ श्रेणी:स्थापत्य श्रेणी:सिनेमा श्रेणी:अभिनेता श्रेणी:माध्यम अनुसार अभिनेता श्रेणी:फ़िल्म अभिनेता श्रेणी:राष्ट्रीयता अनुसार फ़िल्म अभिनेता श्रेणी:अमेरिकी फ़िल्म अभिनेता जॉनी डेप
(Arabic)	**(Hindi)**

Figure 4.5 – Sample Generalization Paths for the entity JOHNNY DEPP in ten languages.

Part II

Taxonomy Induction from Unstructured Text

5 Background and Related Work

5.1 Overview

Taxonomy induction is a well-studied task, and multiple different lines of work have been proposed in the prior academic literature. Early works on taxonomy induction utilize human-compiled knowledge resources including fully-structured resources (such as WordNet) or semi-structured resources (such as Wikipedia). Taxonomy induction approaches based on such resources achieve good precision and hence have been used in a wide variety of NLP-related tasks (Section 2.1). Additionally, the taxonomies extracted from Wikipedia are extremely large-scale, consisting of millions of entities (see Chapters 2-4). However, despite their large scale, such taxonomies still suffer from incomplete coverage over highly specialized domains such as Law and Finance, because such domains are usually under-represented in external knowledge resources. For example, WordNet is mostly limited to frequent nouns, adjectives, verbs, and adverbs [42, 89]. Similarly, Wikipedia articles are disproportionately focused on popular entities [65]. Furthermore, the utility of Wikipedia is further diminished by its slowed growth [127].

To address such issues, another line of work has been proposed, which focuses on building lexical taxonomies completely from *scratch*, i.e., unstructured or raw text such as domain-specific corpus or Web. The main advantage of performing taxonomy induction from scratch is that it can be performed on arbitrary domains because domain-specific text corpora can be easily harvested on a large scale using the Web [19, 102]. Furthermore, most Web documents provide temporal information that can be effectively utilized to induce up-to-date taxonomies even in highly dynamic domains such as Politics [146, 77].

In this chapter, we provide a brief survey of the past approaches towards taxonomy induction from unstructured text. These approaches typically consist of two main stages: (1) **hypernymy extraction**, i.e., extraction of hypernymy (or *is-a*) relations between terms from unstructured text, and (2) **term organization**, i.e., the structured organization of terms into a taxonomy, i.e., a coherent tree-like hierarchy. In Section 5.2, we discuss the past approaches aimed towards the first stage, i.e., hypernymy extraction from unstructured text, whereas, in Section 5.3,

Pattern	Context	Extracted Relation
NP is/was a *NP*	“apple is a fruit”	apple→fruit
NP such as *NP*	“authors such as shakespeare”	shakespeare→author
NP or/and other *NP*	“carrots or other vegetables”	carrot→vegetable
NP, especially *NP*	“swiss cities, especially Zurich”	zurich→swiss city
NP, e.g. *NP*	“scientists, e.g. Einstein”	einstein→scientist

Table 5.1 – Examples of lexico-syntactic patterns and extracted relations. NP indicates a noun phrase.

we focus on the second stage, i.e., term organization. In Section 5.4, we describe a few primary examples of end-to-end taxonomy systems that perform taxonomy induction from unstructured text.

5.2 Hypernymy Extraction

The task of extraction of hypernymy relations from unstructured text has been relatively well-studied in prior literature. Its approaches can be classified into two main categories: ***Distributional*** approaches and ***Pattern-based*** approaches.

Distributional approaches use clustering to extract hypernymy relations from unstructured text [100, 22, 111]. Such approaches draw primarily on the distributional hypothesis [46], which states that terms that are semantically-similar appear in similar contexts. The main advantage of distributional approaches is that they can discover relations, which are not explicitly expressed in the unstructured text.

In contrast, pattern-based approaches utilize pre-defined rules or lexico-syntactic patterns to extract terms and hypernymy relations from text [47, 98, 118]. Pattern-based approaches were pioneered by Hearst [47], and have been fairly popular ever since. Patterns are either chosen manually [47, 69] or learnt automatically via bootstrapping [119]. Table 5.1 shows some examples of these lexico-syntactic patterns along with sample contexts as well as the extracted hypernymy relations. Pattern-based approaches usually result in much higher accuracies [94, 129]. However, unlike distributional approaches, which are fully unsupervised, pattern-based approaches require a set of seed patterns to initiate the extraction process. Furthermore, pattern-based approaches can only extract relations that are explicitly expressed in unstructured text.

A third line of approaches towards hypernymy extraction uses machine learning classifiers, which are trained on distributional features or pattern-based features or a combination of both. For example, Snow et al. [119] search sentences containing two terms known to be in a taxonomic relation, and further automatically learn patterns from their parse trees. A classifier is trained based on such automatically-extracted pattern-based features, and used to identify novel hypernym pairs. Velardi et al. [129] extract hypernyms from a domain corpus and the Web, by extracting definitional sentences such as “apple is a fruit” (*apple→fruit*). Definitional sentences are recognized by a domain-independent machine-learned classifier that utilizes World Class Lattices (a form of regular expressions) trained on a dataset of Wikipedia definitions [93].

A more detailed survey of hypernym extraction techniques from unstructured text can be found in Wang et al. [132]. However, we describe a few of these techniques in detail, as they are employed by our taxonomy induction approaches in the following chapters.

- ***WebIsA*** is one of the most notable efforts towards hypernymy extraction from unstructured text. WebIsA is a dataset of hypernymy relations in English extracted automatically from the CommonCrawl web corpus using 59 hand-crafted lexico-syntactic patterns [118]. WebIsA is extremely large-scale, consisting of more than 400 million[1] hypernymy relations in English. Moreover, it is publicly available and can be downloaded and accessed via simple APIs[2].

- ***PattaMaika*** implements pattern-based knowledge extraction using UIMA Ruta[3], which is a rule-based text annotation engine released by Apache [66]. Similar to WebIsA, PattaMaika is publicly available[4], and has been used in previous works to extract hypernyms for multiple languages including English, Italian, and Dutch [102].

- ***PatternSim*** is a general tool for information extraction based on lexico-syntactic patterns. It has been used in a variety of tasks such as computation of semantic similarity [101] and hypernymy extraction from English and French corpora [102]. Similar to WebIsA and PattaMaika, it is also publicly available[5].

Overall, the task of hypernymy extraction from unstructured text is relatively well-studied. Many large-scale datasets as well as extraction systems have been publicly released and can be reused in an ‘as-is’ fashion by taxonomy induction approaches.

5.3 Term Organization

We now proceed with the discussion of the second stage of taxonomy induction, namely the structured organization of terms into a coherent tree-like hierarchy. Similar to hypernym extraction, approaches towards structured organization of terms can also be divided into two main categories: (1) ***clustering-based*** approaches, and (2) ***graph-based*** approaches.

[1] In contrast, the largest English taxonomies induced from Wikipedia comprised of approximately 12 million hypernymy relations.

[2] http://webdatacommons.org/isadb/

[3] http://uima.apache.org/ruta.html

[4] http://ltmaggie.informatik.uni-hamburg.de/jobimtext/documentation/pattern-extraction-with-pattamaika/

[5] https://github.com/cental/PatternSim

Clustering-based approaches aim to cluster terms that are co-hyponyms, i.e., they share the same hypernym. Typically, hierarchical clustering algorithms are employed to induce a tree-like hierarchy of terms. For example, Song et al. [122] employs an adapted version of hierarchical clustering for induction of large-scale taxonomies from a given set of keywords. Another approach, Alfarone and Davis [5] clusters terms using the K-Medoids algorithm, and computed the lowest common ancestor as the hypernym of a collection of terms.

Graph-based approaches cast the task of term organization as a graph optimization problem. They first construct a noisy hypernym graph from the extracted hypernym relations. The noisy hypernym graph is further pruned using a graph-based optimization algorithm, thus resulting in the induction of the final taxonomy. Graph-based approaches are well-suited for this task because taxonomies are essentially directed graphs with *is-a* edges between terms. One of the first such approaches was proposed by Kozareva and Hovy [68], who discover generalization paths from seed terms to a target root, by finding the longest path in a noisy hypernym graph. Another prominent approach is Ontolearn Reloaded [129], which employs the Chu-Liu/Edmonds's optimal branching algorithm [58] on the noisy hypernym graph with edge weights computed using the topology of the graph.

Both clustering-based and graph-based approaches have been effectively used in the prior academic literature for inducing taxonomies from unstructured text. A more detailed discussion of these approaches can be found in Velardi et al. [129] and Wang et al. [132].

5.4 State-of-the-art Approaches

We now describe a few salient end-to-end systems that perform taxonomy induction from unstructured text. Many of these systems use techniques or resources that are mentioned in the previous sections.

5.4.1 Kozareva's Method

Kozareva and Hovy [68] starts with an initial set of root terms (e.g., *animal*) and basic-level terms[6] (e.g., *lion*). It further employs lexico-syntactic patterns to harvest new candidate hypernyms for the basic-level terms using the Web. This step is performed recursively for the newly-harvested hypernyms until the root term is reached. Another set of lexico-syntactic patterns are employed to validate the extracted hypernymy relations.

Validated hypernymy relations are aggregated, leading to the construction of an initial noisy hypernym graph. The nodes in the noisy hypernym graph that have out-degree below a certain threshold are discarded. Cycles are detected and removed from the noisy hypernym graph, and the hypernymy relations constituting the longest paths between the basic-level terms and the root term form the final taxonomy.

[6] A basic-level term corresponds to the basic-level categories as defined in Rosch [114].

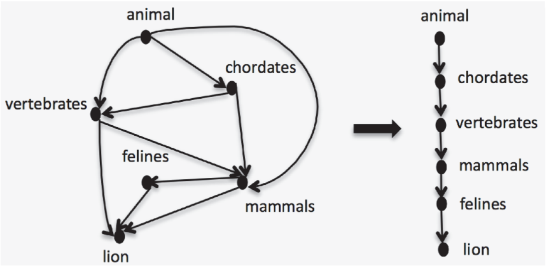


Figure 5.1 – Longest Path Optimization for Taxonomy Induction [68].

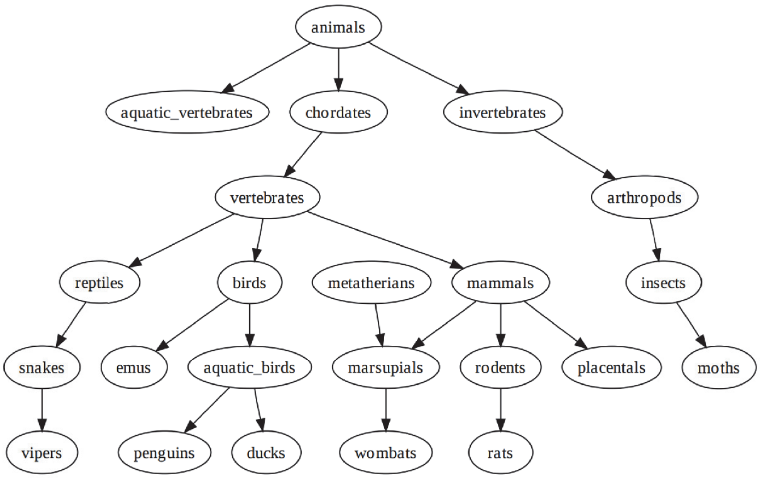


Figure 5.2 – Animal Taxonomy Induced from Scratch by Kozareva and Hovy [68].

Figure 5.1 shows a snippet of the noisy hypernym graph and the extracted longest path between the terms *lion* and *animal*. Figure 5.2 shows the final induced taxonomy for Animal domain. Kozareva and Hovy [68] report that their algorithm reconstructs up to 62% of the original WordNet from scratch over the tested regions, and also discovers novel hypernymy relations that are missing from WordNet.

5.4.2 Ontolearn Reloaded

Ontolearn Reloaded, proposed by Velardi et al. [129], is a novel algorithm that learns taxonomic relations from scratch, by extracting terms, definitions, and hypernyms from the Web. Ontolearn takes two inputs: (1) a domain-specific corpus, and (2) a set of candidate roots. Given these inputs, Ontolearn Reloaded works in four main steps:

1. **Term extraction:** in the first step, TermExtractor [117], which is a standard automated terminology extraction algorithm, is employed to extract a potentially-noisy domain-specific terminology from the given corpus.

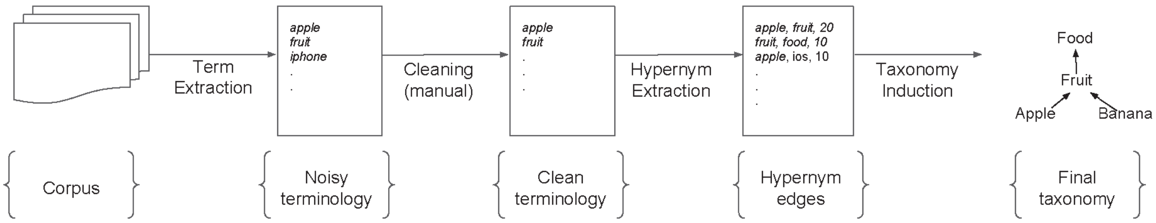


Figure 5.3 – Taxonomy induction process of Ontolearn Reloaded [129].

2. **Manual cleaning:** domain-irrelevant terms are manually discarded from the extracted terminology, thus resulting in a clean terminology.

3. **Hypernym extraction:** lexico-syntactic patterns-based classifiers, also known as Word-Class Lattices ([93]), are used to extract definitions for the domain-relevant terms from the Web. Domain-irrelevant definitions are further detected using a classifier and discarded. The hypernyms are extracted from the syntactic parses of retained definitions. For example, the definition "In graph theory, a flow network is a directed graph..." results in the extraction of the hypernymy relation *flow network→directed graph*. The hypernym extraction process stops when any of the input candidate roots are reached.

4. **Taxonomy induction:** in the final step, the hypernyms extracted from the definitions are aggregated to form an initial noisy hypernym graph. A novel weighting policy is employed for computing the edge weights of the noisy hypernym graph. Similar to Kozareva's method (Section 5.4.1), the weighting policy aims to assign higher weights to hypernym edges that fall on longer generalization paths. Finally, the application of Chu-Liu/Edmonds's optimal branching algorithm on the weighted hypernym graph results in the induction of the final taxonomy.

Figure 5.3 summarizes the above-mentioned taxonomy induction process of Ontolearn Reloaded. Ontolearn Reloaded is the first approach that performs taxonomy induction without making significant simplifying assumptions. As a result, Ontolearn Reloaded is considered a significant advancement over its prior approaches. However, despite such advancements, it still has a severely-limiting constraint: it requires a manual step of cleaning the terminology, which restricts its applicability in a fully-automated setting.

5.4.3 Taxify

Taxify is a hybrid approach that uses clustering-based as well graph-based techniques to induce a taxonomy from a domain-specific corpus in a fully-unsupervised fashion [5]. Taxify runs in four phases. In the first phase, an initial set of *is-a* relations are extracted automatically using a combination of lexico-syntactic patterns as well as distributional semantics. In the second phase, the terms are clustered using the K-Medoids algorithm, and the lowest common ancestor of terms in a cluster is considered as their hypernym. In the third phase, a graph-based optimal branching algorithm is employed to detect and remove potentially-incorrect *is-a* edges. In the final phase, confidence scores are assigned to the *is-a* edges based on the provenance of their discovery. Taxify performs favorably against Kozareva's method as well as Ontolearn Reloaded across five different domains.

5.4.4 SemEval Tasks

More recently, Bordea et al. [15, 16] introduced the first shared SemEval tasks on Taxonomy Extraction Evaluation, thus providing a common ground for evaluation. These SemEval tasks are referred to as TExEval [15] and TExEval-2 [16]. In both tasks, participants were provided with a clean vocabulary of domain-specific terms and a root term and asked to perform taxonomy learning by finding relations between pairs of terms. Resultant taxonomies were evaluated using a variety of methods such as structural evaluation, comparison against a gold standard as well as manual evaluation of edge-level accuracy. While TExEval task only focused on taxonomy induction over English, TExEval-2 task introduced three more languages, i.e., French, Italian and Dutch. INRIASAC, the top system in TExEval, uses features based on substrings and co-occurrence statistics [36] whereas TAXI, the top system in TExEval-2, uses lexico-syntactic patterns, substrings and focused crawling [102]. We now describe TAXI in detail, because it is used for comparative evaluation in Chapter 7.

TAXI. TAXI is a state-of-the-art taxonomy induction system, which reached first place in all the subtasks of the TExEval-2 task [102]. TAXI harvests candidate hypernyms using substring inclusion and lexico-syntactic patterns from unstructured text corpora. TAXI also uses the candidate hypernymy relations from the WebIsA database (see Section 5.2). It further utilizes an SVM trained with edge-level features, such as frequency counts of candidate hypernyms and substring inclusion, to classify edges as positive and negative. The edges that are classified as *is-a* are added to the taxonomy. Panchenko et al. [102] also report that alternate configurations of TAXI with different term-level and edge-level features as well as different classifiers such as Logistic Regression, Gradient Boosted Trees, and Random Forests fail to provide improvements over their approach. The key advantage of TAXI is that it is easily reproducible because its source code, as well as the extracted hypernyms, are released publicly[7].

5.5 Key Challenges

In the first chapter (Section 1.4), we mentioned some of the key shortcomings of taxonomy induction approaches that utilize unstructured text. In this section, we reiterate those shortcomings with additional context from the discussions presented in this chapter. Past approaches

[7] http://tudarmstadt-lt.github.io/taxi/

towards taxonomy induction from scratch typically suffer from these shortcomings:

- **Hypernymy extraction for general terms:** hypernymy extraction approaches based on the lexico-syntactic patterns usually become increasingly erroneous as the generality of terms increases, mainly due to the increase in term ambiguity. This effect was documented by Ontolearn Reloaded [129]. In the next chapter, we further demonstrate this effect using an empirical experiment.
- **Noisy input vocabulary:** most of the previous approaches, which are described in the previous section, require a clean vocabulary of seed terms as input. This constraint can be severely limiting because state-of-the-art automated vocabulary extraction approaches output vocabularies that contain numerous noisy terms. Although Taxify does not explicitly state this constraint, it is still evaluated only with clean vocabularies.
- **Automated root detection:** Kozareva's method, Ontolearn Reloaded and TAXI assume a set of one or more root terms as input. If such a set is unavailable, Ontolearn Reloaded employs higher-level terms from WordNet as the set of root terms. Although Taxify [5] performs taxonomy induction without a set of input roots, the final roots of the induced taxonomies are neither evaluated quantitatively nor qualitatively.

5.6 Summary

In this chapter, we provided a brief overview of the state of the art of taxonomy induction from unstructured text. The main stages of taxonomy induction from unstructured text are hypernymy extraction (Section 5.2) and term organization (Section 5.3). Hypernymy extraction is well-studied in prior literature and is mainly performed using either distributional methods or lexico-syntactic patterns. Although distributional methods are better equipped to extract implicit relations, lexico-syntactic patterns typically result in a higher accuracy of the extracted relations. The second stage, i.e., term organization, is performed using either clustering of terms, or graph-based optimization approaches. Although many of these methods have been utilized effectively for taxonomy induction, they still suffer from multiple shortcomings. In the following chapters, we propose novel methods that attempt to address these shortcomings. More specifically, in Chapter 6, we propose a novel model that utilizes the hypernyms of more specific terms, to choose more accurate hypernyms for more general terms. In Chapter 7, we introduce a novel flow network optimization-based approach towards term organization, that is robust to the presence of significant noise in the input vocabulary. Finally, in Chapter 8, we demonstrate that flow network optimization-based approach can be easily extended to support automated detection of roots.

6 Extraction of Hypernym Subsequences

6.1 Overview

As discussed in the previous chapter, taxonomy induction from unstructured text typically consist of two main stages: (1) extraction of hypernymy relations from unstructured text, and (2) the structured organization of terms into a taxonomy. The hypernym relations, which are extracted in the first phase, are usually directly employed in the second stage. However, in this chapter, we propose a novel approach that first extracts long-range **hypernym subsequences** from the extracted hypernyms. A hypernym subsequence is defined as a series of one or more contiguous hypernym edges (e.g., *apple→fruit→food*). Through experiments, we demonstrate that the subsequences extracted using our approach are significantly more accurate compared to multiple baselines. Moreover, in the next chapter, we demonstrate that the taxonomy induction approaches, which utilize these extracted hypernym subsequences, perform much better than equivalent approaches that solely rely on hypernym edges.

Since hypernymy extraction is relatively well-studied as well as orthogonal to our contribution, we assume the availability of a pre-existing database of hypernymy relations. More specifically, we use WebIsA, which is one of the largest databases of hypernymy relations in English (also described in Section 5.2). WebIsA contains more than 400 million hypernymy relations in English. However, these relations tend to be very noisy, typically containing a mixture of closely-related semantic relations such as hyponymy, meronymy, synonymy, and co-hyponymy (see Section 1.2 for definitions of these semantic relations). For example, WebIsA has more than 12,000 hypernyms for the term *apple*, including numerous noisy hypernyms such as *orange*, *everyone* and *smartphone*. For each hypernymy relation, WebIsA also provides the occurrence frequencies in the CommonCrawl corpus. The hypernymy relations with the highest occurrence frequencies for the term *apple* are shown in Table 6.1.

In the remainder of this thesis, the noisy hypernymy relations present in WebIsA are referred to as the ***candidate hypernyms***. In the next section, we present our approach that extracts hypernym subsequences from these candidate hypernyms.

Candidate hypernym	Occurrence frequency
company	5536
fruit	3898
apple	2119
vegetable	928
orange	797
tech company	619
brand	463
hardware company	460
technology company	427
food	370

Table 6.1 – WebIsA hypernyms for the term *apple* along with their occurrence frequencies [118].

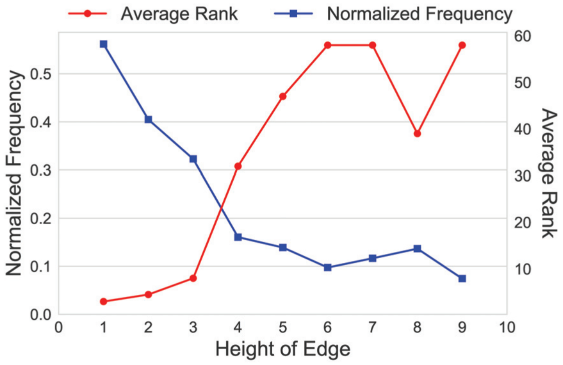


Figure 6.1 – Normalized occurrence frequency and average rank vs. the height of the edge in the paths sampled from WordNet.

6.2 Our Approach

6.2.1 Motivation

To motivate the extraction of hypernym subsequences, we first note that Table 6.1 includes hypernyms of *apple* at different levels of generality, such as *fruit* and *food*. In fact, we observe this pattern in the candidate hypernyms of most terms. This suggests that we can leverage such information to not only extract the direct hypernyms of *apple*, but to also extract longer hypernym subsequences, such as *apple*→*fruit*→*food*.

This becomes even more important given the result by Velardi et al. [129], who demonstrated that hypernym extraction becomes increasingly erroneous as the generality of terms increases, mainly due to the increase in term ambiguity. To further support this hypothesis, we perform an experiment where we first randomly sample 100 paths from WordNet. For each edge a→b in a sampled path, we plot the normalized occurrence frequency[1] of "b as a candidate hypernym for a" against the height of the edge (Table 6.1). We also plot the average rank of b

[1] Normalization is performed by dividing the frequency counts by the maximum.

among candidate hypernyms of a, where candidate hypernyms are ranked by their normalized occurrence frequencies in decreasing order.

Figure 6.1 shows the results of this experiment. Since edges in the WordNet are assumed to be the ground truth, it is desired that they have a higher normalized frequency and lower ranks. However, this small-scale experiment demonstrates that as the height of the edge increases, the normalized frequencies decrease whereas the average ranks increase. Therefore, the accuracy of candidate hypernyms is lower for more general terms that appear higher in WordNet paths. Hence, for such terms, it makes sense to not solely base the hypernym selection on the noisy set of candidate hypernyms. We can potentially improve the accuracy of the selected hypernyms for general terms (such as *fruit*) by relying on hypernym subsequences starting from more specific terms (such as *apple*). Those subsequences would be evidenced by the less-noisy candidate hypernyms of the more specific terms.

In sum, extracting hypernym subsequences is both *possible* and potentially *beneficial*. The remainder of this section describes our model that exploits this intuition.

6.2.2 Model

We now describe our model for extracting hypernym subsequences for a given term. We begin with a general formulation using directed acyclic graphs (hereafter referred to as DAG), and we make simplifying assumptions to derive a model for hypernym subsequences. We first describe some notations, which will serve us for the rest of this section:

- t_0: a given seed term, e.g., *apple*;
- l_t: lexical head of any term t, e.g., l_t=*soup* for t=*chicken soup*;
- E: Hypernym Evidence, i.e., the set of all the candidate hypernymy relations, in the form of 3-tuples (*hyponym, hypernym, frequency*);
- $E_k(t)$: Hypernym Evidence for term t, i.e., the set of top-k candidate hypernyms for the term t, which have the highest occurrence frequency counts (Table 6.1 shows a sample from $E_k(t)$ for t=*apple*);
- $E_k(t,m)$: m^{th} ranked candidate hypernym from $E_k(t)$, where $m \leq k$, and ranks are computed by sorting candidate hypernyms in decreasing order of frequency counts;
- $\text{sim}(t_i, t_j)$: A similarity measure between terms t_i and t_j estimated using evidence E;
- G_t: a DAG consisting of generalizations for a term t (Figure 6.2 shows an example of a possible DAG for t=*apple*).

For a given term t_0, we define the goal of our model as finding a DAG $\hat{G}_{t_0}$, which maximizes

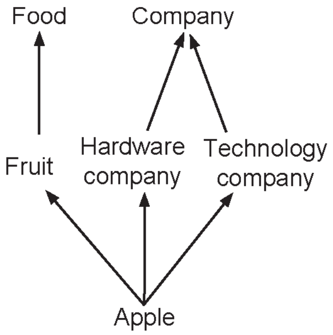


Figure 6.2 – An example DAG built using generalizations of term *apple*.

the conditional probability of G_{t_0}, given the evidence $E_k(t_0)$, for a fixed k:

$$\begin{aligned}\hat{G}_{t_0} &= \underset{G_{t_0}}{\operatorname{argmax}} \Pr(G_{t_0}|E_k(t_0)) \\ &= \underset{G_{t_0}}{\operatorname{argmax}} \Pr(E_k(t_0)|G_{t_0}) \times \Pr(G_{t_0}) \end{aligned} \tag{6.1}$$

Due to the combinatorial nature of the search space of G_{t_0}, finding an exact solution to the above equation is intractable, even for a small k. Therefore, we make the following simplifying assumptions, which facilitate an efficient search through the search space of G_{t_0}:

- G_{t_0} can be approximated as a set of independent hypernym subsequences with possibly repeated hypernyms. In other words, $G_{t_0} = \bigcup_{i=1}^{b} S_{t_0}^i$ where $S_{t_0}^i$ is the i^{th} subsequence and b is a fixed constant. For example, the DAG shown in Figure 6.2 can be approximated as a set of three subsequences: (i) *apple→fruit→food*, (ii) *apple→hardware company→company*, and (iii) *apple→technology company→company*. This assumption intuitively derives from the fact that any DAG can be represented by a finite number of subsequences. These subsequences can be generated in linear time, by first performing a topological sort of the DAG, and further iterating over all the paths from the term t_0 to the terms with the highest ranks in the topological sort [23].

- $\forall i$, the joint events $(E_k(t_0), S_{t_0}^i)$ are independent. Intuitively, this assumption implies that each subsequence independently contributes to the evidence $E_k(t_0)$.

- $\forall i$, the direct hypernyms of t_0 in $S_{t_0}^i$ are unique. In other words, for a candidate hypernym h_c of the given term t_0, there is at most one subsequence with the first edge $t_0 \rightarrow h_c$. Intuitively, this assumption implies that a candidate hypernym h_c uniquely sense-disambiguates the term t_0.

In conjunction, these assumptions imply that G_{t_0} is composed of b hypernym subsequences, where each subsequence independently attempts to generate $E_k(t_0)$. Given these assumptions,

Equation 6.1 transforms into:

$$\hat{G}_{t_0} = \underset{\bigcup_{i=1}^{b} S_{t_0}^{i}}{\operatorname{argmax}} \prod_{i=1}^{b} \Pr(E_k(t_0)|S_{t_0}^{i}) \times \Pr(S_{t_0}^{i}) \tag{6.2}$$

Estimation. We now describe the estimation of $\Pr(E_k(t_0)|S_{t_0}^i)$ and $\Pr(S_{t_0}^i)$ for a hypernym subsequence $S_{t_0}^i$. In order to motivate the estimation of the conditional probability $\Pr(E_k(t_0)|S_{t_0}^i)$, we start with an example. Consider a valid hypernym subsequence for the term *apple*, whose candidate hypernyms are in Table 6.1:

apple → *fruit* → *food* → *substance* → *matter* → *entity*

At first sight, it might seem desirable for a candidate hypernym from $E_k(t_0)$ (e.g., *fruit*) to have a high similarity with as many terms in the subsequence as possible. However, since the similarity measure is estimated using the hypernym evidence E, it is plausible that terms such as *matter* and *entity* have a low similarity with the candidate hypernym *fruit*, simply because they are at a higher level of generality.

To avoid penalizing such valid subsequences, we let the conditional probability $\Pr(E_k(t_0)|S_{t_0}^i)$ be proportional to the maximum similarity possible between the candidate hypernym and *any* term in the subsequence (e.g., for the candidate hypernym *fruit*, the similarity is 1 as *fruit* is in the subsequence). We aggregate those similarity values across the candidate hypernyms.

More formally, assuming subsequence $S_{t_0}^i = t_0 \rightarrow h_{i1} \rightarrow h_{i2} \ldots h_{in}$, where n is the length of $S_{t_0}^i$, we compute the conditional probability as:

$$\Pr(E_k(t_0)|S_{t_0}^i) \propto \sum_{m=1}^{k} (\lambda_1)^m \max_{j\in[1,n]} \big(\operatorname{sim}(E_k(t_0,m), h_{ij})\big) \tag{6.3}$$

where λ_1 (a fixed parameter) serves as a rank-penalty to penalize candidate hypernyms with lower frequency counts.

We now proceed to compute $\Pr(S_{t_0}^i)$, the other constituent of Equation 6.2. Towards that, we assume that the subsequence $S_{t_0}^i$ is a collection of independent hypernym edges. Thus, $\Pr(S_{t_0}^i)$ becomes the product of the probabilities of the individual edges in $S_{t_0}^i$:

$$\Pr(S_{t_0}^i) \propto \Pr_e(t_0, h_{i1}) \times (\lambda_2)^n \prod_{j=1}^{n-1} \Pr_e(h_{ij}, h_{i(j+1)}) \tag{6.4}$$

where $\Pr_e(x_1, x_2)$ is the probability of an individual hypernym edge $x_1 \rightarrow x_2$ between terms x_1 and x_2; λ_2 is a length penalty parameter.

Finally, we estimate $\Pr_e(x_1, x_2)$ as a log-linear model using a set of features **f**, weighted by the learned weight vector **w**:

$$\Pr_e(x_1, x_2) \quad \propto \quad \exp\big(\mathbf{w} \cdot \mathbf{f}(x_1, x_2)\big) \tag{6.5}$$

We also use this edge probability to compute the aforementioned similarity function (sim) as:

$$\mathrm{sim}(x_i, x_j) \quad = \quad \max\big(\Pr_e(x_i, x_j), \Pr_e(x_j, x_i)\big) \tag{6.6}$$

Intuitively, $\Pr(E_k(t_0)|S^i_{t_0})$ promotes subsequences, which contain a larger number of candidate hypernyms from $E_k(t_0)$, whereas $\Pr(S^i_{t_0})$ promotes subsequences, which consist of individual edges with a larger probability of hypernymy.

Subsequence Extraction. After inserting Equations 6.3 & 6.4 into Equation 6.2 and taking logarithm, the objective function becomes:

$$\begin{aligned}\hat{G}_{t_0} = \operatorname*{argmax}_{\bigcup_{i=1}^{b} S^i_{t_0}} \sum_{i=1}^{b} \Big[\log \sum_{m=1}^{k} (\lambda_1)^m \max_{j \in [1,n]} \big(\mathrm{sim}(E_k(t_0, m), h_{ij})\big) \\ + \log \Pr_e(t_0, h_{i1}) + n\lambda_2 + \sum_{j=1}^{n-1} \log \Pr_e(h_{ij}, h_{i(j+1)})\Big]\end{aligned} \tag{6.7}$$

This objective function leads to the following search algorithm for the extraction of hypernym subsequences:

1. For a given term t_0, iterate over all candidate hypernyms in $E_k(t_0)$.
2. For each $h_c \in E_k(t_0)$, perform a depth-limited beam search over the space of possible subsequences by recursively exploring the candidate hypernyms of h_c (i.e., $E_k(h_c)$).
3. For each $h_c \in E_k(t_0)$, choose the subsequence S with the highest score (i.e., $\log(\Pr(E_k(t_0)|S) \times \Pr(S))$).
4. Choose the top-b candidate hypernyms in a greedy fashion, based on their corresponding subsequence scores.

While, in theory, we can iterate over all candidate hypernyms in $E_k(t_0)$, in practice, we employ an alternative two-phase execution that significantly improves the running time as well as produces more meaningful subsequences. These two phases are described as follows.

Search Phase. Proceed as in the aforementioned steps. However, in the special case where a candidate hypernym h_c is a compound term and its lexical head l_{h_c} is also present in $E_k(t_0)$, skip h_c in step (1) of the algorithm. For example, for t_0 = *apple*, candidate hypernyms *tech company*, *software company* and *hardware company* are skipped in step (1) due to the presence of *company* in $E_k(t_0)$ (see Table 6.1).

Expansion Phase. In this phase, we augment the subsequences extracted in the search phase to account for skipped compound terms. We focus on the case where the lexical head of the skipped compound terms occurs in a subsequence. In that case, we expand the incoming edge of the lexical head with zero or more of those compound terms. For example, in the subsequence *apple→company→organization*, a potential expansion of the edge *apple→company* is: *apple→American software company→software company→company*.

However, special attention has to be taken while generating these expansions. For example, the expansion *apple→American software company→British software company→company* is invalid due to the co-hyponymy edge *American software company→British software company*. In contrast, the expansion *apple→American software company→software company→ company* is a valid expansion. To avoid invalid expansions, we restrict the possible expansions to the cases where the set of pre-modifiers of a compound term is a superset of its hypernym's pre-modifiers (e.g., {*American, software* }⊃{*software*}).

We generate all possible expansions for each edge and rank them by averaging a TF-IDF-style metric across the pre-modifiers of compound terms in each expansion. The goal of our approach for ranking the potential expansions is two-fold:

1. promote the pre-modifiers, which frequently appear in the evidence $E_k(t_0)$.
2. penalize the noisy pre-modifiers unrelated to t_0 that frequently occur in compound terms (e.g., *several*, *other*, etc.).

To achieve these goals, we compute the TF score of a pre-modifier as its average frequency of occurrence in the candidate hypernyms $E_k(t_0)$. We compute IDF as the average frequency of occurrences of the pre-modifier in $E_k(t)$ for a random term t. Finally, we choose the top-ranked expansion per edge.

To illustrate the result of the previous steps, we show in Table 6.2 an example of extracted subsequences along with their expanded versions for the food domain. Intuitively, the two-phase execution serves to distinguish between two different forms of generalization:

Initial subsequences
mortadella→sausage→meat→food
laksa→soup→dish→food
Expanded subsequences
mortadella→large italian sausage→sausage→processed meat→meat→food
laksa→spicy noodle soup→noodle soup→soup→dish→food

Table 6.2 – Examples of hypernym subsequences found during the search phase, and their expanded versions.

1. **Type-based generalization**, which provides core types as generalizations. Examples of such core types include *company* and *organization* in the hypernym subsequence *apple→company→organization*.

2. **Attribute-based generalization**, which enriches type-based generalization edges. For example, *apple→american software company→software company→company* enriches the type-based generalization edge *apple→company*.

In our experiments, we observed that models, which distinguish between these two forms of generalizations, performed consistently better than models that attempted to unify them. We hypothesize that it is because these two types of generalization are fundamentally different. In comparison with the type-based generalization edges, attribute-based generalization edges are more likely to be correct, because of the condition of the same lexical head. Hence, attribute-based generalization edges do not require the same strength of evidence as the type-based generalization edges.

Our hypothesis is further corroborated by the observation that type-based and attribute-based generalizations can also be noticed in the taxonomies induced from the Wikipedia categories networks. For example, the generalization path for Johnny Depp sampled from the Heads Taxonomy (cf. Table 3.8) contains the subsequence Johnny Depp→American film producers→American producers→Producers, which evidences the two different forms of generalization. Similar patterns can be observed in multiple languages (Figure 4.5), thus, further supporting our hypothesis.

6.2.3 Features

We now describe the edge features that we employ for estimating the probability of a hypernymy relation between two terms (cf. Equation 6.5). Each edge feature is a function, which takes two terms (i.e., hyponym and hypernym) as input, and return a float value. Edge features can be further divided into three categories based on the data needed for their computation:

Counts-based Features. These features mainly use the frequency counts of the hypernyms found in the hypernym database (cf. Table 6.1). We use the following count-based features:

- ***Normalized Count** (n_f)*: As the name suggests, for the pair (x_i, x_j), this feature returns the normalized frequency count of x_j in the hypernym Evidence for x_i (i.e., $E_k(x_i)$). More specifically, $n_f(x_i, x_j) = \frac{freq(x_i,x_j)}{\max\limits_m freq(x_i,x_m)}$, where $freq(x_a, x_b)$ is the frequency count of x_b in $E_k(x_a)$.
- ***Normalized Diff** (n_d)*: this feature computes an asymmetric hypernymy score based on the frequency counts. It returns the difference of the normalized counts in the two directions ($x_i \rightarrow x_j$ and $x_j \rightarrow x_i$), i.e., $n_d(x_i, x_j) = n_f(x_i, x_j) - n_f(x_j, x_i)$. Intuitively, the normalized diff feature helps in down-ranking noisy relations such as synonyms and co-hyponyms (e.g., *apple* and *orange*) because they usually receive high-frequency counts in both directions. A similar feature is also used by Panchenko et al. [102].

String-based Features. These features mainly use the term strings of the hyponyms and the hypernyms in their computation. We use the following string-based features:

- ***Substring Beginswith***: For the pair (x_i, x_j), this features returns 1 if x_i begins with x_j, otherwise returns 0. For example, (*sportspeople*, *sports*) will receive the value 1, whereas (*sports*, *people*) will receive the value 0.
- ***Substring Endswith***: For the pair (x_i, x_j), this features returns 1 if x_i ends with x_j, otherwise returns 0. For example, (*sportspeople*, *people*) will receive the value 1, whereas (*sports*, *people*) will receive the value 0.
- ***Substring Contains***: For the pair (x_i, x_j), this features returns 1 if x_i contains x_j, otherwise returns 0. For example, both (*sportspeople*, *sport*) and (*sportspeople*, *people*) will receive the value 1.
- ***Length Ratio***: this feature returns the ratio of lengths of the hypernym term to the hyponym term.

Generality-based Features. We introduce two novel features for explicitly incorporating the generality or abstractness of a term in our model. To this end, we first define the generality $g(t)$ of a term t as the log of the number of distinct hyponyms present in all candidate hypernymy relations (E); i.e., $g(t) = \log(1 + |x \mid x \rightarrow t \in E|)$. We also define the generality of an edge as the difference in generality between the hypernym and the hyponym: $g_e(x_i, x_j) = g(x_j) - g(x_i)$. We now describe the generality-based features that use these definitions:

- ***Generality Diff** (g_d)*: Intuitively, this feature aims to promote edges at the right level of generality and penalize edges, which are either too general (e.g., *apple→thing*) or too specific (i.e., edges between synonyms or co-hyponyms, such as *apple→orange*). To realize this intuition, we first sample a random set of terms and collect the edges with the highest Normalized diff (n_d) for these terms (hereafter referred to as *top edges*). We compare the distribution of generality (i.e., g_e) for the top edges vs. the distribution of generality for a set of randomly sampled edges.

Further, we make the assumption that it is more likely to sample the generality of a correct edge (i.e., edge at right level of generality) from the distribution of top edges as compared to random edges. Hence, given D_t and D_r as the Gaussian distributions estimated from the samples of generality for top edges and random edges respectively, we define the feature as: $g_d(x_i, x_j) = \Pr_{D_t}\big(g_e(x_i, x_j)\big) - \Pr_{D_r}\big(g_e(x_i, x_j)\big)$.

- ***Generality Probability (g_p)***: The computation of this feature is similar to the previous feature. However, in the final equation, only the first constituent computed using the top edges distribution is used, i.e., $g_p(x_i, x_j) = \Pr_{D_t}\big(g_e(x_i, x_j)\big)$.

The relative weights for the features (**w** in equation 6.5) are estimated using a support vector machine (hereafter referred to as SVM) trained on a manually annotated set of 500 edges (50 terms, 10 edges per term).

6.3 Evaluation and Results

The subsequence extraction approach presented in the previous section is hereafter referred to as ***SubSeq***. In this section, we evaluate the quality of subsequences extracted by SubSeq using two different evaluation methodologies. First, in Section 6.3.1, we perform automated evaluation using WordNet as a source of ground truth hypernym edges. Second, in Section 6.3.2, we perform manual annotations to assess the quality of the subsequences.

Before we proceed with the evaluations, we first introduce two baselines for comparison purposes. Similar to SubSeq, these baselines also utilize beam search to discover long-range hypernym subsequences for a given starting term using the candidate hypernyms. However, in contrast with SubSeq, which aims to maximize the objective function in Equation 6.7, these baselines aim to maximize the product of probabilities of individual edges in the subsequence. In other words, the objective function for these baselines is $\Pr(S^i_{t_0})$ instead of $\Pr(E_k(t_0)|S^i_{t_0}) \times \Pr(S^i_{t_0})$ (cf. Equation 6.2).

The first baseline uses only the Normalized Count feature (n_f in the previous section) as the probability of an individual edge. It is hereafter referred to as ***TopEdge***. The second baseline uses the SVM for computing the probability of an individual edge. It is hereafter referred to as ***TopEdge+SVM***. These two baselines are collectively referred to as the ***edge-based baselines***.

6.3.1 Automated Evaluation

Experimental Setup. In this experiment, we evaluate the quality of hypernym subsequences extracted by the edge-based baselines as well as SubSeq, using a fully automated approach. Our evaluation requires two inputs: (1) a source for ground truth hypernyms for seed terms, and (2) a vocabulary of seed terms, for which hypernym subsequences can be extracted.

We use WordNet as the source of ground truth hypernyms. In contrast with the candidate hypernymy database, which is automatically extracted from unstructured text, the WordNet is constructed manually by domain experts. As a result, it is highly accurate and usually considered the gold standard for hypernymy extraction and taxonomy induction tasks [133, 120, 10]. For a given term, WordNet provides multiple synsets, where each synset corresponds to a unique sense of the term. Other synsets are provided as the hypernyms for each synset. For example, the WordNet synsets for the term apple *apple* include the "fruit" sense as well as "tree" sense. The hypernym for the synset corresponding to the "fruit" sense is "edible fruit", whereas the hypernym for the synset corresponding to the "tree sense" is "fruit tree". Incidentally, the "company" sense of apple is not present the WordNet, which serves to show its low coverage and hence, the need for automated taxonomy induction approaches.

To construct a vocabulary of seed terms, we randomly sample 100 terms from the food vocabulary released by the Taxonomy Extraction Evaluation task of SemEval 2016 (i.e., TExEval-2 [16]). During sampling, terms that are not present in the WordNet are ignored, because their ground truth hypernyms cannot be extracted.

For each sampled term, we extract hypernym subsequences of different lengths using SubSeq as well as the TopEdge and TopEdge+SVM baselines. To evaluate these subsequences, we also extract a hypernym path for the sampled terms using the WordNet. If multiple hypernym paths are found in WordNet for a seed term, the hypernym path containing the synset "food" is considered as the ground truth.

Table 6.3 shows examples of these hypernym subsequences of different lengths for two terms (i.e., *blintz* and *oat*) as well as the hypernym paths sampled from the WordNet. For a hypernym subsequence S with the corresponding WordNet path W, we compute two scores: (1) *precision*, which is defined as the ratio of number of terms present in both S and W to the number of terms present in S, (2) *recall*, which is defined as the ratio of present in both S and W to the number of terms present in W.

We report four evaluation metrics in this evaluation: (1) *average precision@1* (P@1), which is defined as the average precision of the highest ranked subsequence returned by the subsequence extraction method (i.e., SubSeq or the edge-based baselines), (2) average recall@1 (R@1), which is defined as the average recall of the highest ranked subsequence returned by the subsequence extraction method, (1) average precision@5 (P@5), which is defined as the average precision of the top-5 subsequences returned by the subsequence extraction method, (2) average recall@5 (R@5), which is defined as the average recall of the top-5 ranked

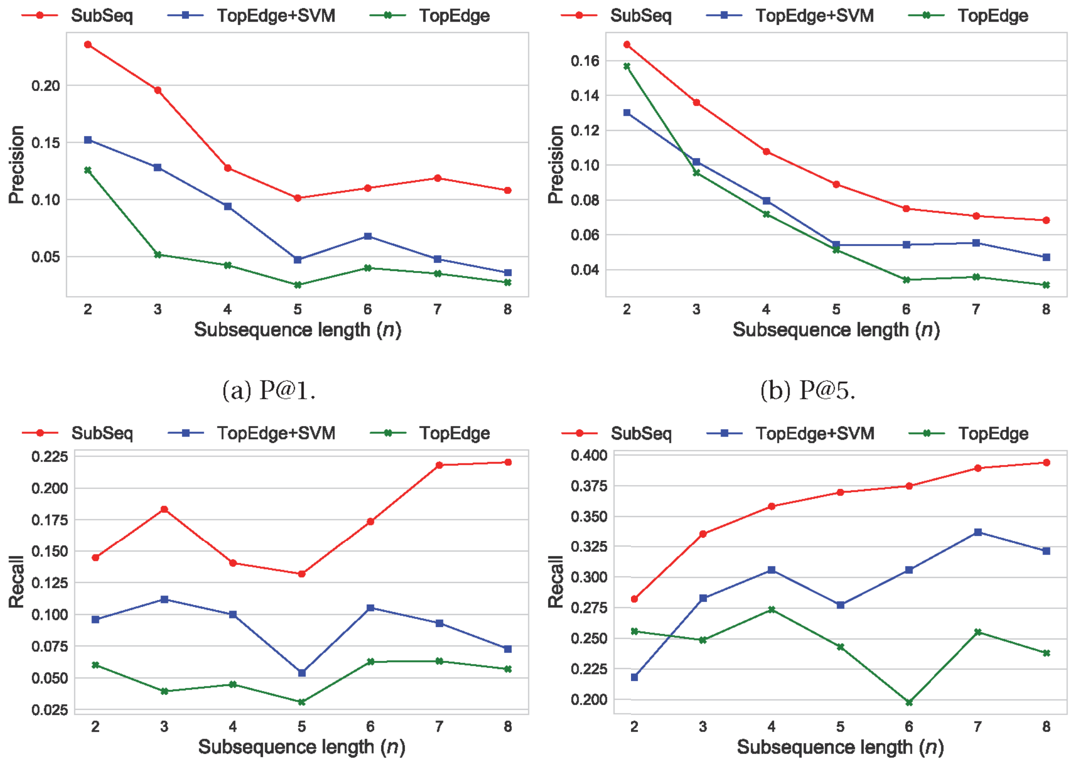
Model	Length (n)	Subsequence
TopEdge	2	blintz→goody
	3	blintz→goody→thing
	4	blintz→goody→ulead→editor
	5	blintz→goody→ulead→social networking →networking→part
	6	blintz→goody→ulead→editor→storyliner→role
	2	oat→food
	3	oat→crop→thing
	4	oat→crop→total loss→partial loss→loss
	5	oat→cereal grain→grain→balanced diet→diet→factor
	6	oat→cereal→industry→field of life→other carrier→carrier
TopEdge+SVM	2	blintz→thin pancake
	3	blintz→homemade jewish food→food→exclusive info
	4	blintz→homemade jewish food→food→exclusive info→stats from the world
	5	blintz→dish→hit→home run→run
	6	blintz→homemade jewish food→food→supply→keyword→beta test→test
	2	oat→cereal grain→grain
	3	oat→cereal grain→grain→supply
	4	oat→cereal grain→grain→balanced diet→diet
	5	oat→cereal grain→grain→balanced diet→diet→factor
	6	oat→cereal grain→grain→supply→keyword→beta test→test
SubSeq	2	blintz→homemade jewish food→food
	3	blintz→homemade jewish food→food→supply
	4	blintz→thin pancake→pastry→snack food→food
	5	blintz→homemade jewish food→food→supply→necessity→thing
	6	blintz→homemade jewish food→food→supply→keyword→beta test→test
	2	oat→cereal grain→grain
	3	oat→cereal grain→grain→supply
	4	oat→cereal grain→grain→complex carbohydrate→carbohydrate
	5	oat→cereal grain→grain→complex carbohydrate→carbohydrate→essential nutrient→nutrient
	6	oat→cereal grain→grain→supply→keyword→beta test→test
WordNet	N/A	blintz→pancake→cake→baked goods→food oat→grain→foodstuff→food

Table 6.3 – Examples of hypernym subsequences of different lengths extracted using TopEdge, TopEdge+SVM and SubSeq approaches. Hypernym paths sampled from the WordNet are also shown.

subsequences returned by the subsequence extraction method. All averages are performed per seed term.

Results. Figure 6.3 shows the comparative values of the evaluation metrics for the three subsequence extraction methods. The results demonstrate that SubSeq consistently outperforms both TopEdge and TopEdge+SVM baselines for all values of subsequences lengths, thus demonstrating the efficacy of our subsequence extraction approach. Furthermore, TopEdge+SVM also consistently outperforms TopEdge, which demonstrates the benefit of using the SVM trained with multiple features over a single feature.

The experimental results also show that as the subsequence length increases, the precision metrics decrease, whereas the recall metrics increase. This effect can be intuitively explained



(a) P@1.

(b) P@5.

(c) R@1.

(d) R@5.

Figure 6.3 – Comparative precision and recall scores for TopEdge, TopEdge+SVM and SubSeq. These scores are computed in an automated fashion using the WordNet as a source of ground truth hypernyms.

by the observation that candidate hypernyms (cf. Table 6.1) usually only contain correct hypernyms up to 3/4 levels of generality. Hence, longer subsequences would typically drift from the original term, thus causing loss of precision. Although the behavior of metrics computed using the highest and the top 5 subsequences is similar, P@5 and R@5 are more smooth compared to P@1 and R@1 metrics as they are averaged over a larger set of values.

Examples presented in Table 6.3 serve in explaining the superior performance of SubSeq over the edge-based baselines. These examples show that the terms in the subsequences returned by TopEdge and TopEdge+SVM baselines start to drift semantically from the seed term as their height in the subsequence increases. For example, the last two terms in the TopEdge subsequence (length=4) for the seed term *blintz* are *ulead* and *editor*, which are completely unrelated to *blintz* or *food*. Similarly, the last two terms in the TopEdge+SVM subsequence are *exclusive info* and *stats from the world*, which are unrelated to *blintz*. However, in contrast, the last two terms in the SubSeq subsequence are *snack food* and *food*, which are correct hypernyms for the seed term *blintz*. While, this drift is also present in some of the subsequences extracted by SubSeq, it is significantly reduced due to the first constituent of

Model	**n=3**		**n=4**		**n=5**	
	Micro	Macro	Micro	Macro	Micro	Macro
TopEdge	0.45	0.42	0.34	0.36	0.16	0.18
TopEdge+SVM	0.61	0.62	0.39	0.37	0.22	0.22
SubSeq	**0.89**	**0.88**	**0.67**	**0.65**	**0.51**	**0.50**

Table 6.4 – Micro-averaged and macro-averaged precision values for subsequences of different lengths extracted by SubSeq and the edge-based baselines.

the Equation 6.2 (i.e., $\Pr(E_k(t_0)|S^i_{t_0})$). A similar effect can be observed for most subsequences up to length 5. As the subsequence length is further increased, precision decreases for SubSeq subsequences as well, as is corroborated by the results in Figure 6.3.

The automated evaluation performed in this section demonstrates that the SubSeq model produces hypernym subsequences, which are significantly more accurate than the edge-based baselines. In the next section, we further corroborate our findings by performing a manual evaluation of the subsequences.

6.3.2 Manual Evaluation

In the previous section, we compared the subsequences extracted by SubSeq with the subsequences extracted by the edge-based baselines. We also plotted the precision and recall values for different subsequences lengths in Figure 6.3. However, it is noteworthy that the absolute values of precision in Figure 6.3 are low (< 0.25). This can be partly attributed to the low coverage of WordNet. Since we perform an automated evaluation using WordNet as a gold standard, it is quite possible that many correct hypernyms are marked as incorrect, just because they are absent from the WordNet. This is further exacerbated by the fact that WordNet typically does not contain noun compounds (e.g., *complex carbohydrate*), thus resulting in lowering of computed precision and recall scores.

Therefore, to mitigate these shortcomings of the automated evaluation, in this section, we perform a direct manual evaluation of the hypernym subsequences returned by different models. To this end, we sample 60 hypernym subsequences (20 per model) for lengths$=3,4,5$ and manually annotate the correctness of each hypernym[2].

Table 6.4 summarizes the results of this evaluation. Micro-averaged and macro-averaged precision for different (model, subsequence length) pairs are reported separately. Similar to the previous evaluation, this evaluation also demonstrates that SubSeq significantly outperforms the edge-based baselines. As expected, the precision decreases as the subsequence length increases. The results of this experiment indicate that the relative precision scores obtained in the automated evaluation correlate well with those computed in the manual evaluation.

[2] Two annotators independently annotated each hypernym. The inter-annotator agreement (Pearson's correlation coefficient) was 93.8%.

Overall, these experiments demonstrate that SubSeq outperforms the edge-based baselines in both automated and manual evaluation. Hence, we safely conclude that SubSeq is a better approach for extracting generalization subsequences. In the next chapter, we demonstrate that the superior quality of hypernym subsequences extracted by SubSeq results in the induction of more accurate taxonomies.

We note that we do not employ the path-level metrics introduced in Section 3.3.3 (i.e., ACPP and ARCPP) for the evaluation of the subsequences, mainly due to the following reasons: (1) the extracted subsequences are usually much shorter than the generalization paths sampled from the Wikipedia taxonomies. (2) Hypernym extraction from unstructured text is prone to noise, which would render ACPP and ARCPP excessively penalizing. (3) The hypernym subsequences are only intermediate results, which are further aggregated and filtered for the induction of final taxonomy. The discussion related to taxonomy induction from the hypernym subsequences is presented in the next chapter.

6.4 Analysis

In this section, we perform a variety of experiments to gain further insights into the SubSeq model. More specifically, in Section 6.4.1, we demonstrate the effects of various features used for the computation of individual edge probabilities. In Section 6.4.2, we study the effect of various parameters on the performance of SubSeq. Finally, in Section 6.4.3, we analyze the effect of the expansion phase employed during the extraction of hypernym subsequences.

6.4.1 Feature Analysis

In Section 6.2.3, we described the different features used by SubSeq and the edge-based baselines for computation of individual edge probabilities. TopEdge+SVM and SubSeq methods use the SVM, which is trained over these features using a manually annotated set of 500 edges (50 terms, 10 edges per term). In this Section, we perform an experiment to analyze the relative performance of these features. To this end, we compute the values of all the features for the set of 500 edges. For each feature, we first sort the edges (per term) by the feature value in descending order. Further, we select top-k edges for varying values of $k \in [1, 10]$, and compute precision@k using the manually annotated set as the ground truth.

Figure 6.4 plots the results of this experiment. Precision@k for edges sorted by the SVM probabilities are also plotted. In general, the count-based features achieve the highest precision, followed by generality-based and string-based features. SVM achieves better performance than all individual features, thus demonstrating its usefulness in computing more accurate edge probabilities. The utility of the SVM was also corroborated by the experiments in the previous section, which demonstrated that TopEdge+SVM consistently outperforms TopEdge. For $k = 10$, all features achieve the same precision, because all the edges per term are selected.

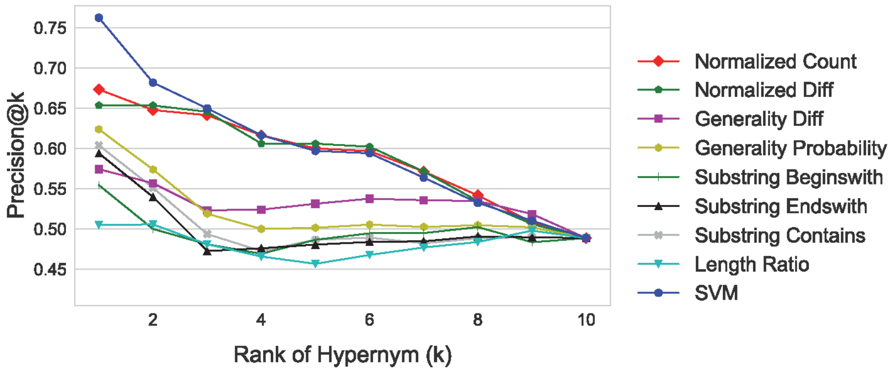


Figure 6.4 – Relative Performance of Features.

6.4.2 Parameter Senstivity

We discussed the effect of the subsequence length parameter (i.e., n in Equation 6.7) in Section 6.3.1. In this section, we discuss the effect of the remaining parameters on the performance of subsequence extraction. Similar to the automated evaluation in 6.3.1, we first construct a gold standard by sampling a set of 100 terms from the food domain randomly and extracting their generalization paths from WordNet. For a given set of parameters, we run the subsequence extraction using SubSeq model and compute the precision, recall and F1 averaged over the top-5 subsequences per term. The most important parameters that we focus on are: the number of hypernyms used (i.e., k in Equation 6.3), and the rank-penalty (i.e., λ_1 in Equation 6.3).

Figure 6.6 shows the effect of the number of candidate hypernyms used (k) for subsequence extraction. As k increases, both precision and recall increase initially but drop afterward, which shows the benefit of utilizing lower-ranked hypernyms for subsequence extraction. However, it also illustrates the significant noise present in candidate hypernyms beyond a certain k. Figure 6.5 shows the effect of rank-penalty (λ_1), the parameter used to penalize candidate hypernyms with lower frequency counts. Both precision and recall are low for lower values of λ_1 and peak at λ_1=0.95.

We also evaluated the sensitivity to other parameters. We found out that subsequence extraction is stable across different values of beam width greater than 20 as well as the length penalty (λ_2). The number of subsequences extracted (i.e., b in Equation 6.3) depends on the use case at hand and is typically set to 5.

6.4.3 Effect of Expansion Phase

We now analyze the effect of the expansion phase (Section 6.2.2), which aims to expand the type-based generalization edges (e.g., *apple→company*) with attribute-based generalization edges (e.g., *apple→american software company→software company→company*).

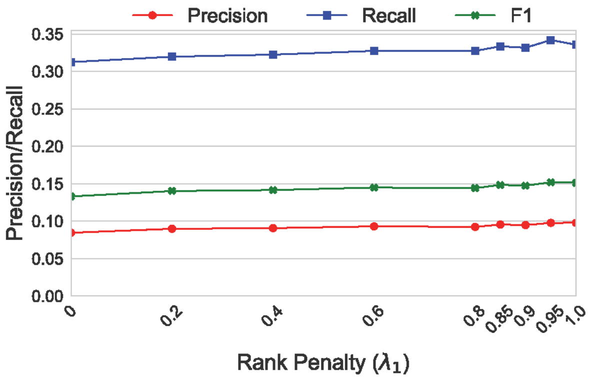


Figure 6.5 – P@5/R@5/F1@5 vs. the rank penalty parameter (λ_1).

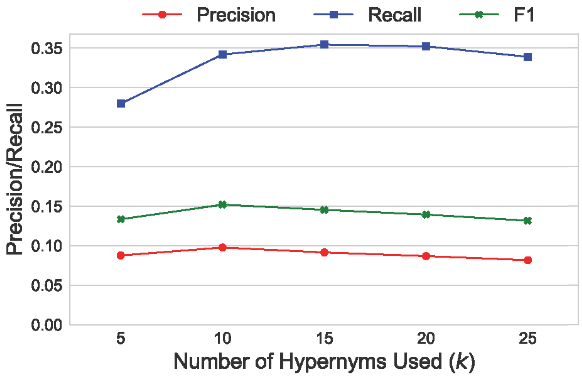


Figure 6.6 – P@5/R@5/F1@5 vs. number of candidate hypernyms used (k).

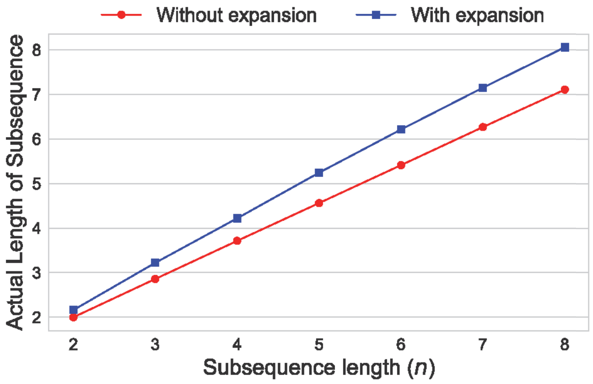
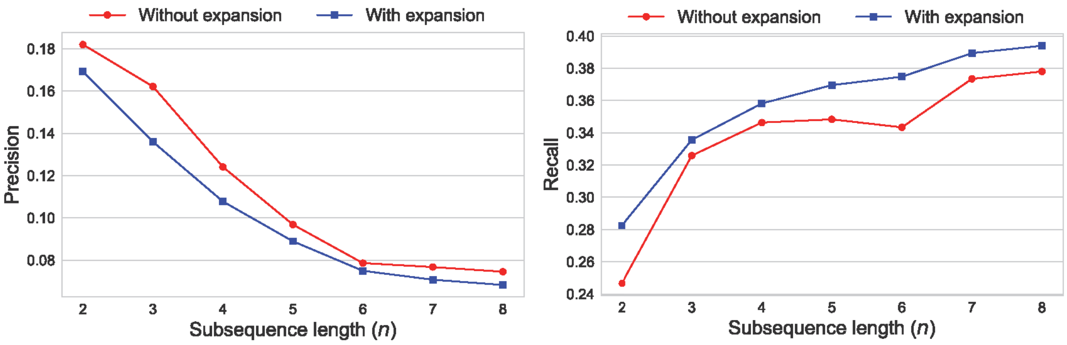


Figure 6.7 – Length of the extracted subsequence with and without the expansion phase.



(a) Precision@5.

(b) Recall@5.

Figure 6.8 – Precision/Recall metrics for subsequences with and without the expansion phase.

In Figure 6.7, we plot the average lengths of extracted subsequences with and without the expansion phase. As expected, the expansion phase typically leads to longer subsequences. Figure 6.8 plots the comparative precision and recall values for subsequences extracted with and without expansion phase. Similar to Section 6.3.1 & 6.4.2, these scores are computed automatically using WordNet as the gold standard. The expansion phase results in lower precision but greater recall, mainly due to the lengthening of subsequences.

However, the lowering of precision can be misleading, because the expansion phase adds noun compound terms (e.g., *american software company*) to the hypernym subsequences, which are usually under-represented in the WordNet. For example, WordNet does not contain most of the noun compound hypernyms shown in Table 6.3 such as *homemade jewish food*, *complex carbohydrate* and *cereal grain*. Therefore, we also perform a manual evaluation to judge the utility of the expansion phase. To this end, we manually annotate the correctness of the expansions of a random sample of 100 edges. An expansion is annotated as correct if all of its edges are correct. Our evaluations show that the expansion phase produces correct expansions for 88% of the edges, thus demonstrating its utility for subsequence extraction.

6.5 Summary

In this chapter, we presented SubSeq, a novel probabilistic model for extracting long-range hypernym subsequences from noisy hypernymy relations that are extracted automatically from unstructured text. Except for a manually annotated set of 500 edges used for computing edge probabilities, SubSeq is fully-unsupervised and runs in an automated fashion. Our experiments demonstrate that SubSeq significantly outperforms equivalent baselines TopEdge and TopEdge+SVM. In the next chapter, we propose an approach that induces taxonomies from the subsequences extracted by SubSeq and demonstrate that it performs favorably against a variety of baselines as well as the state of the art.

7 Taxonomy Induction Using Flow Network Optimization

7.1 Overview

As discussed in the previous chapters, taxonomy induction from unstructured text consists of two main stages: (1) extraction of hypernymy relations from unstructured text, and (2) the structured organization of terms into a taxonomy. In this chapter, we focus on the second stage, namely the structured organization of terms into a taxonomy. We propose a novel approach that casts the task of structured organization of terms as an instance of the minimum-cost flow optimization problem [4, 63]. Unlike previous approaches that assume the availability of a clean vocabulary of input terms (see Section 5.4 for a survey), our approach is specifically designed from the ground up to handle significant noise in the input vocabulary. We describe our approach in detail in the remainder of this chapter.

7.2 Our Approach

Given a potentially-noisy vocabulary of seed terms as input, the goal of our approach is to induce a taxonomy that consists of these seed terms (and possibly other terms). Our approach runs in three phases:

- **Extraction of hypernym subsequences:** in the first phase, hypernym subsequences are extracted for the seed terms in the input vocabulary. The subsequences can be extracted using any of the approaches described in the previous chapter, i.e., SubSeq, TopEdge or TopEdge+SVM (see Section 6.3).
- **Initial graph construction:** in the second phase, a noisy hypernym graph is constructed through the aggregation of the extracted subsequences.
- **Flow network optimization:** in the final phase, the noisy hypernym graph is transformed into a flow network with carefully-designed costs and capacities. An optimal flow is computed over the flow network. The edges with positive flow constitute the final taxonomy.

7.2.1 Initial Graph Construction

The first phase of our approach, i.e., extraction of hypernym subsequences, is already described in detail in the previous chapter. We now describe the second phase of our taxonomy induction approach, namely the construction of a hypernym graph from the extracted subsequences. This phase involves two main steps, which are described hereafter.

Domain Filtering. Given a seed term, the usual case is that multiple hypernym subsequences corresponding to different senses of the seed term are extracted. For example, *apple* can be a company or a fruit, thus resulting in extraction of subsequences *apple→fruit→food* and *apple→software company→company*. However, many of these subsequences will not pertain to the domain of interest, which is usually defined by either the domain-specific corpus or the input vocabulary of seed terms.

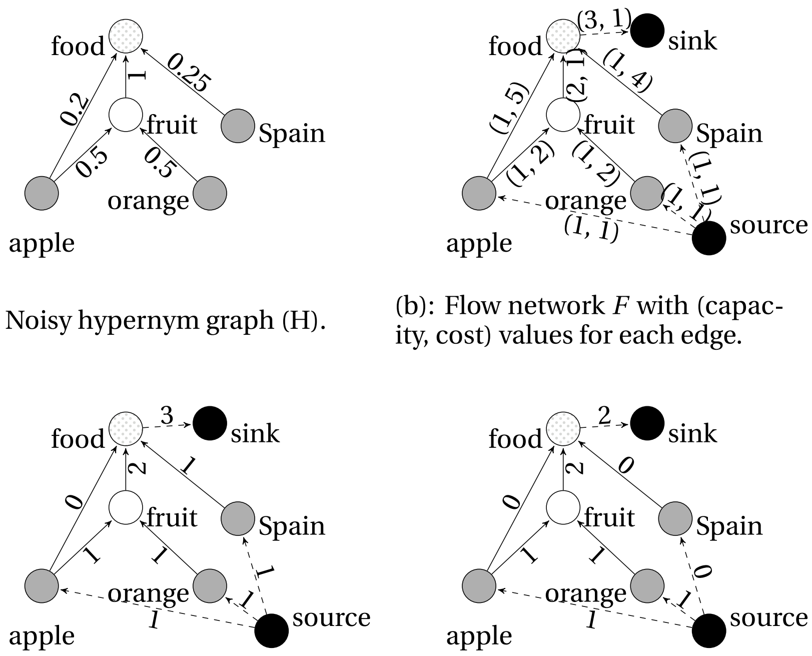
To eliminate such irrelevant subsequences, we first estimate a smoothed unigram model using the vocabulary of seed terms and the hypernym terms in the extracted subsequences[1]. Subsequently, we compute the generation probabilities for each extracted subsequence as the average of the generation probabilities for each hypernym term computed using the unigram model. Finally, we remove all the extracted subsequences that have generation probabilities below a fixed threshold.

Hypernym Graph Construction. In this step, we aggregate the filtered subsequences into an initial hypernym graph. We construct this graph by grouping the edges that have the same start and end terms in all the filtered subsequences. The weight of an edge is computed as the sum of the scores of the filtered subsequences that contain that edge. The score of a subsequence is the same as computed during the extraction phase (i.e., $\log(\Pr(E_k(t)|S) \times \Pr(S))$ for SubSeq, and $\Pr(S)$ for TopEdge and TopEdge+SVM in Equation 6.2).

To increase the coverage for compound seed terms that do not yet have a hypernym, we also add an hypernym edge to their lexical head with weight=∞ (i.e., an extremely large value), whenever the lexical head is already present in the hypernym graph. We use a large weight for such edges, as they tend to be usually correct.

The hypernym graph resulting from the above steps may contain cycles. As shown in the prior literature, the presence of cycles in a hypernymy graph is usually a result of incorrect hypernym edges [132]. Therefore, to remove these cycles, we first detect such cycles using the algorithm proposed in Johnson [56]. Further, for each detected cycle, we remove the edge with the smallest weight. As a result, the initial hypernym graph is transformed into a directed acyclic graph (i.e., DAG). In addition to correct hypernym edges, this DAG also contains many noisy terms and edges, which are pruned in the next step of our approach.

[1] In our experiments, we used a weighting function (i.e., a step function with cut-off at 50% of the height of the subsequence) to favor terms at lower heights as they are usually more domain-specific.



(a): Noisy hypernym graph (H).

(b): Flow network F with (capacity, cost) values for each edge.

(c): Flow values (f) for each edge found using demand $d = 3$.

(d): Flow values (f) for each edge found using demand $d = 2$.

Figure 7.1 – Execution of the minimum-cost flow optimization algorithm for taxonomy induction starting from a noisy hypernym graph.

7.2.2 Flow Network Optimization

In the final phase of our approach, we induce a taxonomy from the noisy hypernym DAG obtained in the previous phase. We cast this task as an instance of the minimum-cost flow optimization problem (usually referred to as ***MCFP***).

MCFP is an optimization problem, which aims to find the cheapest way of sending a certain amount of flow through a flow network. It has been used to find the optimal solution in applications like the *transportation problem* [64], where the goal is to find the cheapest paths to send commodities from a group of facilities to the customers via a transportation network. Analogously, we cast the problem of taxonomy induction as finding the cheapest way of sending the seed terms to the root terms through a carefully designed flow network F. We use the *network simplex algorithm* [99] to compute the optimal flow for F, and we select all edges with a positive flow as part of our final taxonomy. We now describe our method for constructing the flow network F. In what follows, we refer to Figure 7.1 at different steps.

Flow Network Construction. Let V be the vocabulary of input seed terms (e.g., *apple*, *orange*, and *Spain* in Figure 7.1); H is the noisy hypernym graph constructed in Section 7.2.1 (cf. Figure 7.1(a)); $w(x, y)$ is the weight of the edge $x \rightarrow y$ in H; D_x is the set of descendants of term

x in H (e.g., *apple* is a descendant of *food*); R is the set of given roots (e.g., *food* in Figure 7.1). The construction of the flow network F proceeds as follows (cf. Figure 7.1(b)):

1. For an edge $x \to y$ in H, add the edge $x \to y$ in F. Set the capacity (c) of the added edge as $c(x, y) = |D_x \cap V|$, i.e., the number of seed descendants of the term x. Set the cost (a) of the edge $x \to y$ as $a(x, y) = 1/w(x, y)$. This lowers the costs of the edges that have higher weights.

2. Add a sentinel *source* node s. $\forall v \in V$, add an edge $s \to v$ with $c(s, v) = a(s, v) = 1$.

3. Add a sentinel *sink* node t. $\forall r \in R$, add edge $r \to t$ with $c(r, t) = |D_r \cap V|$ and $a(r, t) = 1$.

Minimum-cost Flow. Given a demand d of the total flow to be sent from s to t, the goal of MCFP is to find flow values (f) for each edge in F that minimize the total cost of flow over all edges: $\sum_{(u,v)\in F} a(u, v) \cdot f(u, v)$.

In our construct, demand d represents the maximum number of seed terms that can be included in the final taxonomy. Figures 7.1(c) & 7.1(d) show the minimum-cost flow for the demand d=3 and d=2 respectively. In both cases, the edge *apple→food* receives f=0 due to the presence of edges *apple→fruit* and *fruit→food* with lower costs. For d=2, the edge *source→Spain* has f=0, implying that the noisy term *Spain* would be removed from the final taxonomy.

Intuitively, demand d serves as a parameter for discarding potentially noisy terms in the input vocabulary. More formally, d can be defined as $\alpha|V|$, where α, a user-defined parameter, indicates the desired *coverage* over seed terms. If the vocabulary contains only accurate terms, α is set to 1. For a given α, we run the network simplex algorithm with $d=\alpha|V|$ to compute the minimum-cost flow for F. The final taxonomy consists of all edges with flow > 0.

7.3 Evaluation and Results

In this section, we evaluate the taxonomy induction approach, which is presented in the previous section. The aim of the empirical evaluation is to address the following questions:

- How does our approach compare against the state-of-the-art approaches under the assumption of a clean input vocabulary?

- How does our approach perform on a noisy input vocabulary?

- What are the benefits of extracting longer hypernym subsequences compared to single hypernym edges?

To answer these questions, we perform two experiments. In Section 7.3.1, we compare our taxonomy induction approach against the state of the art, under the simplifying assumption of a clean input vocabulary. Evaluations are performed automatically by computing standard edge-based precision, recall and F1 measures against a gold standard.

We then drop the simplifying assumption in Section 7.3.2, where we show that our taxonomy induction performs well even under the presence of significant noise in the input vocabulary. Evaluation is performed both manually as well as automatically against WordNet as the gold standard. We also demonstrate that the subsequences-based approach significantly outperforms edges-based variants, thus demonstrating the utility of hypernym subsequences.

In the remainder of this chapter, we use ***SubSeq+Flow*** to refer to our approach towards taxonomy induction that uses the SubSeq model followed by the minimum-cost flow optimization.

7.3.1 Evaluation against the State of the Art

Setup. We use the setting of the TExEval-2 task for taxonomy extraction [16]. The task provides six sets of input terminologies, related to three domains (food, environment, and science), for four different languages (English, Dutch, French and Italian), thus totaling a set of 24 (terminology, language) pairs. The task requires participants to generate taxonomies for each (terminology, language) pair, which are further evaluated using a variety of techniques, including comparison against a gold standard. Except for a few restricted resources used to construct gold standard, the participants are allowed to use external corpora for hypernymy extraction and taxonomy induction. Participants are compared against each other and a high-precision string inclusion baseline.

We compare SubSeq+Flow with TAXI [102] (also described in Section 5.4.4), the system that reached the first place in all subtasks of the TExEval-2 task. TAXI utilizes an SVM trained with individual hypernymy edge features extracted from unstructured text, such as frequency counts and substring inclusion to classify edges into *is-a* or *not-is-a*. The edges, which are classified as *is-a*, are added to the taxonomy. Panchenko et al. [102] report that alternate configurations of TAXI with different term-level and edge-level features as well as different classifiers such as Logistic Regression, Gradient Boosted Trees, and Random Forest do not provide any improvements. As a result, the performance of TAXI reflects the collective performance of a wide variety of edge-based taxonomy induction approaches.

Before we proceed with the evaluation results, we perform an additional modification to adapt SubSeq+Flow to the setting of TExEval-2. The TExEval-2 task provides the additional assumption that all the terms in the gold standard taxonomies (i.e., including leaf terms and non-leaf terms) are present in the input vocabulary. This assumption would unfairly lower the performance of SubSeq+Flow, as it would find hypernyms, which are possibly correct but not present in the gold standard. Hence, to ensure a fair comparison, we restrict the subsequence extraction and the hypernym graph construction step of SubSeq+Flow (see Sections 6.2.2

	TAXI			SUBSEQ		
	P	R	F1	P	R	F1
EN	33.2	31.7	32.2	**44.9**	**31.9**	**37.2**
NL	**48.0**	19.7	27.6	42.3	**20.7**	**27.9**
FR	33.4	24.1	27.7	**41.0**	**24.4**	**30.5**
IT	**53.7**	20.7	29.1	49.0	**21.8**	**29.9**

Table 7.1 – Precision (P), Recall (R) and F1 Metrics for TAXI vs. SubSeq across different languages. Results are aggregated over all domains per language.

& 7.2.1) to candidate hypernyms that are present in the input vocabulary. Furthermore, for all languages, we use the same candidate hypernymy relations that are used by TAXI. As a result, TAXI and SubSeq are identical in input data conditions as well as the evaluation metrics and only differ in the core taxonomy induction approach.

Evaluation Results. Table 7.1 shows the language-wise precision, recall and F1 values computed against the gold standard for SubSeq+Flow and TAXI. Aggregated over all domains, SubSeq+Flow outperforms TAXI for all four languages. It achieves >15% relative improvement in F1 for English and 7% improvement overall. Both methods perform significantly better for English, which can be attributed to the higher accuracy of candidate hypernymy relations for English.

Figure 7.2 shows the performance of SubSeq+Flow compared to TAXI and the TExEval-2 baseline across different domains and languages. SubSeq+Flow performs best for food domain, where it outperforms TAXI across all the languages. SubSeq+Flow performs best for English, where it outperforms TAXI across 3/4 domains.

In our experiments, we noticed that SubSeq+Flow achieves the largest improvements when a greater number of hypernym subsequences are found during subsequence extraction. For example, SubSeq+Flow achieves an average 32.23% relative improvement in F1 over TAXI for the food domain, where on an average 0.67 subsequences are found per term, compared to only 0.44 for other domains. Similarly, SubSeq+Flow performs best for the English datasets, where, on an average, 1.09 subsequences are found per term, compared to only 0.32 for other languages.

The variation in the number of extracted subsequences per term can be attributed to two factors: (1) number of terms in the input vocabulary, and (2) number of candidate hypernymy relations available. Due to the assumption that all candidate hypernyms belong to the input vocabulary, larger vocabularies of food domain make it more likely for a candidate hypernym to be found, and hence for a subsequence to be extracted. Similarly, the larger set of available candidate hypernyms for English (~400 million vs. <3.2 million for other languages) makes it more likely for a subsequence to be extracted for English datasets.

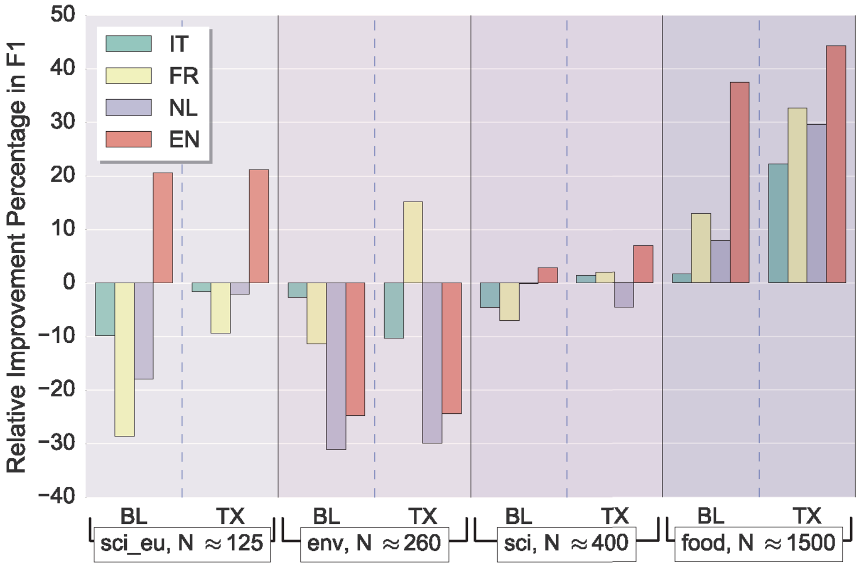


Figure 7.2 – Relative improvement % in F1 for SubSeq, compared to TAXI (TX) and the TExEval-2 Baseline (BL), for different domains and languages. *N* is the average number of terms in the input vocabulary for that domain. *Science eurovoc* datasets are shown separately, as they have significantly fewer input terms than other science datasets.

Overall this experiment shows that under the assumption of a clean input vocabulary, SubSeq+Flow is more accurate than TAXI for most domains in English, and domains with large vocabularies such as food in other languages.

7.3.2 Evaluation with Noisy Vocabularies

In the previous experiment, we performed taxonomy induction under the simplifying assumption that a clean input vocabulary of relevant domain terms is available. In this section, we drop this simplifying assumption and evaluate the performance of SubSeq+Flow in the presence of significant noise in the input vocabulary.

TAXI and most previous taxonomy induction approaches are inapplicable in this setting, as they assume a clean input vocabulary of seed terms. Therefore, instead, we compare SubSeq+Flow against a variety of baselines. Similar to SubSeq+Flow, these baselines also take the seed terms and a required coverage (α) as input. However, they differ from SubSeq+Flow in one of the two aspects: (1) the approach towards subsequence extraction, or (2) the approach towards filtering the noisy hypernym graph. We now describe the baselines in detail:

- **TopEdge+Flow:** instead of the subsequences extracted using the SubSeq model, this baseline employs the subsequences extracted using the TopEdge model. The construction of the noisy hypernym graph and the flow network optimization step are identical to SubSeq+Flow.

- **TopEdge+SVM+Flow:** this baseline is analogous to TopEdge+Flow. However, instead of the TopEdge model, it uses the subsequences extracted by the TopEdge+SVM model.

- **SubSeq+TopWeights:** Similar to SubSeq+Flow, this baseline also employs the SubSeq approach for extracting the subsequences. However, instead of the flow network optimization step for filtering the noisy graph, this baseline simply retains the top $\alpha|E|$ edges with the highest weights, where α is the required coverage, and E is the set of all the edges in the noisy hypernym graph.

  This baseline is specifically designed to assess the efficacy of the flow network optimization step towards filtering of noisy edges. It is important to note that this baseline applies the required coverage parameter (i.e., α) to the set of edges. In contrast, the flow network optimization-based approaches apply the required coverage parameter to the set of seed terms. However, the sizes of the taxonomies induced by SubSeq+TopWeights are similar to other baselines for different values of α.

We also evaluate the quality of input seed terms retained by yet another baseline, which simply selects the top $\alpha|V|$ terms with the highest occurrence frequencies from the Vocabulary V. This baseline is hereafter referred to as **TopTF**.

Setup. We first build a corpus of relevant documents for the food domain. To this end, we collect all English Wikipedia articles with titles that match at least one seed term (post lemmatization) in the English food vocabulary released in the TExEval-2 task. In total, 1,344 matching Wikipedia articles are found from the initial set of 1,555 seed terms. We run *TermSuite* [24], a state-of-the-art term extraction approach to extract an initial terminology of 12,645 terms. Further, we perform two pre-processing steps: (1) first, we remove all terms with occurrence frequencies < 10 in the corpus, (2) we remove all terms that have more than 3 tokens (e.g., *lanterne lasagne lasagnette linguettine*), as they are usually a result errors in the term extraction algorithm. The pre-processing steps result in a final terminology of 1,299 terms.

Table 7.2 shows the top 20 terms from the extracted terminology, which have the highest occurrence frequencies in the food corpus. As the table shows, the extracted terminology contains various noisy terms that are not food items, such as *usage* and *privacy policy*.

We run SubSeq+Flow and all the baselines with varying values of required coverage, i.e., α (Section 7.2.2). For each value of α, we evaluate the output taxonomies on two aspects: (1) quality of the input seed terms retained by the taxonomy; and (2) quality of the taxonomic edges present in the taxonomy. Figure 7.3 shows a section of the SubSeq+Flow taxonomy for α=0.9.

Evaluation Results. Similar to the automated evaluation of extracted subsequences (Section 6.3), we use WordNet to evaluate the quality of the taxonomies induced by the baselines

cake	cuisine	bytes	sausage
usage	portal	node count	hot
noodles	node	preprocessor	flour
oil	cache	lua	related
navigation	privacy policy	isbn	pdf

Table 7.2 – Examples of terms with the highest term frequencies in the automatically-extracted food terminology.

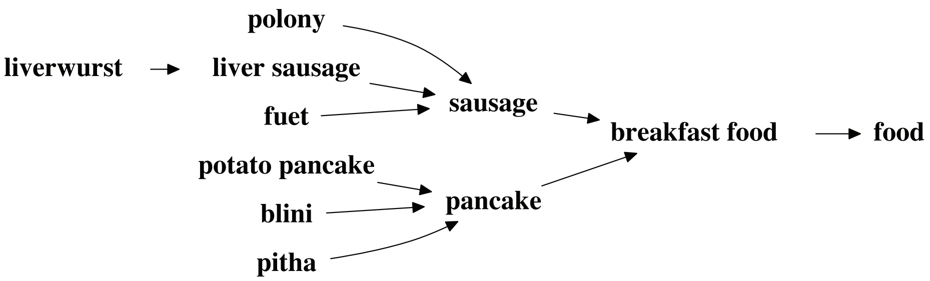


Figure 7.3 – A section of the SubSeq+Flow taxonomy for the food domain (α=0.9).

and SubSeq+Flow. We compare these taxonomies against the sub-hierarchy of WordNet rooted at *food*, which we consider as the gold standard.

We compute two metrics, i.e., *term precision* and *edge precision*. Term precision of a taxonomy is computed for the set of the input vocabulary terms retained by the taxonomy as the ratio of the number of terms in the food sub-hierarchy of WordNet to the total number of terms present in WordNet. Edge precision is computed as the ancestor precision: all nodes from the taxonomy that are not present in the WordNet are removed, and precision is computed on the hypernymy relations from the initial vocabulary to the root. Trivial edges $t \rightarrow$ *food* are ignored for all terms t.

Term Precision. Figure 7.4 reports the term precision of the terms retained by different approaches for varying values of required coverage (i.e., α). In general, all flow network-based approaches significantly outperform SubSeq+TopWeights approach, thus demonstrating the efficacy of the flow network optimization step in the taxonomy induction process. SubSeq+Flow outperforms TopEdge+Flow and TopEdge+SVM+Flow, thus further corroborating the utility of the subsequence extraction model proposed in the previous chapter (cf. Section 6.2.2).

The baseline TopTF underperforms significantly, achieving very low precision for all values of α. This result demonstrates that the occurrence frequency of a term in a domain-specific corpus is not a good indicator of its relevance to the domain.

When all input terms are included in the final taxonomy (i.e., α=1), term precision is 45%,

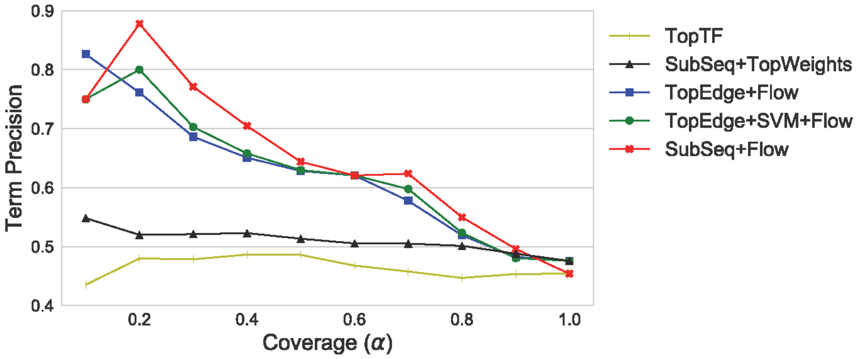


Figure 7.4 – Term precision of the seed terms retained by various approaches vs. the required coverage (i.e., α).

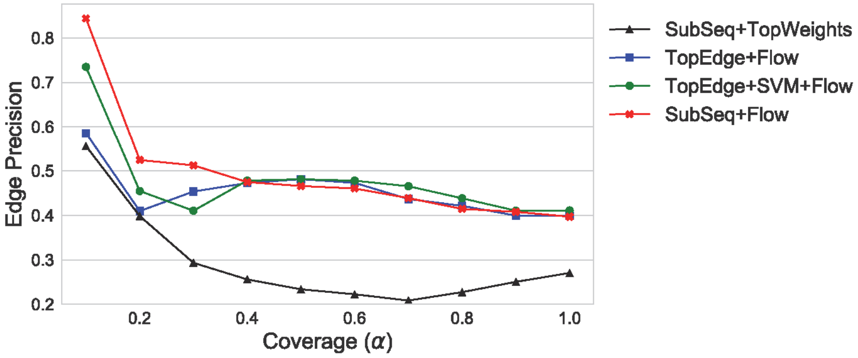


Figure 7.5 – Edge precision (ancestor-level) of the taxonomies induced by various approaches vs. the required coverage (i.e., α).

indicating that only 45% of the terms, which are extracted by the terminology extraction algorithm, belong to the WordNet food sub-hierarchy. In contrast, the term precision for the original seed terms provided by the TExEval-2 task is 75.8%. The large difference in these values confirms the presence of significant noise in the output of the automated terminology extraction approach.

Edge Precision. Figure 7.5 show the edge precision of the taxonomies induced by different approaches for varying values of required coverage, i.e., α. Results demonstrate that the relative performances of different approaches on edge precision is similar to their relative performances in term precision. SubSeq+Flow outperforms all other approaches, whereas flow network optimization-based approaches outperform SubSeq+TopWeights.

For all approaches, both term and edge precision scores decrease with increase in α. This behavior is expected because as α increases additional (potentially-noisy) seed terms and edges are included in the output taxonomies. The value of α can be adjusted manually to control the relative trade-off between precision and coverage of output taxonomies.

Overall, these experiments support the following conclusions: (1) taxonomies induced using the subsequences extracted by the SubSeq model outperform its edge-based counterparts (i.e., TopEdge and TopEdge+SVM), (2) the flow-network optimization results in more accurate selection of seed terms as well more accurate taxonomies, and (3) SubSeq+Flow is an effective approach for taxonomy induction under the presence significant noise in the input vocabulary. In the next chapter, we demonstrate more examples of taxonomies induced using the SubSeq+Flow approach.

7.4 Discussion and Related Work

Taxonomy induction is a well-studied task, and multiple different lines of work have been proposed in the prior literature. The first line of work on taxonomy induction aims to extend the existing partial taxonomies (e.g., WordNet) by inserting missing terms at appropriate positions [133, 120, 141]. The second line of work aims to exploit collaboratively-built semi-structured content such as Wikipedia for inducing large-scale taxonomies [125, 108, 109, 91, 50, 31, 41]. However, as pointed out by Hovy et al. [52] and further discussed in Chapter 2, these taxonomy induction approaches are non-transferable, i.e., they only work for Wikipedia, because they employ lightweight heuristics that exploit the semi-structured nature of Wikipedia content. Although taxonomy induction approaches based on external lexical resources such as WordNet or Wikipedia achieve high precision, they usually suffer from incomplete coverage over specific domains. To address this issue, another line of work focuses on building lexical taxonomies automatically from scratch, i.e., unstructured text present in a domain-specific corpus or Web. Chapter 5 provides a survey of the state of the art of this research direction.

In contrast to taxonomy induction approaches that use external resources, taxonomy induction approaches that use unstructured text typically face three key obstacles. First, they assume the availability of a clean input vocabulary of seed terms. This requirement is not satisfied for most domains, thus requiring a time-consuming manual cleaning of noisy input vocabularies [129]. Second, as discussed in Sections 1.4 & 5.5, these approaches typically require a set of roots as manual input. Third, these approaches ignore the relationship between terms and senses, i.e., they produce term taxonomies. In contrast, taxonomies induced from WordNet or Wikipedia are concept taxonomies, i.e., they have different hypernyms for each sense of a term (e.g., *apple* is a *fruit* or a *company*). To tackle with this obstacle, taxonomy induction approaches from unstructured text employ domain filtering techniques, which perform implicit sense disambiguation by removing the hypernyms corresponding to domain-irrelevant senses of the terms [129]. Although taxonomies should ideally contain concepts rather than terms, term taxonomies have still shown significant efficacy in a variety of NLP tasks [12, 129, 10].

To put it in context, our approach is similar to the previous attempts at inducing taxonomies from unstructured text. However, one key differentiator is that our approach is robust to the presence of significant noise in the input vocabulary, thus dealing with the first obstacle above. We address the second obstacle in the next chapter, where we propose an automated approach for detection of roots. To deal with the third obstacle, our approach performs implicit sense disambiguation via domain filtering at two different steps: (i) domain filtering of subsequences (Section 7.2.1); (ii) assigning lower cost for likely in-domain edges when applying the minimum-cost flow optimization (Sections 7.2.1 & 7.2.2). However, despite the implicit sense disambiguation, we still note that the taxonomies induced by our approach are term taxonomies, and not concept taxonomies.

7.5 Summary

In this chapter, we presented a novel flow network-based optimization approach for inducing a clean taxonomy from hypernym subsequences. Given a potentially-noisy vocabulary of input seed terms, our approach first extracts hypernym subsequences for these seed terms and aggregates them into a noisy hypernym graph (Section 7.2.1). The task of inducing a taxonomy from the noisy hypernym graph is cast as an instance of the minimum-cost flow optimization problem over a carefully-constructed flow network. Our approach provides a control parameter, i.e., required coverage (α), which can be effectively used for regulating the term and edge precision of output taxonomies. The key advantage of our approach is that it is robust to the presence of significant noise in the input vocabulary. However, the approach presented in this chapter still assumes two manual inputs: (1) an input vocabulary, and (2) the roots of the taxonomy. In the next chapter, we further extend the flow network framework to eliminate the need for these manual inputs.

Limitations and Future Work. Similar to past approaches that perform taxonomy induction from unstructured text, the key limitation of our approach is that it induces a term taxonomy rather than a concept taxonomy. An interesting and highly beneficial future work would be to combine our approach with a clustering or synonymy detection approach to combine terms into well-defined concepts. The second interesting future work would be to further explore different configurations of the cost and capacity values within the flow network framework. For example, an alternate configuration could be to use a two-stage approach, i.e., a different flow network design for term selection followed by another flow network or graph optimization algorithm for edge selection.

8 Extensions to the Flow Network Framework

8.1 Overview

In the previous chapter, we proposed a novel flow network optimization-based framework for taxonomy induction given an input vocabulary of seed terms. Empirical experiments demonstrate that our approach not only performs favorably against the state of the art but also is robust to the presence of noisy terms in the input vocabulary. Despite such advancements, our proposed approach still suffers from three limiting constraints:

- **Branching factor of seed terms:** the design of the flow network from the noisy hypernym, which is introduced in the previous chapter (see Section 7.2.2), introduces the constraint that the branching factor[1] of the seed terms be ≤ 1. It is because the capacity of *Source* → *Seed term* edges is set as 1, hence implying that at most one outgoing edge for a seed term can be picked in the final taxonomy.
- **Manually-input root terms:** the flow network optimization step requires a set of root terms as input, which would be connected to the sentinel *sink* node in the constructed flow network (Section 7.2.2). This requirement may not always be satisfied. For example, taxonomy induction is frequently performed from a domain-specific corpus, where the set of roots may not be available beforehand.
- **Fixed vocabulary:** our proposed approach assumes that the input vocabulary of seed terms is fixed. However, it would be desirable if new seed terms can be discovered and integrated into the taxonomy automatically.

In this chapter, we propose three extensions to the flow network optimization framework, which serve to relax the aforementioned constraints. More specifically, in Section 8.2, we introduce a parameter in the flow network optimization framework, which can be used for

[1] We recall that branching factor of a taxonomy is defined as the average out-degree of any node in the taxonomy. Branching factor of a specific node is defined as the out-degree of that node in the taxonomy.

controlling the branching factor of the seed terms in the induced taxonomies. In Section 8.3, we present two different approaches for automatic detection of taxonomy roots given an input vocabulary of seed terms. In Section 8.4, we modify the flow network optimization framework to discover new seed terms and integrate them into the induced taxonomies. Section 8.5 shows examples of taxonomies induced using our approach in a variety of different settings.

8.2 User-defined Branching Factor

8.2.1 Our Approach

In the previous chapter, we cast the problem of inducing a tree-like taxonomy from a noisy hypernym graph as an instance of the minimum-cost flow optimization problem (i.e., MCFP). However, as discussed above, the design of the flow network graph introduces the limitation that at most one outgoing edge per seed term can be picked in the final taxonomy. This is because the capacity of *source* → *seed Term* edges is set as 1 (cf. Section 7.2.2).

In this section, we demonstrate that this limitation can be easily mitigated by making minor modifications in the design of the flow network (cf. Section 7.2.2). More specifically, we introduce a novel parameter b, which serves to control the required branching factor of the seed terms in the induced taxonomy. Given b, we set the capacities of the flow network as follows:

- For all seed terms v, set the capacity of the edge *source*→ v as b.
- For all seed terms, set the capacity of their outgoing edges as 1.
- For an edge $x \rightarrow y$ originating from a non-seed term x, set the capacity (c) of the edge (x, y) as $c(x, y) = b \times |D_x \cap V|$, where D_x is the set of descendants of term x in the noisy hypernym graph, and V is the input vocabulary of seed terms.

The costs of all edges are set in the same fashion as in Section 7.2.2. Figure 8.1(a) shows the design of the flow network using an artificially constructed example. As shown in Figure 8.1(b), when $b = 1$, only one outgoing edge out of *apple* is selected by the flow network optimization algorithm. However, when $b = 2$, both outgoing edges (i.e., *apple*→*fruit* and *apple*→ *tree*) are selected, thus resulting in a higher branching factor.

8.2.2 Evaluation and Results

To evaluate the efficacy of our model for user-defined branching factor, we employ the same setting that was used in the automated evaluation of the induced taxonomies (cf. Section 7.3.2). More specifically, we extract subsequences for 1000 terms, which are randomly sampled from the TExEval-2 English food vocabulary [16]. We construct the initial potentially noisy

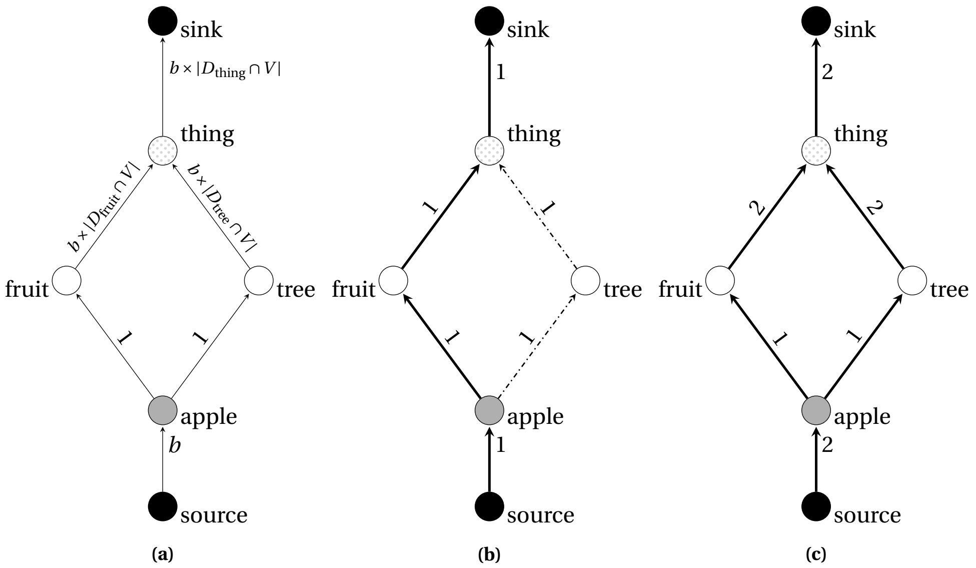


Figure 8.1 – Sample executions of the flow network optimization algorithm for different required branching factors (i.e., the parameter b). (a) An example of a flow network designed for taxonomy induction with branching factor b. The values on the edges represent their capacities. $D(x)$ represents the number of seed descendants of the node x. (b) Execution of the flow network optimization for $b = 1$. The edges shown in bold receive flow > 0, and hence, are selected in the final taxonomy. (c) Execution of the flow network optimization for $b = 2$.

hypernym graph as described in Section 7.2.1. Finally, we construct the flow network for different values of the branching factor parameter b and run the flow network optimization algorithm (i.e., MCFP) for each case.

Table 8.1 shows some examples of hypernyms in the final taxonomies for different values of the parameter b. For $b = 1$, all seed terms have a single hypernym in the induce taxonomy (e.g., *wheatgrass→leafy vegetable*). When b in increased, further hypernyms are added for the seed terms in the final taxonomy. For example, when $b = 2$, the hypernym edge *wheatgrass→complete protein* is added.

It is important to note that b is an indicative parameter, and does not guarantee that exactly b hypernyms will be picked for each seed term. For example, for $b = 3, 4$, the final taxonomy still contains only two hypernyms for the seed term *candy corn* (i.e., *chewy candy* and *halloween candy*). It is because the maximum number of hypernyms, which can be picked for a seed term, are limited by the number of candidate hypernyms in the noisy hypernym graph obtained after subsequence aggregation.

Table 8.2, reports the edge precision, the actual branching factor and the total number of edges in the induced taxonomies for varying values of the parameter b. Similar to Section 6.3, edge precision is computed using WordNet as the gold standard. However, instead of computing ancestor precision, we rather compute precision only using the direct parents. This is to

Seed term	Hypernyms ($b = 1$)	Hypernyms ($b = 2$)	Hypernyms ($b = 3$)	Hypernyms ($b = 4$)
wheatgrass	leafy vegetable	leafy vegetable complete protein	leafy vegetable complete protein superfood	leafy vegetable complete protein superfood
kohlrabi	root vegetable	root vegetable cruciferous vegetable	root vegetable cruciferous vegetable root crop	root vegetable cruciferous vegetable root crop cole crop
yam	root vegetable	root vegetable food crop	root vegetable food crop root crop	root vegetable food crop root crop starchy vegetable
candy corn	chewy candy	chewy candy halloween candy	chewy candy halloween candy	chewy candy halloween candy
tangerine	citrus fruit	citrus fruit fruit tree	citrus fruit fruit tree essential oil	citrus fruit fruit tree essential oil
millet	food crop	food crop ancient grain	food crop ancient grain	food crop ancient grain

Table 8.1 – Examples of hypernyms extracted for different values of the branching factor (b).

b	Edge Precision	Branching Factor	Total Number of Edges
1	0.136	1.15	1011
2	0.128	1.33	1317
3	0.127	1.37	1450
4	0.126	1.53	1470

Table 8.2 – Edge Precision, branching factor and total number of edges of the induced taxonomies for different values of parameter b.

ensure that terms that are hypernyms at different levels of generality are not simultaneously considered as valid hypernyms for the seed terms. For example, only one of *tropical fruit* and *fruit* should be considered as a valid hypernym for the term *apple*.

As expected, as b increases, the edge precision decreases, whereas the branching factor and number of edges increase. The edge precision values are quite low, mainly because they are computed as direct precision. Furthermore, this can also be partly attributed to the low coverage of WordNet. The low coverage of WordNet plays a special role, because most of the additional hypernyms are noun compounds, which are typically missing from the WordNet. We note that a similar effect was also observed during the evaluation of the expansion phase of the SubSeq model (Section 6.4.3).

We also note that the branching factor of induced taxonomies is significantly less than the branching factor parameter (e.g., 1.53 for $b = 4$), mainly due to two reasons: (1) as mentioned above, the maximum number of hypernyms picked for a seed term are limited by the number of candidate hypernyms for the term in the noisy hypernym graph. (2) Our flow network design only affects the branching factor of seed terms. For non-seed terms, taxonomy induction process proceeds in the same fashion as described in Section 7.2.2.

8.3 Automated Root Detection

Up till now, in all our experiments, we assumed that a set of roots are provided to the flow network optimization step. However, this assumption is rarely satisfied in practice. For example, the most common use case of taxonomy induction is inducing a taxonomy from a domain-specific corpus, where a pre-determined set of roots is unavailable. Therefore, in this section, we propose an extension to the flow network optimization step, which aims to detect roots automatically given an input vocabulary of seed terms.

8.3.1 Our Approach

Given an input vocabulary of seed terms, we first extract hypernym subsequences using the SubSeq model. Subsequently, we perform the following steps to detect roots automatically:

- **Selection of initial root candidates:** we first generate an initial set of terms that are likely to be roots for the given input vocabulary. To this end, we aggregate all the hypernym terms in all extracted subsequences and pick the top-k_1 hypernym terms with the highest occurrence frequencies in the extracted subsequences. This set of candidate roots is referred to as C_r.
- **Flow network optimization with root candidates:** second, we run the flow network optimization with the candidate roots (i.e., C_r) as the set of roots that would be connected to the *sink* node (cf. Section 7.2.2). The taxonomy induced as a result of this step is hereafter referred to as T_{C_r}.
- **Selection of final roots:** in this step, we filter the candidate roots to generate the final set of roots. We propose two different approaches for filtering the roots. Each approach takes a parameter k_2 as input, where $k_2 < k_1$, and outputs k_2 roots. These approaches are detailed hereafter:
  1. **Greedy approach:** in this approach, we simply pick the top k_2 roots that have the most number of seed terms as descendants in the induced taxonomy T_{C_r}.
  2. **Beam search:** in this approach, we perform a guided beam search through the space of subsets of candidate roots to determine the most appropriate subset. The steps of this approach are as follows: (1) initiate the beam with the original set of candidate roots (i.e., C_r) as the candidate subset. (2) In each iteration, pick all subsets from the beam, and for each subset, iteratively remove one root to create new subsets. (3) For each subset, compute a fitness function and keep track of the subsets with highest fitness values. (4) Return the subset containing k_2 roots that has the highest fitness value.

> We use the $\frac{\text{Total Cost}}{|\text{seeds}|}$ as the fitness function, where *Total Cost* is the total cost of sending the flow from the seed terms to the root, and |*seeds*| is the number of seed terms that have at least one hypernym in the induced taxonomy. Intuitively, this fitness function selects the subset of candidate roots (i.e., C_r) that minimizes the average cost of sending a seed term to the roots contained in the subset.

We now describe the above process of automated root detection using an example. We first create an input vocabulary of seed terms by randomly sampling the descendants of the WordNet synset *insect*. Some examples of terms from this constructed vocabulary are *bee*, *horse tick* and *oil beetle*. This vocabulary is hereafter referred to as the *insect* vocabulary. We extract the hypernym subsequences for the seed terms in the insect vocabulary using the SubSeq model (cf. Section 6.2.2).

To detect roots automatically, we aggregate the hypernyms in the extracted subsequences and pick the top-k_1 ($k_1 = 10$) most frequent terms as candidate roots (i.e., C_r). In the next step, we run the flow network optimization using the candidate roots as the roots in flow network optimization and compute the number of seed descendants of each root in the resulting taxonomy. Table 8.3 shows the candidate roots along with the number of seed descendants for the *insect* vocabulary.

In the final step of root detection, we filter the candidate roots to output the final set of roots. We take the number of desired roots as an input parameter (i.e., k_2). In the greedy approach, we simply pick top-k_2 roots from the candidate roots, with the highest number of seed descendants. For example, for the insect vocabulary, the terms *insect* and *pest* will be picked as the final roots for $k_2 = 2$ (Table 8.3).

In the beam search approach, we perform a search over the space of subsets of the candidate roots. Figure 8.2 shows a snippet of the subsets of the candidate roots explored during the beam search. Finally, the subset with k_2 roots and the highest value of the fitness function is returned as the final set of roots.

8.3.2 Evaluation and Results

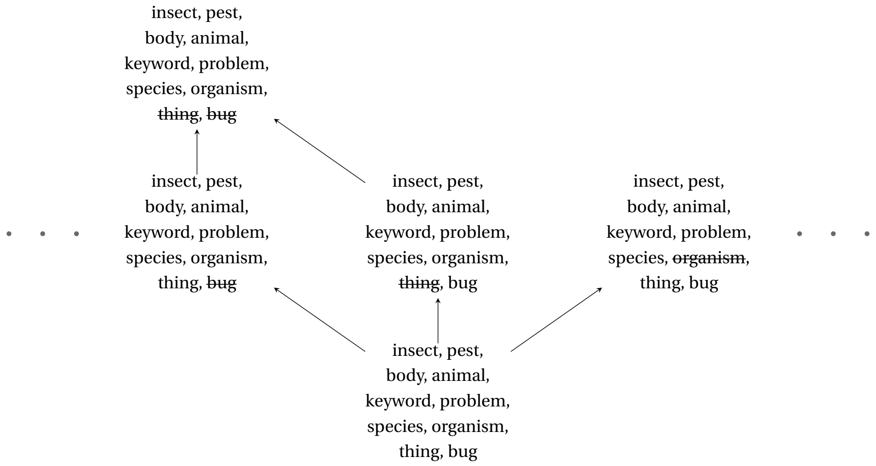
As throughout this thesis, we evaluate our root detection approach in an automated fashion using WordNet as a source of ground truth hypernyms. We proceed in three steps: (1) first, we sample a set of WordNet synsets, which would be considered as the gold standard roots. (2) Second, we sample a set of WordNet descendants of the gold standard roots. These sampled descendants constitute the input vocabulary of seed terms. (3) Finally, we detect roots using the input vocabulary and evaluate the detected roots against the gold standard roots.

We now describe these steps in detail. We first sample a set of 500 synsets from WordNet that are at a height[2] between 3 and 15. This set of synsets is hereafter referred to as W_s.

[2] The height of a synset is computed as its average distance from its descendant leaves in the WordNet.

Candidate Root	Number of Seed Descendants
insect	53
pest	34
body	29
animal	18
keyword	16
problem	12
species	12
organism	12
thing	11
bug	10

Table 8.3 – Candidate roots (i.e., C_r) and the number of seed descendants for the *insect* vocabulary.



insect, pest, body, animal, keyword, problem, species, organism, ~~thing~~, ~~bug~~
insect, pest, body, animal, keyword, problem, species, organism, thing, ~~bug~~
insect, pest, body, animal, keyword, problem, species, organism, ~~thing~~, bug
insect, pest, body, animal, keyword, problem, species, ~~organism~~, thing, bug
insect, pest, body, animal, keyword, problem, species, organism, thing, bug

Figure 8.2 – Beam search through the subset space of candidate roots of the *insect* vocabulary. The roots that are struck out are removed from the corresponding subsets.

Gold Standard Root	Sampled Descendants
niger-kordofanian	swahili, wolof, nyamwezi, songa, swazi, sesotho, kamba, kordofanian, mwera, kongo, yoruba, gikuyu
bird	monal, eurasian woodcock, sheldrake, piping plover, horned screamer, caprimulgiform bird, cream-colored courser, roseate spoonbill

Table 8.4 – Gold standard roots and some examples of their seed descendants, which are sampled from WordNet.

WordNet Roots	Candidate Roots	Detected Roots (greedy)	Detected Roots (beam search)
niger-kordofanian (46), bird (110)	african languages, animal, bantu language, bird, bird species, group, language, species	**bird**, species	**bird species**, **african languages**
aircraft (40), autoloader (23), perception (60), scientist (30)	aircraft, condition, factor, item, keyword, scientist, thing, weapon	**aircraft**, keyword, thing, **weapon**	**aircraft**, condition, **scientist**, **weapon**
classification (20), clothing (91), illness (20), perception (80), sewing (30), tissue (40)	best thing, condition, disease, factor, item, keyword, material, thing, tissue	best thing, item, keyword, **material**, thing, **tissue**	**disease**, factor, item, **material**, thing, **tissue**
carnivore (111), cutter (40), equine (97), food (60), reproductive structure (30), sensitivity (20), separation (20), young (40)	animal, breed, dog, food, horse, item, product, species, thing	**animal**, breed, **dog**, **food**, **horse**, item, product, **species**	**animal**, **breed**, **dog**, **food**, item, product, **species**, thing

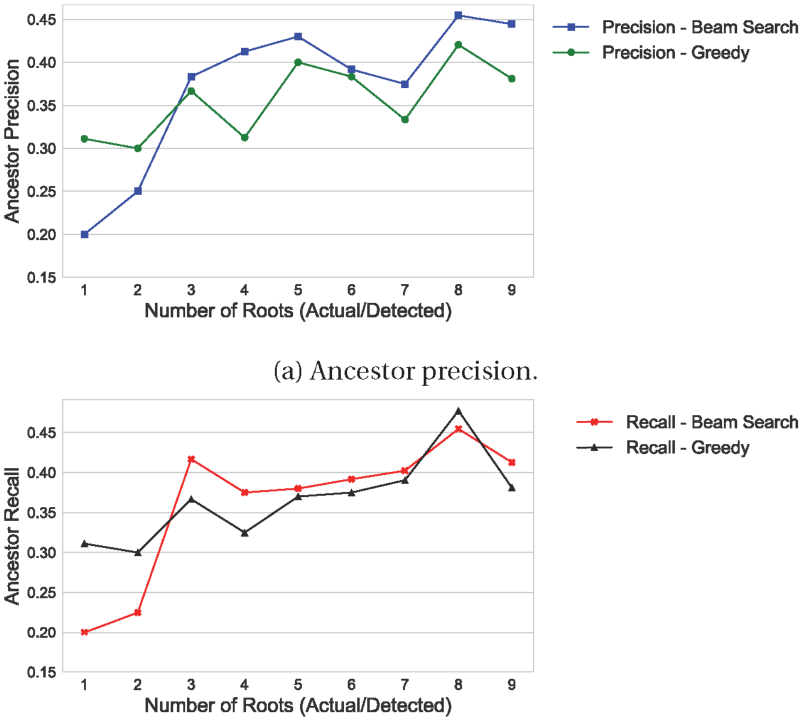
Table 8.5 – Examples of roots detected by greedy and beam search approaches. The number of sampled descendants for each WordNet root are shown in brackets. Detected roots, which are correct, are highlighted in bold.

Given W_s, we randomly sample r synsets from W_s. This set of r synsets would be considered as the gold standard roots. Further, for each gold standard root, we randomly sample n WordNet descendants, where n is randomly varied between $[20, 30, .., 150]$. These sampled descendants are collected and constitute the input vocabulary. We repeat these steps 180 times, i.e., 20 times for each value of $r \in [1, 9]$. As a result, we construct 180 input vocabularies.

Table 8.4 shows an example of a constructed vocabulary for $r = 2$. The gold standard roots, i.e., *niger-kordofanian* and *bird*, and their corresponding descendants are shown. For each vocabulary, we detect roots using both greedy and the beam search-based root detection approach. The desired number of detected roots (i.e., k_2 in Section 8.3.1) is set to the value r, i.e., the number of gold standard roots that were used for the construction of the input vocabulary. Table 8.5 shows some examples of the gold standard roots and the corresponding roots that are detected using the greedy and beam search-based approaches.

Results. To evaluate the quality of the detected roots, we compute two metrics: (1) ***Ancestor precision***, which is computed as the ratio of detected roots that are ancestors of at least one gold standard root in WordNet. (2) ***Ancestor recall***, which is computed as the ratio of gold standard roots that are a descendant of at least one of the detected roots.

Figure 8.3 shows the ancestor precision and recall values for greedy and beam search root detection approaches. Although both approaches achieve similar performance, greedy approach performs better for smaller values of r (i.e., $r \leq 2$), whereas the beam search approach performs better for $r > 2$.



(a) Ancestor precision.

(b) Ancestor recall.

Figure 8.3 – Ancestor precision/recall metrics for greedy and beam search root detection approaches.

We hypothesize that for $r \leq 2$, most of the gold standard roots receive a large number of seed descendants and therefore, these gold standard roots are picked by the greedy approach, hence resulting in better performance. However, when the number of gold standard roots increase, the diversity of seed terms in the vocabulary increases. This diversity results in many gold standard roots ranking significantly lower in the number of seed descendants. These gold standard roots are masked by candidate roots related to other gold standard roots. For example, *pest* receives the second highest number of descendants in Table 8.3, which could potentially mask other gold standard roots. Such gold standard roots are not picked by the greedy approach, hence resulting in its lower performance. In contrast, the beam search approach is better equipped to handle such diversity, as it searches through the space of subsets of the candidate roots. Therefore, it can detect a set of roots that receive a significantly different number of seed descendants.

Figure 8.3 shows that ancestor precision is approximately 0.4 for most values of r, which can be considered low. However, we noticed that in many cases, the gold standard roots used for the construction of the vocabularies are highly technical terms. Such technical terms are difficult to detect automatically, because WordNet is manually constructed by domain experts, whereas our subsequence extraction approach uses candidate hypernyms that are

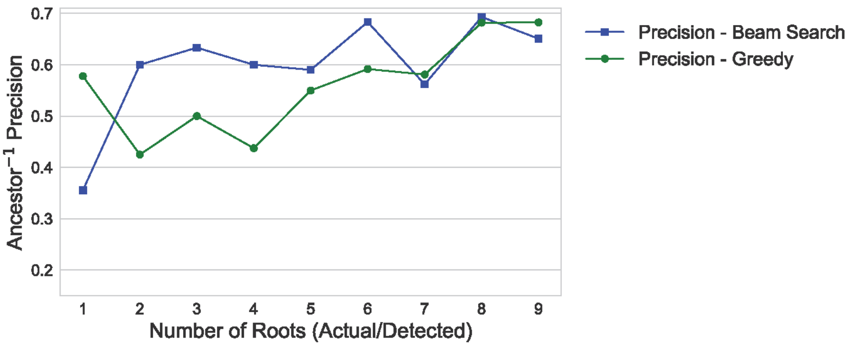


Figure 8.4 – Ancestor^{-1} precision metric for greedy and beam search root detection approaches.

extracted automatically from unstructured text. However, we noticed that in such cases, our approach frequently detects one of the descendants of the gold standard roots as the root. For example, the gold standard root *gramineous plant* from WordNet results in the detection of its child *grass* as a root. To assess this effect quantitatively, we compute an additional evaluation metric ***ancestor^{-1} precision***, which is defined which as the ratio of detected roots that are either ancestors or children of at least one gold standard root in the WordNet. Figure 8.4 plots the ancestor precision of greedy and beam search approaches for different values of r. The absolute value of ancestor^{-1} precision is much higher and stays greater than 0.6 for most values of r, thus showing that the children of gold standard roots are frequently detected by our root detection approach.

8.4 Automated Expansion of Taxonomies

Up till now, in all our experiments, we assumed that the input vocabulary of seed terms is fixed. However, in this section, we demonstrate that we can also extend the flow network framework to expand the input vocabulary automatically by discovering new seed terms. Such an extension would help to increase the coverage of output taxonomies. Hence, it would be useful when a domain-specific corpus is either unavailable or too small to extract a high-coverage vocabulary.

8.4.1 Our Approach

We now describe our approach towards automated expansion of taxonomies. Given an initial vocabulary of seed terms, our approach aims to discover new seed terms that are potentially relevant to the domain specified by the input vocabulary. Our approach runs in a fixed number of iterations. In each iteration, we perform the following steps:

- **Taxonomy induction using current seeds:** in this step, we extract hypernym subse-

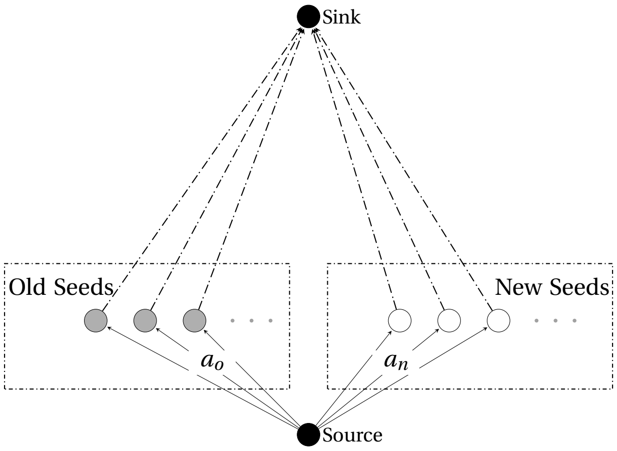


Figure 8.5 – Design of the flow network with old and new seeds. Different costs are used for *source → old seed* edges (i.e., a_o) and *source → new seed* edges (i.e., a_n). $a_n \gg a_o$ indicates that the old seeds will be preferred over the new seeds.

quences for the current seed terms and perform taxonomy induction using the flow network optimization. If this is the first iteration, the input vocabulary is used as the set of current seed terms.

- **Discovery of new candidate seeds:** in this step, we discover new candidate seed terms using the taxonomy induced in the previous step. To this end, we compute top-n candidate hyponyms of all higher-level nodes[3] in the induced taxonomy, and aggregate their counts. These candidate hyponyms are computed using the noisy candidate hypernymy database (see Section 6.1). Further, we sort the hyponyms by the number of occurrence counts in a descending order and pick the top n_d most frequent hyponyms.
- **Update current seeds and demand:** in this step, we add the newly discovered candidate seeds to the set of current seeds. We also increase the value of demand for flow network by $\alpha' n_d$, where α' serves as a parameter for controlling the growth of taxonomy in each iteration.

Our approach introduces two parameters: (1) n_d, which represents the number of new seeds discovered in each iteration, (2) $\alpha' \in [0,1]$, which represents the ratio of the newly discovered seeds that should be included in the output taxonomy. These parameters can be used to control the growth of taxonomy in each iteration. For example, increasing the values of n_d or α' or both, would result in a faster growth of the output taxonomy.

It is noteworthy that in the current approach, the seed terms provided in the input vocabulary are *replaceable*, i.e., they can be potentially replaced by the newly discovered seeds. However, this may not always be desirable. For example, if the input vocabulary is constructed manually or cleaned after extraction, it would be desirable that most of the seed terms in the input

[3] A higher-level node is a node which has at least one child in the induced taxonomy.

Discovered Terms	Discarded Terms	Selected Terms
cassava, maize, cane, tobacco, strawberry, mint, can, salad, if, will	will, mint, can, salad, if	cassava, maize, cane, tobacco, strawberry

Table 8.6 – Examples of terms discovered during the expansion of a *food* taxonomy. Terms that are selected or discarded are also shown.

vocabulary be present in the final taxonomy. To mitigate this issue, we introduce a small modification in the design of the flow network construction. More specifically, we set the costs of *source* → *new seed* edges to be significantly higher than *source* → *old seed* edges. Figure 8.5 shows this modification graphically. As a result, given a certain demand, the new seeds are picked only if it is not possible to further pick any of the old seeds.

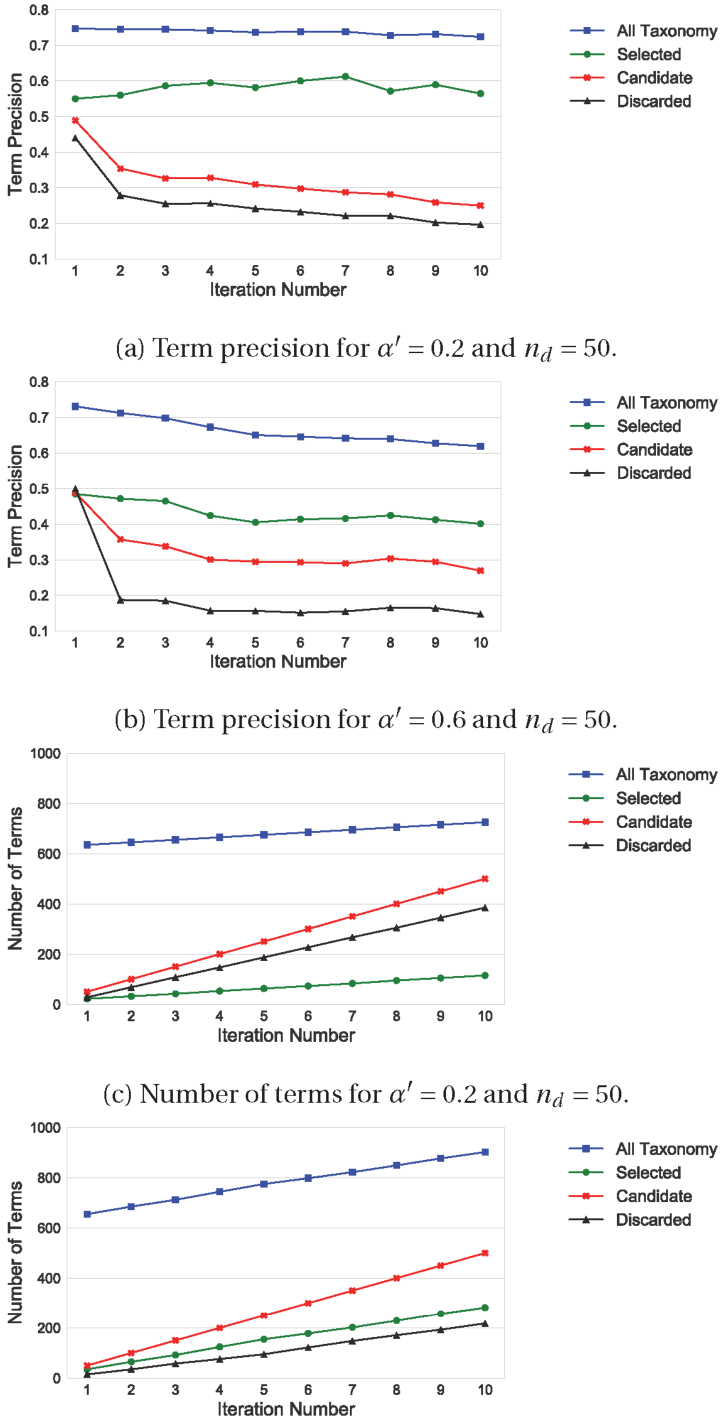
8.4.2 Evaluation and Results

In this section, we perform a quantitative evaluation of our approach for the automated expansion of the taxonomies. We reuse the same setting that was used in the automated evaluation of extracted subsequences (cf. Section 6.3). We start with an initial vocabulary of 600 food terms that are randomly sampled from the TExEval-2 English food vocabulary [16]. Further, we run our taxonomy expansion approach for 10 iterations using two different sets of parameters: (1) $\alpha' = 0.2$ and $n_d = 50$, and (2) $\alpha' = 0.6$ and $n_d = 50$.

Table 8.6 shows some examples of the terms, which are discovered during an iteration, as well as the terms that are selected or discarded. Discarded terms also contain some terms that are valid expansions such as *mint* and *salad*. However, in general, selected terms are significantly more precise (i.e., correct descendants of food) than the discarded terms.

This observation is further corroborated by the term precision values plotted in Figure 8.6(a,b). Similar to Section 7.3.2, the term precision is computed as the ratio of the number of seed terms that are present in the food sub-hierarchy of WordNet to the total number of seed terms that are present in WordNet. For both values of α' (i.e., 0.2 and 0.6), the term precision of the selected terms is significantly higher than both candidate and discarded terms, which shows the effectiveness of our approach in the removal of noisy terms. When a lower value of α' is used (i.e., 0.2), the term precision of the selected terms is significantly higher. As a result, the term precision of the final taxonomy (i.e., all the seed terms in the output taxonomy) also decreases slowly. In contrast, when $\alpha' = 0.6$, the term precision of the selected terms is significantly lower. As a result, the term precision of the final taxonomy also decreases rapidly.

Figure 8.6(c,d) plots the total number of terms in the taxonomy, as well as the number of selected and discarded terms at each iteration. As expected, the taxonomy grows faster for larger values of α', because more discovered seeds are included in the final taxonomy.



(a) Term precision for $\alpha' = 0.2$ and $n_d = 50$.

(b) Term precision for $\alpha' = 0.6$ and $n_d = 50$.

(c) Number of terms for $\alpha' = 0.2$ and $n_d = 50$.

(d) Number of terms for $\alpha' = 0.6$ and $n_d = 50$.

Figure 8.6 – Term precision and number of terms for different values of α' for $n_d = 50$.

These experiments show the functioning of our approach towards the automated expansion of taxonomies. They also demonstrate that the α' and n_d parameters can be used to control the relative trade-off between the precision and the size of output taxonomies. The results of these experiments follow directly from the fact that the SubSeq+Flow model is capable of handling significant noise in the input vocabulary. This noise-robustness allows us to introduce new (potentially-noisy) seed terms using a simple hyponym aggregation technique because they will be further filtered by the flow network optimization step.

8.5 Demonstrations

In this section, we show some examples of taxonomies that are induced using the SubSeq+Flow approach. Since we have already performed extensive quantitative evaluations of SubSeq+Flow, in this section, we restrict ourselves to qualitative discussions. We use SubSeq+Flow under three different settings, which are described hereafter.

Taxonomy Induction from Clean Vocabulary. In this experiment, we perform taxonomy induction under the assumption that a clean input vocabulary of seed terms is available. More specifically, we use a publicly available[4] vocabulary of computer science-related terms, which are extracted from 20 computer science papers. In total, the vocabulary consists of 558 terms. Given this vocabulary, we employ SubSeq+Flow in conjunction with the greedy approach for automated detection of roots (see Section 8.3.1). The top-4 detected roots are *field*, *method*, *technique*, and *information*. Snippets of the induced taxonomy, which are rooted at these detected roots, can be seen in Figure 8.7 (page 122).

Taxonomy Induction from Domain-specific Corpus. In this experiment, we use the most frequently-used setting of taxonomy induction from unstructured text, i.e., taxonomy induction from a domain-specific corpus. To this end, we first extract a corpus of tweets related to the disease *diabetes* through a handful of manually-compiled keywords. Subsequently, we perform terminology extraction from the corpus using TermSuite [24], thus resulting in a vocabulary of 3328 terms. We perform taxonomy induction using the SubSeq+Flow approach (required coverage, i.e., α is set to 0.5) in conjunction with the greedy approach for automated detection of roots. The top-4 detected roots are *food*, *disease*, *condition*, and *thing*. Figure 8.8 (page 123) shows snippets of the induced taxonomy, which are rooted at the detected roots *disease* and *food*.

Taxonomy Induction through Automated Expansion. In this experiment, we demonstrate the automated expansion of the seed vocabulary. However, instead of taking an initial seed vocabulary as input, we rather take a single term as input. This term would be considered the

[4] The WIKI-20 dataset at https://code.google.com/archive/p/maui-indexer/downloads.

root of the induced taxonomy. Given the root term, we construct an initial noisy vocabulary by collecting all the candidate hyponyms of the root term in the candidate hypernymy database. The rest of steps are followed as described in Section 8.4. Figure 8.9 (page 124) shows the full taxonomy induced using these steps, with the term *cancer* as the root term.

Discussion. Although the output taxonomies contain some noisy terms (e.g., *incurable* in Figure 8.8) as well as some overly-generic terms (e.g., *other cancer* in Figure 8.9), the overall quality of the taxonomies is good in all the experiments. The key advantage of SubSeq+Flow is that it is robust to the presence of noise in input vocabulary. This noise-robustness allows us to use automatically-constructed significantly-noisy input vocabularies, hence enabling the application of our approach in a variety of settings such as taxonomy induction from domain-specific corpus as well as a single input term as root.

However, one key issue with these induced taxonomies is that they are term taxonomies. For example, in Figure 8.9, the concept LUNG CANCER is present in the form of two different terms, i.e., *lung* and *lung cancer*. This issue is faced by most approaches that perform taxonomy induction from unstructured text. However, despite this drawback, term taxonomies have been shown to be beneficial in a wide variety of NLP tasks [129, 10].

8.6 Summary

In this chapter, we proposed three extensions to the flow network optimization framework for taxonomy induction, which was introduced in Chapter 7. These extensions serve to provide better control over the taxonomy induction process. The first extension introduces a parameter, i.e., the required branching factor, which can be used to control the relative tradeoff between precision and branching factor of the seed terms (Section 8.2). The second extension enables automated detection of taxonomy roots, thus eliminating the requirement of a manually-input set of roots (Section 8.3). Finally, the third extension demonstrates that a taxonomy can be expanded automatically by the discovery of new seed terms (Section 8.4). An interesting outcome of this extension is the induction of taxonomies given a single term as input (Figure 8.9 with the term *cancer*).

The main advantage of our approach is the robustness towards the presence of noisy terms in the input vocabulary, which enables us to use relatively inaccurate term extraction or collection methods such as automated terminology extraction approaches [24] or noisy candidate hyponyms (Section 8.4). We also note that other approaches such as finding semantically-similar words using word embeddings [85, 104] can also be used for expansion in an equivalent fashion. Overall, our approach facilitates the relaxation of many assumptions employed by previous taxonomy induction approaches including clean and fixed vocabularies of seed terms as well as pre-determined sets of taxonomic roots, thus automating the process of taxonomy induction from unstructured text in the true sense. In the next part of the thesis, we focus on the applications of the taxonomies that are induced using our approaches.

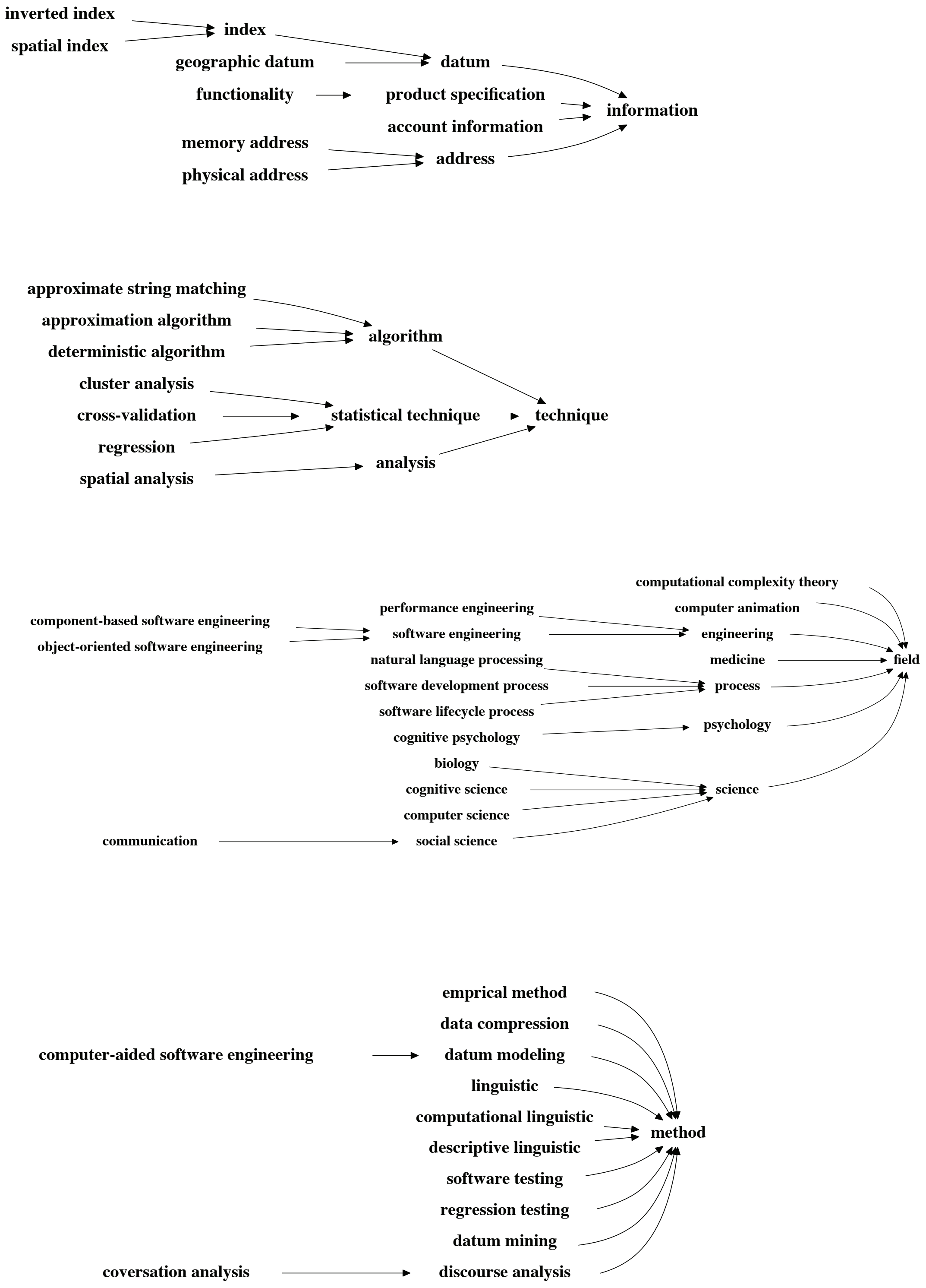


Figure 8.7 – Snippets of the *computer science* taxonomy, rooted at each of the detected roots.

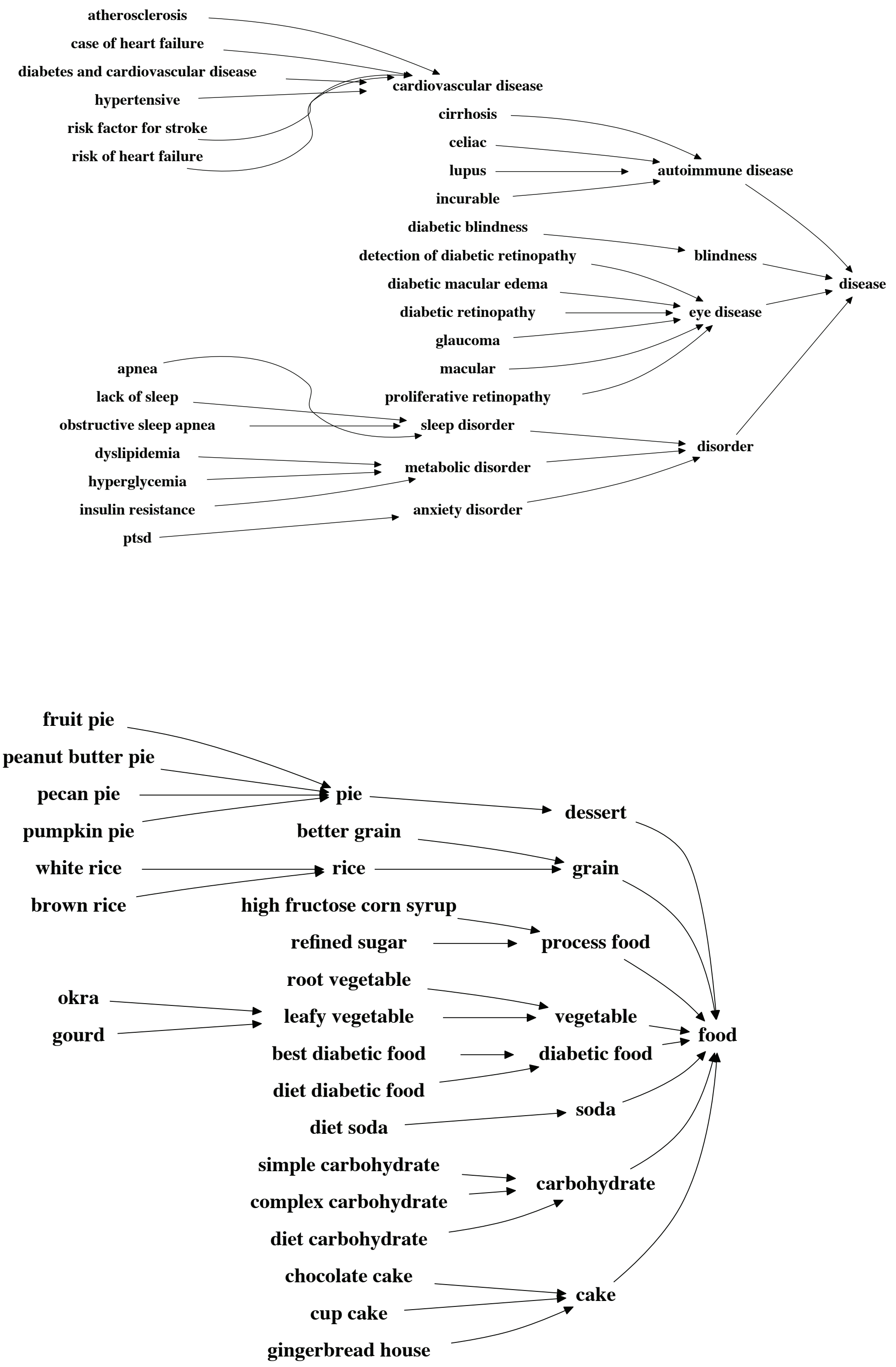


Figure 8.8 – Snippets of the *diabetes* taxonomy, rooted at *disease* and *food.*

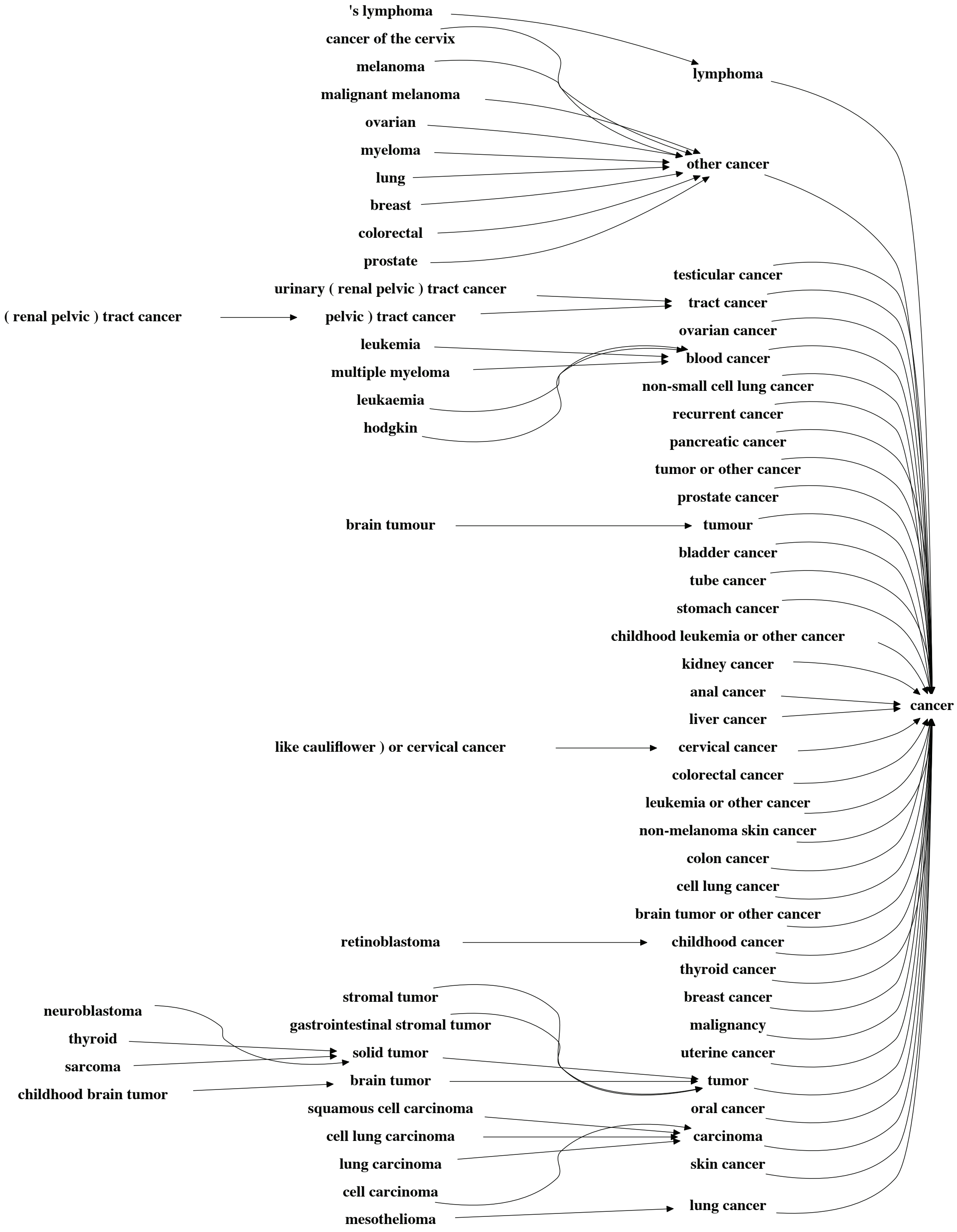


Figure 8.9 – Taxonomy induced using the term *cancer* as root.

Applications of Taxonomies Part III

9 Applications of Taxonomies

9.1 Overview

In the previous parts of this thesis, we focused on the task of taxonomy induction in a variety of different settings. In Chapters 3 & 4, we induced large-scale taxonomies from Wikipedia for English as well as other Wikipedia languages. In Chapters 7 & 8, we proposed a novel approach SubSeq+Flow, which induces taxonomies from unstructured text. We also demonstrated that SubSeq+Flow provides a significant advancement over the state of the art, by relaxing many of the simplifying assumptions that were frequently used by past approaches.

In this chapter, we demonstrate the utility of our approaches through some applications of their induced taxonomies. We first provide a brief survey of past approaches that utilize taxonomies for a variety of NLP-related tasks. Further, we introduce a novel task of mining of generalization templates such as *passport of X*, and demonstrate that the HEADS taxonomy (induced in Chapter 3) can be effectively used for generalizing the fillers in placeholders of such templates (e.g., *X* = COUNTRIES). While we focus on the templates in English, our approach is language-independent and can be easily adapted to any of the Wikipedia languages using our multilingual taxonomies (induced in Chapter 4). Finally, we provide a brief qualitative comparison of the task of finding semantically-similar terms using our taxonomies vs. more frequently-used approaches such as word embeddings.

9.2 Literature Survey

Knowledge in the form of term or concept taxonomies has been shown to benefit a wide variety of NLP-related tasks as well as real-world applications. WordNet is a prime example of a knowledge base, which has been utilized extensively for its taxonomic information [86]. The main utility of taxonomies such as WordNet is that they provide additional semantic features, which augment other textual and context-based features, thus resulting improved performance in many tasks. Detailed surveys of such tasks that benefit from taxonomies can be found in survey papers such as Biemann [12], Hovy et al. [52] and Wang et al. [132].

In this section, we provide a brief overview of these tasks, and also discuss a few of their corresponding approaches that use taxonomies.

Word-sense Disambiguation. Word-sense disambiguation aims to identify the correct senses for a term in a specific context. For example, the term "apple" usually refers to the software company APPLE INC. in the technology domain, whereas to the fruit APPLE in food domain. WordNet is frequently used as a sense-inventory for word-sense disambiguation tasks [129]. Many approaches have demonstrated that WordNet augmented with semantic relations such as hypernymy, which are extracted from Wikipedia, aid in improving the accuracy of word-sense disambiguation systems [106, 92].

Semantic Similarity Between Words. Many NLP tasks including information retrieval and coreference resolution benefit from a quantified measure of semantic similarity between words. Semantic similarity is frequently computed using WordNet as a knowledge base [32]. However, Ponzetto and Strube [108] demonstrated that a taxonomy induced automatically from Wikipedia, through the removal of *not-is-a* edges from the Wikipedia categories network, results in a performance similar to manually-constructed WordNet.

Document Clustering and Classification. The clustering or classification of text documents is typically performed by computing their features vectors and using standard machine learning techniques such as K-means or Naive Bayes over the feature vectors. The augmentation of these features with the information present in taxonomies has been shown to be beneficial for document clustering [53] as well as document classification [121].

Question Answering. Question answering systems are one of the prime examples of real-world applications, which have benefitted from taxonomic information. The most popular example is IBM Watson, a state-of-the-art question answering system, which employs the semantic type information present in publicly available taxonomies such as YAGO for restricting the set of answer candidates [29]. IBM Watson was shown to consistently outperform its human opponents at the task of answering general knowledge-based questions in the game show Jeopardy! [137]. Another example is Snow [121], who demonstrate that features computed from taxonomies result in improved performance of the QACTIS question answering system ([115]).

Information Retrieval. The semantic knowledge present in taxonomies aids in the development of information retrieval applications, which go beyond the traditional bag-of-words model. For example, Liu et al. [75] demonstrates that automatically-induced large-scale taxonomies result in better performance of a nearest neighbor search task over short queries. Chuang and Chien [21] construct a taxonomy of search queries in an automated fashion, and demonstrate that it can be used for web-based IR systems. Demartini et al. [26] presents an entity retrieval system, which utilizes taxonomies induced from Wikipedia.

Named Entity Recognition and Disambiguation. The goal of named entity recognition (or NER) is to identify the mentions of named entities (e.g., JOHNNY DEPP) in raw text. NER is a well-studied task in NLP and has received lots of attention due to its wide-scale applicability in real-life applications such as Web Search [7]. NER is typically performed by identifying the mentions of the entities in text and further classifying them into coarse-grained semantic classes such as PERSON or LOCATION. Taxonomies can be used to add more contextual information (such as fine-grained semantic classes) to the terms in the classification task, thus resulting in improved performance [38, 43]. Another related task is named entity disambiguation, which aims to associate the entity mentions with an appropriate reference from a lexical knowledge base. Bunescu and Pasca [18] use the taxonomic information present in Wikipedia categories network to augment context-based features, resulting in improved performance of named entity disambiguation.

9.3 Generalization Templates

Generalization is a form of inductive reasoning that forms an important part of the human cognitive abilities [79, 48]. The key utility of taxonomies is that they serve a mechanism for generalizing a set of terms or concepts. In this section, we introduce a novel task that aims to assess the quality of generalizations produced by a taxonomy. More specifically, this task aims to discover generalization templates from the titles of Wikipedia entities, and further generalizes the placeholder field to a suitable generalization category.

Before we proceed, we first provide some formal definitions. A ***generalization template*** is defined as a lexicalized linguistic template with placeholders, which can be replaced by suitable ***fillers*** to generate the titles of Wikipedia entities. For example, *Bank of X* is a generalization template with a placeholder *X*, which can generate a title such as “Bank of Switzerland” by the substitution of X with the filler “Switzerland”. A ***prefix template*** contains the placeholder at the beginning (*e.g. X railway station*), whereas a ***suffix template*** contains the placeholder at the end (*e.g. bank of X*).

Given these definitions, the goal of our task is two-fold: (1) discover generalization templates that can be used to generate titles of Wikipedia entities. (2) For each template, select Wikipedia categories that are suitable generalizations for the set of fillers of the template.

To achieve these goals, we perform three steps. In the first step, we discover candidate generalization templates from the titles of Wikipedia entities. We restrict the discovery of generalization templates to prefix templates and suffix templates in English. However, we note that our overall approach is general, and can be easily extended to templates with multiple placeholders as well as other languages.

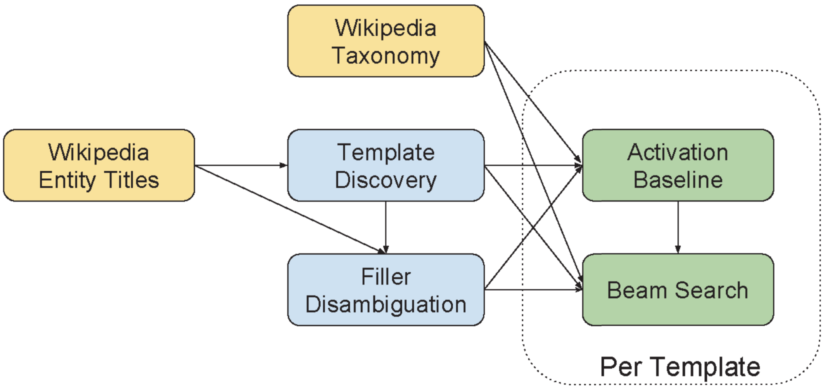


Figure 9.1 – Pipeline for discovery and category selection of generalization templates.

In the second step, we disambiguate the fillers of the discovered templates to Wikipedia entities. In the final step, we select Wikipedia categories that are suitable generalizations of the disambiguated filler entities. To this end, we first compute scores for the Wikipedia categories, by activating categories that are ancestors of the disambiguated filler entities in a Wikipedia taxonomy. This step is hereafter referred to as the ***activations baseline***. Further, we collect all categories that receive positive scores in the activations baseline and employ a beam search-based method to select the set of categories that are the most suitable generalizations. Figure 9.1 summarizes our approach in a graphical fashion. We now describe these three steps in detail.

9.3.1 Template Discovery

We now describe the algorithm for discovering the generalization templates as well as their fillers from the titles of Wikipedia pages. The pseudocode of the algorithm is provided in Algorithm 1. Let TF be the initially empty set of (candidate template, filler) pairs (line 1). The algorithm proceeds in two phases. In the first phase, it iterates over all possible prefix-suffix splits of each Wikipedia page title (line 3-6). For each *valid* split of a Wikipedia title, the algorithm assigns the part containing the lexical head of the title as the candidate template and the other part as the filler (line 7-11). A split is considered *valid* if it satisfies each of the following conditions:

1. Both prefix and suffix of the split should match the title of at least one Wikipedia page after ignoring the stop words and sense disambiguation string in the title. For example, this condition is satisfied for BANK OF INDIA, because a Wikipedia page exists for both *bank* and *india*.

2. Both the prefix and the suffix should contain at least one noun as determined by a POS tagger (e.g., Stanford parser [62]). For example, the split *(At, first sight)* is not a valid split, because “At” is not tagged as a noun.

Algorithm 1: Template Discovery Algorithm

```
Input  : Set of all Wikipedia page titles $T$
Output : Set of discovered templates and their fillers
 1: TF := $\emptyset$
 2: for $p_t \in T$ do
 3:     tokens := tokenize($p_t$)
 4:     for i=1,...,length(tokens) − 1 do
 5:         prefix := concat(tokens[1, $i$])
 6:         suffix := concat(tokens[$i+1$, length(tokens)])
 7:         if valid_split(prefix, suffix) then
 8:             if contains_head(prefix) then
 9:                 TF := TF ∪ (concat(prefix, "$X$"), suffix)
10:             else if contains_head(suffix) then
11:                 TF := TF ∪ (concat("$X$", suffix), prefix)
12: Aggregate TF templates and sort by number of fillers
13: Return templates with number of fillers > $t$
```

Prefix Template	**Sample Fillers**
X railway station	new delhi, kingsland
X river	sierra leone, saint marie
X district	shurugwi, bago
X airport	east london, belleville
X high school	baden, karate
Suffix Template	**Sample Fillers**
battle of X	northampton, santa clara
list of X	codec, redheads
history of X	tennesse, rapid transit
university of X	sucre, queensland
flag of X	sri lanka, las vegas

Table 9.1 – Examples of prefix and suffix templates along with their fillers.

In the second phase, the algorithm group all the (candidate template, filler) pairs in TF by the templates, thus generating generate an aggregate set of fillers for each template (line 12). Finally, the algorithm selects all templates, for which the number of fillers is greater than a fixed threshold t (line 13).

Using this algorithm, we discovered a set of 8674 templates that had at least ten fillers each. The set consists of 5727 prefix templates and 2947 suffix templates. Some examples of these prefix and suffix templates as well as their example fillers are shown in Table 9.1.

9.3.2 Entity Disambiguation

In the previous step, we discovered pairs of generalization templates and their fillers from the titles of Wikipedia pages. However, these fillers are *lexicalized*, i.e., they are present in their raw string forms. In contrast, the taxonomies induced from Wikipedia (such as the HEADS taxonomy or MultiWiBi taxonomies, see Chapter 3) contain entities or categories. Therefore, before such a Wikipedia taxonomy can be employed for generalization, these lexicalized fillers must be mapped to their corresponding Wikipedia entities. However, this task of mapping fillers to Wikipedia entities is non-trivial, because multiple entities with the same string are present in Wikipedia (e.g., BANK and BANK (GEOGRAPHY)).

To disambiguate lexicalized fillers to Wikipedia entities, we use the state-of-the-art approach towards entity disambiguation proposed in Carmel et al. [20]. More specifically, we build an offline database of lists of candidate entities for each lexicalized string by exploiting the titles, redirects and disambiguation pages within Wikipedia. Further, we use this database to compute the probability of an entity given a title string. Finally, for a lexicalized filler, we choose the entity, which has the highest probability given the lexical string of the filler, as the corresponding entity for the filler.

9.3.3 Category selection

In this step, we aim to find a set of Wikipedia categories that provide reasonable, commonsense explanations for the filler entities of a generalization template. For example, for the template *Bank of X*, we wish to pick a category that represents geographic entities such as countries. Since a brute-force search through all possible subsets of categories is computationally intractable, we instead employ a two-step approach. In the first step, we use an activations-based method (i.e., **activations baseline**) to identify the most relevant Wikipedia categories for the given template. In the second step, we use a beam search-based method guided by these activations scores to determine the most suitable subset of the identified categories. We now describe these steps in detail.

Activations Baseline. We now describe the activations baseline approach for computing the scores of Wikipedia categories given a set of entities disambiguated from the fillers. The pseudocode of the activations baseline is provided in Algorithm 2. The algorithm takes as input a taxonomy T, the set of filler entities E_f and the set of all Wikipedia entities E_a. Given these inputs, the algorithm proceeds by first computing the set of ancestor categories (using breadth-first search, or BFS) in taxonomy T for all entities in E_a (line 1). In the second step, the algorithm initializes the set of activations received by filler entities (act$_f$) as well as the set of activations received by all entities (act$_a$) to zero for all categories (line 2). Further, the algorithm runs in a fixed number of iterations n_i (line 3). In each iteration, the algorithm computes a subset of fixed size (i.e., sample_size) from the set of filler entities (i.e., E_f) and updates the activations act$_f$ received by their ancestor categories (line 6-8). Similarly, it computes a subset from the set of all Wikipedia entities (i.e., E_a) and updates the activations act$_a$ of their ancestor categories (line 9-11). This sampling process normalizes contributions by entities to offset possible errors in taxonomy or disambiguation process.

Algorithm 2: Sampled Activations Baseline

```
Input  : Wikipedia Taxonomy T, filler entities $\mathrm{E}_f$, all entities $\mathrm{E}_a$
Output : Set of (category, score) pairs
 1: $\mathrm{A} := \{(e, \mathrm{BFS}(e, \mathrm{T}))\ \forall e \in \mathrm{E}_a\}$
 2: $\mathrm{act}_f := \mathrm{act}_a := \mathrm{scores} := \{(c, 0)\ \forall c \in T\}$
 3: for i=1..$n_i$ do
 4:     $\mathrm{S}_{\mathrm{E}_f} := \mathrm{sample}(E_f, \mathrm{sample\_size})$
 5:     $\mathrm{S}_{\mathrm{E}_a} := \mathrm{sample}(E_a, \mathrm{sample\_size})$
 6:     for all $p \in S_{\mathrm{E}_f}$ do
 7:         for all $c \in \mathrm{A}[p]$ do
 8:             $\mathrm{act}_f[c]$++
 9:     for all $p \in \mathrm{S}_{\mathrm{E}_a}$ do
10:         for all $c \in \mathrm{A}[p]$ do
11:             $\mathrm{act}_a[c]$++
12: $\mathrm{act}_f := \{(c, \frac{v}{n_i})\ \forall (c, v) \in \mathrm{act}_f\}$, $\mathrm{act}_a := \{(c, \frac{v}{n_i})\ \forall (c, v) \in \mathrm{act}_a\}$
13: $\mathrm{score} := \{(c, v * (v - \mathrm{act}_a[c]))\ \forall (c, v) \in \mathrm{act}_f\}$
14: return score
```

The activation scores of categories are averaged over the number of iterations (line 12). The final score of a category is computed as $\mathrm{act}_f \times (\mathrm{act}_f - \mathrm{act}_a)$ (line 13). In this formulation, the first term, i.e., act_f, promotes categories that receive high activations from filler entities. In contrast, the second term penalizes popular categories that would generally receive high activations independent of the given template. The primary intuition behind the algorithm is that categories relevant for a given template should on an average receive higher activations from a subset of filler entities than from a random subset of all entities.

Beam Search. In this step, our aim is to use the activations scores computed in the activations baseline to determine the most suitable subset of categories for a generalization template. To this end, we perform a beam search over the space of subsets of categories that receive positive activations scores. We now describe the method in detail.

We maintain two separate beams[1], i.e., B_p for storing the partial solutions and B_f for storing the final solutions. Initially, the set of all the filler entities E_f is added as a partial solution in the beam B_p. In each iteration, we derive new solutions from the existing solutions in B_p and insert them into B_p as well as B_f.

To derive new solution from an existing partial solution $s_o \in B_p$, we first duplicate s_o, i.e., we copy all the nodes (both pages and categories) of s_o. Further, we select each parent p for each node $n \in s_o$ iteratively, and add it to s_0, thus creating a new candidate solution. For a new solution s_n, which is created through the selection of parent p, we remove all nodes in s_n that are subsumed[2] by p. Score of a solution is computed as average of activations scores of its constituent nodes (activations scores for all entities are set to 0). Finally, we pick the solution

[1] Beams of width=1000 worked well for our development set.

[2] Given a taxonomy T and filler entities E_f, the node n_1 is *subsumed* by the node n_2, if either n_1 is a direct descendant of n_2 in T, or all $e \in E_f$ that are descendants of n_1 are also descendants of n_2.

from B_f that has the highest score as the final set of categories for the generalization template.

9.3.4 Evaluation and Results

Evaluation of categories selection is a complex task due to the significantly large number of Wikipedia categories. Therefore, to perform this evaluation, we make a series of simplifying assumptions. First, we assume that if a category provides a possible explanation for a template p, it must receive more activations from filler entities of p than a random set of entities. In other words, the activations-based score of the category should be greater than 0.

Given this assumption, we first construct a subgraph G_s of the candidate categories as follows: (1) add all categories that receive positive score during the activations-based scoring step as a node in G_s. (2) Add an edge from category $c_1 \in G_s$ to $c_2 \in G_s$, if there exists a path from c_1 to c_2 in the Wikipedia taxonomy. Subsequently, we partition the G_s into its weakly connected components (hereafter referred to as WCC). For each WCC, three expert human judges annotate the most suitable (possibly null) set of categories. Finally, all categories as well as entities, which are descendants of the categories annotated as suitable, form the set of ground truth categories (or entities).

To evaluate a categories selection procedure, we create the set of selected categories (entities) in a similar fashion and compute precision-recall statistics against the ground truth set. We compare the beam search-based approach against two baselines:

- **All-roots:** in this baseline, we simply selects all roots of G_s. By definition, this baseline generates 100% recall, however, at the cost of precision.

- **Greedy:** in this baseline, we perform a downward traversal starting from each root in G_s. During traversal, for each category c, we move to its children only if the sum of activations scores of its children are greater than c. Intuitively, this baseline aims to find local maxima of activations scores in the subgraph G_s.

For these experiments, we use the HEADS taxonomy induced in Chapter 3 as the Wikipedia taxonomy. Figure 9.2 shows the categories with positive activations scores for the template *Highways of X* as well as the category selected by the beam search-based method. Table 9.2 show the precision-recall statistics for categories and entities respectively. As the results show, beam search-based categories selection outperforms both baselines and produces better F1 scores. The all-roots baseline achieves the highest recall, which is expected because all descendants (both categories and entities) are picked.

Table 9.3 shows some examples of generalization templates as well as the categories selected by the beam search approach. An interesting example is *X marina* (shown in Table 9.3), because it shows that our approach is capable of capturing different groups of generalizations of a

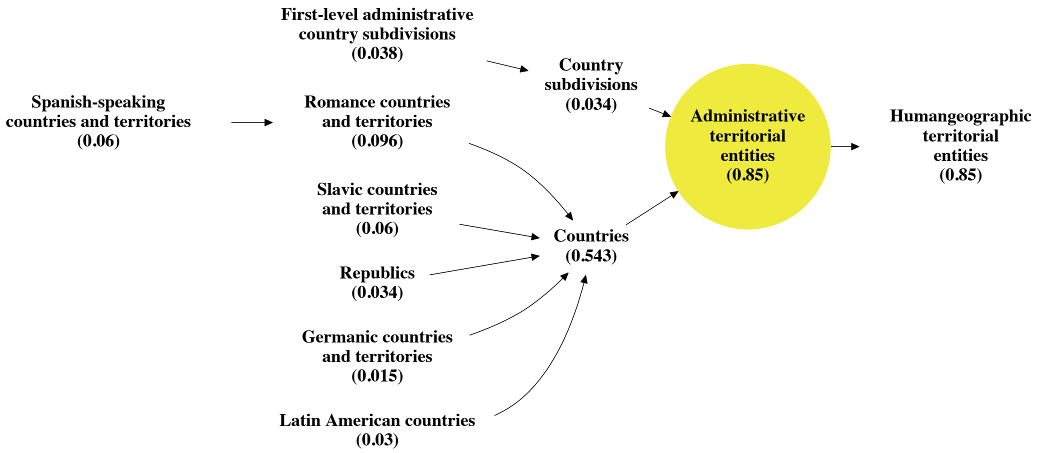

First-level administrative country subdivisions (0.038)
Spanish-speaking countries and territories (0.06)
Romance countries and territories (0.096)
Country subdivisions (0.034)
Administrative territorial entities (0.85)
Humangeographic territorial entities (0.85)
Slavic countries and territories (0.06)
Countries (0.543)
Republics (0.034)
Germanic countries and territories (0.015)
Latin American countries (0.03)

Figure 9.2 – Categories along with their activations scores for the template *Highways of X*. The category ADMINISTRATIVE TERRITORIAL ENTITIES is picked as the final generalization by the beam search-based method.

	Entities			Categories		
Method	P	R	F1	P	R	F1
all-roots	0.38	**1.00**	0.56	0.30	**1.00**	0.46
greedy	0.41	0.74	0.53	0.32	0.78	0.46
beam search	**0.58**	0.70	**0.63**	**0.44**	0.62	**0.51**

Table 9.2 – Evaluation of different approaches for selection of categories.

Template	**Fillers**	**Selected Generalizations**
Railways in X	nepal, plymouth, sydney	administrative territorial entities, populated places
Flag of the X	orange free state, second spanish republic	regions, events, administrative territorial entities, territories, landforms, social groups
X obscura	dysgonia, cynaeda	plants, animal orders, organisms, vertebrates, genera
X marina	brighton marina, amata marina, osney mill marina, najas marina	plants, monotypic taxa, organisms, pollinators, legumes, populated places, administrative territorial entities
Timeline of the X	2007 pet food recalls, samnite wars	events
Casualties of the X	iraq war, ukranian crisis	invasions, disasters, conflicts, human right abuses
Law society of X	scotland, england	administrative territorial entities
X cottage hospital	uxbridge, turriff	populated places
Dancesport at the X	1998 asian games, world games 2005	international sports competitions

Table 9.3 – Lists of selected generalizations computed using the beam search-based method for selection of categories.

template such as places (i.e., POPULATED PLACES[3], ADMINISTRATIVE TERRITORIAL ENTITIES[4]) as well as taxonomic classifications of living entities (i.e., PLANTS, ORGANISMS).

Although we used the HEADS taxonomy for these experiments, theoretically, our approach is compatible with any taxonomy that provides generalizations for Wikipedia entities. However, it would still be desirable that the taxonomy has a good path-level accuracy. It is because, in taxonomies with good path-level accuracy, filler entities (such as FRANCE, SWITZERLAND) would consistently activate the same set of good generalizations (e.g., COUNTRIES). In contrast, a taxonomy, which has lower path-level accuracy, would activate significant noisy generalizations, thus leading to a poor set of selected categories.

To demonstrate this effect, we repeat the category selection step, i.e., activations baseline followed by the beam search-based category selection, for the state-of-the-art MultiWiBi taxonomies, i.e., WIBI$_E$ and WIBI$_C$+H$_E$ (see Section 3.3.3). Table 9.4 (page 140) shows the results of this experiment. A quantitative comparison of these results requires significant annotations, and is outside the scope of this thesis. However, it is immediately clear that the generalization categories obtained by WIBI$_E$ as well as WIBI$_C$ taxonomies are significantly noisy. For example, GENETICS is selected as a candidate generalization for *Tomb of X* by WIBI$_C$+H$_E$, whereas FINE ART is selected by WIBI$_E$. In contrast, the results obtained by HEADS are more accurate, thus demonstrating its superior ability to select meaningful generalization categories for the filler entities.

This task demonstrates that a taxonomy can be used effectively to generate commonsense explanations for a set of related entities. While in this experiment the sets of related entities are discovered using linguistic templates of compound entities from Wikipedia, the overall approach is general and can be extended to many other cases. For example, instead of entity names, the templates can be generated from verb-noun phrases (e.g., *eat X*). Such templates along with our beam-search approach can be used for discovering commonsense knowledge facts (e.g., *birds fly*, *people eat food*) in a fully-automated fashion. Another advantage of our approach is that it is language-independent. Therefore, in conjunction with the multilingual Wikipedia taxonomies (Chapter 4), this approach can be easily extended to all Wikipedia languages. This task also serves to demonstrate the utility of generating taxonomies with higher path-level accuracies (Section 4.3.2), as it results in more accurate sets of generalization categories for the sets of related entities.

During the course of this experiment, we experimented with a wide variety of scoring techniques for category selection (Section 9.3.3). However, we noticed one clear pattern: all scoring methods that used the precision of a category as a feature performed significantly worse than models that ignored the precision. We hypothesize that it is primary because human commonsense reasoning is inherently inductive and approximate. While we reserve a more rigorous analysis of this hypothesis for future work, we consider this as an important

[3] Populated places indicate cities or towns.

[4] Administrative territorial entities indicate regions such as states or countries.

insight for building models that perform human-like commonsense reasoning.

9.4 Word Embeddings vs. Taxonomies

Word embeddings represent a set of language modeling techniques, which are aimed towards finding mathematical vector representations for words or phrases. Intuitively, word embedding techniques perform a mathematical embedding of words (or phrases) from a space with one dimension per word (or phrase) to a continuous vector space with much lower dimensions. One of the key use cases of word embeddings is to discover words (or phrases) that are semantically similar to a given term [85, 104, 35].

In this section, we qualitatively compare the set of semantically-similar terms that are returned by state-of-the-art word embeddings against those returned by taxonomies induced with the SubSeq+Flow approach. To this end, we first manually choose a set of terms across four different languages, i.e., English, French, Dutch and Italian. For each (term, language) pair, we find the most semantically-similar terms as computed using the fastText embeddings [35]. Further, we induce taxonomies using our SubSeq+Flow approach with each term as the root (as performed in Section 8.5). To compute similar terms, we randomly sample a set of terms from the set of direct children and grandchildren (i.e., second-level descendants) of the root term in the induced taxonomy.

Table 9.5 (page 141) shows the results of this experiment, and demonstrates that both approaches perform well in discovering semantically-similar words. A quantitative evaluation of this experiment is inherently complex, and outside the scope of this thesis. However, it is noticeable immediately that the terms output by word embeddings are usually a mix of synonyms (e.g., *havenstad* for *stad*), hyponyms (e.g., *leukemia* for *cancer*) or frequently co-occurring words (e.g., *prostate* for *cancer*). In contrast, the terms discovered by SubSeq+Flow are mostly hyponyms (e.g., *vinh* for *stad*).

Overall, this experiment demonstrates that SubSeq+Flow is a viable approach for discovering semantically-similar hyponym terms for a given term. Furthermore, in comparison with word embeddings, the behavior of the terms discovered by SubSeq+Flow is more well-defined. Therefore, SubSeq+Flow can serve as a complementary approach to Word Embeddings for discovering semantically-similar hyponyms.

9.5 Summary

In this chapter, we focused on the applications of taxonomies in various NLP-related tasks and applications. We first provided a brief survey of the past approaches that utilize taxonomies. Further, we presented our approach for discovering and generalizing linguistic templates from Wikipedia entities such as *Passport of X*. We demonstrated that the entities, which usually replace the placeholder (i.e., *X*), can be generalized to suitable Wikipedia categories using a beam search-based approach for category selection. Our experiments also demonstrate qualitatively that HEADS taxonomy (induced in Chapter 3) results in significantly better generalizations than state-of-the-art MultiWiBi taxonomies. Although in this chapter, we only focused on English linguistic templates for entity names, our approach is general and can be easily extended to other languages as well as other kinds of linguistic templates.

Finally, in the last section, we show examples of semantically-similar terms discovered by SubSeq+Flow across four languages. While we reserve a rigorous quantitative evaluation for future work, the examples demonstrate that quality of these terms is similar to those returned by state-of-the-art word embeddings. Moreover, the terms returned by SubSeq+Flow are more likely to be hyponyms, whereas those returned by word embeddings are usually a mix of many semantic relations such as synonyms, hyponyms, or frequently co-occurring words.

Limitations and Future Work. The work presented in this chapter is still ongoing and many research question remain to be answered. First, the task of generalization templates can be extended to other languages and other kinds of linguistic templates, thus resulting in automated extraction of multilingual commonsense facts. Second, there are many parallels between the category selection approach for generalization templates (Section 9.3.3) and the beam search-based method for automated root detection (Section 8.3). For example, both approaches use a scoring method for scoring categories (or terms) followed by a beam search optimization to pick the right set of generalizations. However, the key difference is that in the former a taxonomy is given, whereas in the latter, a taxonomy is constructed along with root detection. It would be interesting and useful to unify the two approaches under a common conceptual framework. Finally, the comparison of hyponyms detected by taxonomies vs. word embeddings is performed qualitatively. An important future work is to do a more rigorous quantitative evaluation. Another interesting future work could be to train word embeddings using taxonomic information, and compare their performance to original word embeddings.

	Selected Generalizations		
Template	**WiBi**$_E$	**WiBi**$_C$**+H**$_E$	**Heads**
railways in [X]	Tool, Entity, Publication, Operation (mathematics), Property (philosophy), Administrative division, Fine art, Material, Wealth, Combination	Geography, Countries, Statistics, Mathematical and quantitative methods (economics), Least developed countries, Capitals	Cities, Least developed countries, Administrative territorial entities
[X] squads	Playing field, Season (sports), Instruction (computer science), Object (philosophy), Thought, Magic (paranormal), World championship, Physical exercise, 7th Edition (Magic: The Gathering), Championship (professional wrestling), Cycling team, FIFA U-20 World Cup, Competition	Competitions, Association football governing bodies, South American international sports competitions, Sports competitions, Asian international sports competitions, Puerto Rican volleyball clubs, Sports clubs, European rugby league competitions, Sports by type, Hassanal Bolkiah Trophy, Scottish rugby union competitions, Bangladesh Premier League seasons, Events,*[+8 more]*	FIFA World Cup tournaments, UEFA Futsal Championship tournaments, Volleyball clubs, FIFA Confederations Cup tournaments, Copa América tournaments, UEFA European Championship tournaments, FIFA Club World Cup tournaments, Phenomena, AFC Asian Cup tournaments, Rugby World Cup tournaments
forestry in [X]	Entity, Wealth	Muslim-majority countries, Geography, Countries, Statistics, Mathematical and quantitative methods (economics), French-speaking countries and territories, Least developed countries	Muslim-majority countries, Least developed countries, Administrative territorial entities
[X] reader	Economic system, Entity, Document, Property (philosophy), Fine art, Material, Wealth	Philosophical concepts, Branches of philosophy, Concepts in metaphysics, Digital technology, Society, Psychology, Intelligence, Classification systems	Intellectual works, Concepts, Storage media, Literary characters
[X] free zone	Tool, Entity, Publication, Operation (mathematics), Property (philosophy), Fine art, Material, Combination	Public economics, Ports and harbours, Populated places, Economic policy	Ports and harbours, Populated places, Social groups
[X] marina	Tool, Publication, Source code, Sovereign state, Musical technique, Operation (mathematics), Being, Television program, Property (philosophy), Body of water, Fine art, Taxonomic rank, Material, Proteobacteria, Combination, Computer programming, Measure (mathematics)	Landforms, Bodies of water, Countries, Habitats, Transport by mode, Water and politics, Port cities and towns, Angiosperms, Country subdivisions, Water, Taxonomic categories	Eukaryotes, Places, Genera, Administrative territorial entities
tomb of [X]	Tool, Value (mathematics), Publication, Proclamation, Official, Document, Instance (computer science), Fine art, Capital (economics), Material, Aesthetics, Electoral district, *[+2 more]*	People by nationality, Countries by continent, Hebrew Bible people, Ancient people, Religion, Genetics, Behavior, People by occupation, Fields of application of statistics, Jewish priests, Monarchy, Statistics, *[+16 more]*	People, Families, Ethnic groups, Noble titles
[X] obscura	Computer program, Member of the European Parliament, Musical technique, Taxonomic rank, Measure (mathematics)	Amphibians, Euchromiina, Nudariina, Phyla, Concepts in physics, Plants, Animals, Introductory physics	Organisms, Invertebrate taxonomy, Moths, Plants, Animals, Taxonomic categories

Table 9.4 – Lists of selected generalization categories computed using the beam search-based categories selection method over the Heads and MultiWiBi taxonomies.

		Most Similar Terms	
Language	**Input Term**	**Embeddings (FastText)**	**Taxonomies (SubSeq+Flow)**
English	cancer	prostate, leukemia, cancers, colorectal, melanoma, pancreatic, tumour, leukaemia, lymphoma, tumor, cancerous, cancerdr, cancer„ myeloma, preleukemia, ovarian, diabetes, chemotherapy	testicular cancer, ovarian cancer, blood cancer, non-small cell lung cancer, recurrent cancer, pancreatic cancer, squamous cell carcinoma, cell lung carcinoma, neuroblastoma, cell carcinoma
	disease	diseases, diseasel, disease„ disease—, diseasem, disease—a, ‘disease, diseases„ diseas, disease—and, disease—the, diseasing, disease—that, predisease, ‘diseases, infection, disease/ncl, diseased	influenza, haemophilia, hemolytic uremic syndrome, nasopharyngeal cancer, body fatness, norovirus, neoplasm, diabesity, hiv, multiple myeloma, tuberculosis, grippe, salmonella
French	médicament	médicament, médicamens, promédicament, médicamenteurs, médicaments, médicamenteux, médicamenteuse, médicaments, médicaments„ médicaments. . . , médicamenteuses, medicament	paracétamol, sulfamide hypoglycémiant, corticoïde, somnifère, aliment, neuroleptique, anticoagulant, sulfamide, antiinflammatoire, tisane laxatif, benzodiazépine, kératoplastique, viagra
	légume	légumes, légumes. . . , légumineuse, condiment, légumineux, légumes„ légumier, comestible, léguman, condiments, condimentaire, légumineuses, tomates, tomate, potage, fécule, poivron, haricots	ail, artichaut, asperge, aubergine, betterave, blet, brocoli, carde, carotte, champignon, chou, chou-fleur, concombre, cornichon, courge, courgette, céleri, fève, haricot, lentille, navet, oignon, oignons
Dutch	nagerecht	nagerechten, bijgerecht, hoofdgerecht, voorgerecht, vleesgerecht, bijgerechten, roerbakgerechten, lunchgerecht, deeggerecht, visgerecht, vleesgerechten, hoofdgerechten	tiramisu, santo dit, chocoladetaart, jumbo boodschappenmagazine, panforte, panforte panforte, vla, panna cotta, baklava, kolek, mous, vin santo dit, far breton, bavarois, cotta, malvapudding
	stad	steden, havenstad, hoofdstad, stadje, buurstad, stad„ provinciehoofdstad, lakenstad, oblasthoofdstad, prefectuurshoofdstad, stadskern, zakenstad, dorpenstad, modehoofdstad, handelsstad, oudstad	vinh, straat, istanbul, istanboel, lubelski, mar del plata, brussel, gent, san diego, del plata, san fernando, maastricht, podlaski, utrecht, berlijn, brugge, oakland, mazowiecki, zdrój
Italian	insetto	insetti, insettoide, imenottero, insettini, antinsetto, insettario, coleottero, insettari, larve, inset, entomofago, omottero, dittero, insettivore, insettivoro, parassitoide, insettivora, ectoparassita	vespa, bruco, locustere, contro le zanzara, contro zanzara, zecca, formica, mosca, anfibo, zanzara, cimico, afido, apo, artropodo, farfalle, falenere, tarlare, vermo, scarafaggio, libellula
	liquido	liquido„ solido/liquido, liquido/gas, illiquido, fluido, semiliquido, lubrorefrigerante, raffreddante, fluido„ refrigerante, liquidi, liquida, gassosa, raffreddatore, liquefatto, gocciolamento, raffreddava	nel kerosenere, cherosene –, tisana, tè, polvere, birra, bibita, cioccolato, bevanda, inchiostro, kerosenere, all’ acqua, acqua, zavorra, uovo, brodo

Table 9.5 – Examples of semantically-similar terms found using fastText embeddings vs. SubSeq+Flow taxonomies.

10 Conclusion

Machine-readable semantic knowledge lies at the core of the fields of Artificial Intelligence (AI) and Natural Language Processing (NLP). It has been shown to be a key ingredient in building AI that can achieve human-like performance in intelligence-oriented tasks. However, the acquisition of large-scale machine-readable semantic knowledge is not trivial by any means and has inspired a substantial and growing body of research over the last few decades. The earlier work in this direction involved large-scale manual efforts (such as WordNet or Cyc). However, they were quickly deemed insufficient given the vast scale of knowledge, thus paving the way for semi-automated and automated knowledge acquisition approaches.

In this thesis, we focused on the automated acquisition (or induction) of a specific type of knowledge resource, i.e., a taxonomy, which is a collection of *is-a* relations that represent a coherent tree-like hierarchy between terms (or concepts). We addressed two of the most popular settings of automated taxonomy induction, namely taxonomy induction from Wikipedia, and taxonomy induction from unstructured text. In both settings, we proposed novel approaches that resulted in significant improvements over the state of the art. Furthermore, for taxonomy induction from unstructured text, our work also facilitated the relaxation of many simplifying assumptions, which limited the applicability of previous approaches. In the final part of the thesis, we discussed some use cases of the induced taxonomies. The next section provides an overview of the main achievements of this thesis. Section 10.2 proposes possible directions for future work.

10.1 Achievements

The main achievements of this thesis in different tasks are as follows:

- **Taxonomy induction from English Wikipedia:** in Chapter 3, we focused on a specific case of taxonomy induction, i.e., taxonomy induction from the Wikipedia categories network (WCN) in English. We proposed a novel set of heuristics, which exploit the lexical head of Wikipedia categories to pick suitable generalizations for Wikipedia entities and categories. The application of our heuristics results in the induction of a large-scale unified taxonomy (referred to as the HEADS taxonomy) consisting of millions of Wikipedia entities and categories. Our experiments demonstrate that the HEADS taxonomy achieves higher edge-level accuracy than state-of-the-art taxonomies released by MultiWiBi [31]. However, more importantly, our experiments also demonstrate that the generalization paths obtained using our taxonomies are twice as accurate as the MultiWiBi taxonomies, thus indicating a significant improvement over the state of the art. This work also serves to demonstrate that edge-level accuracy of taxonomies may not always correlate well with their path-level accuracy.

A key outcome of this work is the release of HEADS taxonomy[1]. This work also has multiple consequences on the rest of the thesis. First, the HEADS taxonomy is projected to other languages using the interlanguage links, thus leading to the construction of taxonomies in all Wikipedia languages (Chapter 4). Second, the path-level measures (i.e., ACPP and ARCPP) introduced in this chapter are further reused for evaluation of taxonomies across multiple languages (Chapter 4). Finally, HEADS taxonomy is utilized for selection of suitable generalization categories for the fillers of generalization templates of Wikipedia entities (Chapter 9).

- **Taxonomy induction from multilingual Wikipedia:** in Chapter 4, we presented a novel fully-automated approach towards inducing taxonomies from Wikipedia in languages other than English. Given an English Wikipedia taxonomy, our approach leverages the interlanguage links of Wikipedia to project an initial taxonomy in the target language. Training datasets are constructed automatically using the projected taxonomy. Standard text classifiers are trained on the constructed datasets and used in an optimal path discovery framework to induce a high-precision, wide-coverage taxonomy in the target language. Taxonomies induced using our approach outperform the state-of-the-art MultiWiBi taxonomies on both edge-level and path-level metrics across multiple languages. Furthermore, our approach also provides a control parameter for regulating the trade-off between the precision and the branching factor of the induced taxonomies, thus providing better control over the taxonomy induction process. Our approach differs from most previous approaches aimed towards taxonomy induction from Wikipedia in a significant fashion: it does not employ any complex heuristics. As a result, our approach is simpler, principled and easy to replicate.

  A key outcome of this work is the release of our taxonomies across 280 languages, which are significantly more accurate than the state of the art and provide higher coverage.

- **Extraction of hypernym subsequences:** in Chapter 6, we presented a novel probabilistic model (referred to as SubSeq), which extracts long-range hypernym subsequences from noisy automatically-harvested hypernymy relations. Barring a small manually-annotated set of hypernymy edges, SubSeq is fully-unsupervised and runs in an automated fashion. SubSeq captures the intuition that more accurate hypernyms for general

[1] HEADS taxonomy is available at http://headstaxonomy.com.

terms (such as *fruit*) can be extracted by utilizing the candidate hypernyms of its descendants (such as *apple* or *banana*). Furthermore, empirical evaluation demonstrates that SubSeq significantly outperforms multiple baselines, thus resulting in the extraction of more accurate hypernym subsequences. The utility of SubSeq is further demonstrated in Chapter 7, where it is shown that the subsequences extracted by the SubSeq model result in the induction of more accurate taxonomies.

- **A flow network optimization-based framework for taxonomy induction:** in Chapter 7, we presented a novel flow network-based optimization approach for inducing a clean taxonomy from a noisy hypernym graph. The noisy hypernym graph is constructed through the aggregation of hypernym subsequences extracted for the seed terms in the input vocabulary. The task of taxonomy induction from the noisy hypernym graph is cast as an instance of the minimum-cost flow optimization problem over a carefully-designed flow network. Our experiments demonstrate that our approach outperforms a state-of-the-art taxonomy induction system, i.e., TAXI, across multiple languages in the TExEval-2 task of taxonomy extraction [16, 102].

  The key advantage of our approach is that the design of the flow network provides for a control parameter, i.e., required coverage (α), which can be modulated to control the ratio of input seed terms that would be present in the final vocabulary. As a result, our taxonomy induction approach is robust to the presence of significant noise in the input vocabulary. This noise robustness has far-reaching consequences, because it eliminates the need for a time-consuming manual cleaning step of input vocabularies, thus automating the process of taxonomy induction in the true sense.

- **Extensions to the flow network framework:** in Chapter 8, we extended the flow network optimization-based framework to enable better control over the taxonomy induction process. First, we introduced a new parameter that can be modulated to control the relative tradeoff between precision and branching factor of the seed terms in the output taxonomies. Second, we proposed two approaches aimed towards automated detection of roots of the taxonomy, thus eliminating the requirement of a manually-input set of roots. This extension is a considerable improvement over the state of the art because most previous approaches assumed the availability of a pre-determined set of roots. Finally, we presented an extension, which automatically discovers new seed terms given an initial vocabulary. This extension leads to the induction of taxonomies given a single root term as input. This extension is essentially enabled by the noise-robustness of the flow network optimization-based taxonomy induction, which allows us to use relatively inaccurate term extraction or collection methods for construction of seed vocabularies. Overall, these extensions further help in relaxing many of the simplifying assumptions, which limited the applicability of prior taxonomy induction approaches.

- **Applications of taxonomies:** in Chapter 9, we focused on the applications of automatically-induced taxonomies. We introduced a novel task, which aims towards the discovery of suitable generalizations for the placeholder in lexicalized templates (e.g., *X* is the placeholder in *Passport of X*). We first discover such lexicalized templates from the titles Wikipedia entities. Further, we demonstrate that the set of entities, which replace the placeholder in a template, can be generalized to suitable Wikipedia categories using a Wikipedia taxonomy and a beam search-based approach for category selection. We also demonstrate qualitatively that the HEADS taxonomy results in the selection of more appropriate generalization categories than the state-of-the-art MultiWiBi taxonomies. Finally, we showed some examples of semantically-similar terms, which are discovered by the taxonomy induction approach presented in Chapters 7 & 8. A qualitative comparison suggests that our taxonomy induction approach might be more effective than word embeddings for computing semantically-similar hyponyms.

10.2 Future Work

The field of automated taxonomy induction is quite challenging, with a large number of unaddressed issues and open questions. While we consider the work done in this thesis as an important advancement towards the field, there are still a variety of issues and challenges in taxonomy induction that need to be addressed. In the remainder of this section, we identify a few of these issues and propose possible directions for future work:

- **Use-case based evaluation metrics:** in this thesis, we proposed path-level metrics as an alternative measure for assessing the quality of taxonomies and demonstrated qualitatively that higher path-level accuracy results in better performance on the task of generalizing a set of entities (Chapters 3 & 9). In the prior work, a variety of evaluation measures for taxonomies have been introduced [16], which evaluate different aspects of taxonomies including structural properties and accuracy. However, despite so many evaluation measures, the relationship between the performance on these measures and performance in external tasks is not clear. It would be useful if the relationship between such evaluation measures and the utility of the taxonomy in external tasks can be characterized and quantified. This line of work is especially important because taxonomies are intermediate resources, and mainly beneficial through their utility in external applications.
- **Continuous representations:** one of the biggest disadvantages of taxonomies is that they are discrete representations of knowledge. This is in contrast with word embeddings, which provide continuous vector representations for words or phrases. The continuous nature of their representations allows word embeddings to be used directly in a wide variety of machine learning models such as neural networks. While there is some recent work towards inducing continuous representations of words that incorporate taxonomic information [95], it is in its nascent stages. However, still, this research direction is promising and consequential, as it would serve to further expand the applicability of taxonomies.
- **Different types of hierarchies:** in this thesis, we focused on a specific type of term hier-

archy that contains *is-a* relations between terms or concepts. However, many different kinds of hierarchies exist and can be beneficial for many intelligence-oriented tasks. For example, Harkous [45] demonstrates that a state-of-the-art question answering system, which answers questions related to privacy policies, can be built using a hierarchy of topics related to privacy.

Another limitation of our work is that it largely focuses on noun phrases as terms or concepts, due to the designs of the vocabulary extraction and candidate hypernymy extraction approaches. However, taxonomies can be induced on other linguistic units such as adjectives, verbs or relational phrases. For example, Grycner et al. [37] induces a taxonomy of relational phrases instead of unitary terms and demonstrates its utility in the document retrieval task. Expansion of taxonomy induction approaches to such linguistic units would benefit many NLP applications. Another interesting but challenging research direction could be the induction of taxonomies for non-linguistic information types such as images or videos. Research efforts in such directions would lead to more widespread applications of generalization knowledge and further enhance the capabilities of artificially intelligent systems.

- **A unified approach:** in this thesis, we presented a wide variety of approaches towards taxonomy induction under different settings. While some general principles were repeated, the taxonomy induction methods still diverged significantly across different settings. A very important and beneficial future work is to unify the ideas presented in this thesis into a comprehensive framework that performs taxonomy induction from a possibly-multilingual heterogeneous set of resources. Such unified approach in conjunction with continuous representations of taxonomies would facilitate much wider applicability of taxonomies for intelligence tasks.

Bibliography

[1] Steven S. Aanen, Damir Vandic, and Flavius Frasincar. Automated product taxonomy mapping in an e-commerce environment. *Expert Syst. Appl.*, 42(3):1298–1313, 2015. doi: 10.1016/j.eswa.2014.09.032. URL https://doi.org/10.1016/j.eswa.2014.09.032.

[2] S. F. Adafre and Maarten de Rijke. Finding Similar Sentences across Multiple Languages in Wikipedia. *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 62–69, 2006.

[3] David Ahn, Valentin Jijkoun, Gilad Mishne, Karin Müller, Maarten de Rijke, and Stefan Schlobach. Using wikipedia at the TREC QA track. In *Proceedings of the Thirteenth Text REtrieval Conference, TREC 2004, Gaithersburg, Maryland, USA, November 16-19, 2004*, 2004. URL http://trec.nist.gov/pubs/trec13/papers/uamsterdam.qa.pdf.

[4] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network flows - theory, algorithms and applications*. Prentice Hall, 1993. ISBN 978-0-13-617549-0.

[5] Daniele Alfarone and Jesse Davis. Unsupervised learning of an IS-A taxonomy from a limited domain-specific corpus. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1434–1441, 2015. URL http://ijcai.org/Abstract/15/206.

[6] Luis Espinosa Anke, Horacio Saggion, Francesco Ronzano, and Roberto Navigli. Extasem! extending, taxonomizing and semantifying domain terminologies. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 2594–2600, 2016. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12219.

[7] Javier Artiles, Satoshi Sekine, and Julio Gonzalo. Web people search: results of the first evaluation and the plan for the second. In *Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008*, pages 1071–1072, 2008. doi: 10.1145/1367497.1367661. URL http://doi.acm.org/10.1145/1367497.1367661.

[8] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A nucleus for a Web of open data. In *Lecture Notes in Computer*

Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 4825 LNCS, pages 722–735, 2007.

[9] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. Dbpedia: A nucleus for a web of open data. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007.*, pages 722–735, 2007. doi: 10.1007/978-3-540-76298-0_52. URL https://doi.org/10.1007/978-3-540-76298-0_52.

[10] Mohit Bansal, David Burkett, Gerard de Melo, and Dan Klein. Structured learning for taxonomy induction with belief propagation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 1041–1051, 2014. URL http://aclweb.org/anthology/P/P14/P14-1098.pdf.

[11] Boualem Benatallah, Fabio Casati, Dimitrios Georgakopoulos, Claudio Bartolini, Wasim Sadiq, and Claude Godart, editors. *Web Information Systems Engineering - WISE 2007, 8th International Conference on Web Information Systems Engineering, Nancy, France, December 3-7, 2007, Proceedings*, volume 4831 of *Lecture Notes in Computer Science*, 2007. Springer. ISBN 978-3-540-76992-7. doi: 10.1007/978-3-540-76993-4. URL https://doi.org/10.1007/978-3-540-76993-4.

[12] Chris Biemann. Ontology learning from text: A survey of methods. *LDV Forum*, 20(2): 75–93, 2005. URL http://www.jlcl.org/2005_Heft2/Chris_Biemann.pdf.

[13] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.

[14] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. Dbpedia - a crystallization point for the web of data. *Web Semant.*, 7(3):154–165, September 2009. ISSN 1570-8268. doi: 10.1016/j.websem.2009.07.002. URL http://dx.doi.org/10.1016/j.websem.2009.07.002.

[15] Georgeta Bordea, Paul Buitelaar, Stefano Faralli, and Roberto Navigli. Semeval-2015 task 17: Taxonomy extraction evaluation (texeval). In *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015*, pages 902–910, 2015. URL http://aclweb.org/anthology/S/S15/S15-2151.pdf.

[16] Georgeta Bordea, Els Lefever, and Paul Buitelaar. Semeval-2016 task 13: Taxonomy extraction evaluation (texeval-2). In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 1081–1091, 2016. URL http://aclweb.org/anthology/S/S16/S16-1168.pdf.

[17] Paul Buitelaar and Bernardo Magnini. Ontology learning from text: An overview. In *In Paul Buitelaar, P., Cimiano, P., Magnini B. (Eds.), Ontology Learning from Text: Methods, Applications and Evaluation*, pages 3–12. IOS Press, 2005.

[18] Razvan C. Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy*, 2006. URL http://aclweb.org/anthology/E/E06/E06-1002.pdf.

[19] Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, and Stelios Piperidis, editors. *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*, 2016. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/lrec2016.

[20] David Carmel, Ming-Wei Chang, Evgeniy Gabrilovich, Bo-June Paul Hsu, and Kuansan Wang. Erd '14: entity recognition and disambiguation challenge. In *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast , QLD, Australia - July 06 - 11, 2014*, page 1292, 2014. doi: 10.1145/2600428.2600734. URL http://doi.acm.org/10.1145/2600428.2600734.

[21] Shui-Lung Chuang and Lee-Feng Chien. Automatic query taxonomy generation for information retrieval applications. *Online Information Review*, 27(4):243–255, 2003. doi: 10.1108/14684520310489032. URL https://doi.org/10.1108/14684520310489032.

[22] Trevor Cohen and Dominic Widdows. Empirical distributional semantics: Methods and biomedical applications. *Journal of Biomedical Informatics*, 42(2):390–405, 2009. doi: 10.1016/j.jbi.2009.02.002. URL https://doi.org/10.1016/j.jbi.2009.02.002.

[23] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 3rd Edition.* MIT Press, 2009. ISBN 978-0-262-03384-8. URL http://mitpress.mit.edu/books/introduction-algorithms.

[24] Damien Cram and Béatrice Daille. Terminology extraction with term variant detection. In *Proceedings of ACL-2016 System Demonstrations, Berlin, Germany, August 7-12, 2016*, pages 13–18, 2016. doi: 10.18653/v1/P16-4003. URL https://doi.org/10.18653/v1/P16-4003.

[25] Gerard de Melo and Gerhard Weikum. MENTA: inducing multilingual taxonomies from wikipedia. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, pages 1099–1108, 2010. doi: 10.1145/1871437.1871577. URL http://doi.acm.org/10.1145/1871437.1871577.

[26] Gianluca Demartini, Claudiu S. Firan, Tereza Iofciu, Ralf Krestel, and Wolfgang Nejdl. Why finding entities in wikipedia is difficult, sometimes. *Inf. Retr.*, 13(5):534–567, 2010. doi: 10.1007/s10791-010-9135-7. URL https://doi.org/10.1007/s10791-010-9135-7.

[27] Ofer Egozi, Shaul Markovitch, and Evgeniy Gabrilovich. Concept-based information retrieval using explicit semantic analysis. *ACM Trans. Inf. Syst.*, 29(2):8:1–8:34, 2011. doi: 10.1145/1961209.1961211. URL http://doi.acm.org/10.1145/1961209.1961211.

[28] Edward A Feigenbaum. Knowledge engineering: The applied side. *Intelligent Systems*, pages 37–55, 1983.

[29] David A. Ferrucci, Eric W. Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John M. Prager, Nico Schlaefer, and Christopher A. Welty. Building watson: An overview of the deepqa project. *AI Magazine*, 31(3):59–79, 2010. URL http://www.aaai.org/ojs/index.php/aimagazine/article/view/2303.

[30] Tiziano Flati, Daniele Vannella, Tommaso Pasini, and Roberto Navigli. Two is bigger (and better) than one: the wikipedia bitaxonomy project. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 945–955, 2014. URL http://aclweb.org/anthology/P/P14/P14-1089.pdf.

[31] Tiziano Flati, Daniele Vannella, Tommaso Pasini, and Roberto Navigli. Multiwibi: The multilingual wikipedia bitaxonomy project. *Artif. Intell.*, 241:66–102, 2016. doi: 10.1016/j.artint.2016.08.004. URL https://doi.org/10.1016/j.artint.2016.08.004.

[32] Jian-Bo Gao, Bao-Wen Zhang, and Xiao-Hua Chen. A wordnet-based semantic similarity measurement combining edge-counting and information content theory. *Engineering Applications of Artificial Intelligence*, 39(Supplement C):80 – 88, 2015. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2014.11.009. URL http://www.sciencedirect.com/science/article/pii/S0952197614002814.

[33] Maayan Geffet and Ido Dagan. The distributional inclusion hypotheses and lexical entailment. In *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*, pages 107–114, 2005. URL http://aclweb.org/anthology/P/P05/P05-1014.pdf.

[34] Oren Glickman, Ido Dagan, and Moshe Koppel. A probabilistic classification approach for lexical textual entailment. In *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*, pages 1050–1055, 2005. URL http://www.aaai.org/Library/AAAI/2005/aaai05-166.php.

[35] Edouard Grave, Tomas Mikolov, Armand Joulin, and Piotr Bojanowski. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 427–431, 2017. URL http://aclanthology.info/papers/E17-2068/bag-of-tricks-for-efficient-text-classification.

[36] Gregory Grefenstette. INRIASAC: simple hypernym extraction methods. In *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015*, pages 911–914, 2015. URL http://aclweb.org/anthology/S/S15/S15-2152.pdf.

[37] Adam Grycner, Gerhard Weikum, Jay Pujara, James R. Foulds, and Lise Getoor. RELLY: inferring hypernym relationships between relational phrases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 971–981, 2015. URL http://aclweb.org/anthology/D/D15/D15-1113.pdf.

[38] Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. Named entity recognition in query. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*, pages 267–274, 2009. doi: 10.1145/1571941.1571989. URL http://doi.acm.org/10.1145/1571941.1571989.

[39] Amit Gupta, Francesco Piccinno, Mikhail Kozhevnikov, Marius Pasca, and Daniele Pighin. Revisiting taxonomy induction over wikipedia. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 2300–2309, 2016. URL http://aclweb.org/anthology/C/C16/C16-1217.pdf.

[40] Amit Gupta, Rémi Lebret, Hamza Harkous, and Karl Aberer. Taxonomy induction using hypernym subsequences. In *Proceedings of the 26th ACM International Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 6-10, 2017*, 2017.

[41] Amit Gupta, Rémi Lebret, Hamza Harkous, and Karl Aberer. 280 birds with one stone: Inducing multilingual taxonomies from wikipedia using character-level classification. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, February 2-7, 2016, New Orleans, Louisiana, USA.*, 2018.

[42] Iryna Gurevych and Elisabeth Wolf. Expert-built and collaboratively constructed lexical semantic resources. *Language and Linguistics Compass*, 4(11):1074–1090, 2010. doi: 10.1111/j.1749-818X.2010.00251.x. URL https://doi.org/10.1111/j.1749-818X.2010.00251.x.

[43] Sherzod Hakimov, Salih Atilay Oto, and Erdogan Dogdu. Named entity recognition and disambiguation using linked data and graph-based centrality scoring. In *Proceedings of the 4th International Workshop on Semantic Web Information Management, SWIM 2012, Scottsdale, AZ, USA, May 20, 2012*, page 4, 2012. doi: 10.1145/2237867.2237871. URL http://doi.acm.org/10.1145/2237867.2237871.

[44] Sanda M. Harabagiu, Steven J. Maiorano, and Marius Pasca. Open-domain textual question answering techniques. *Natural Language Engineering*, 9(3):231–267, 2003. doi: 10.1017/S1351324903003176. URL https://doi.org/10.1017/S1351324903003176.

[45] Hamza Harkous. *Data-Driven, Personalized Usable Privacy*. PhD thesis, IC, Lausanne, 2017.

[46] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[47] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *14th International Conference on Computational Linguistics, COLING 1992, Nantes, France, August 23-28, 1992*, pages 539–545, 1992. URL http://aclweb.org/anthology/C92-2082.

[48] Evan Heit. Properties of inductive reasoning. *Psychonomic Bulletin & Review*, 7(4): 569–592, 2000.

[49] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

[50] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. YAGO2: A spatially and temporally enhanced knowledge base from wikipedia. *Artif. Intell.*, 194: 28–61, 2013. doi: 10.1016/j.artint.2012.06.001. URL https://doi.org/10.1016/j.artint.2012.06.001.

[51] Eduard H. Hovy, Zornitsa Kozareva, and Ellen Riloff. Toward completeness in concept extraction and classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6-7 August 2009, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 948–957, 2009. URL http://www.aclweb.org/anthology/D09-1099.

[52] Eduard H. Hovy, Roberto Navigli, and Simone Paolo Ponzetto. Collaboratively built semi-structured content and artificial intelligence: The story so far. *Artif. Intell.*, 194: 2–27, 2013. doi: 10.1016/j.artint.2012.10.002. URL https://doi.org/10.1016/j.artint.2012.10.002.

[53] Xiaohua Hu, Xiaodan Zhang, Caimei Lu, E. K. Park, and Xiaohua Zhou. Exploiting wikipedia as external knowledge for document clustering. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, pages 389–396, 2009. doi: 10.1145/1557019.1557066. URL http://doi.acm.org/10.1145/1557019.1557066.

[54] Wen Hua, Zhongyuan Wang, Haixun Wang, Kai Zheng, and Xiaofang Zhou. Understand short texts by harvesting and analyzing semantic knowledge. *IEEE Trans. Knowl. Data Eng.*, 29(3):499–512, 2017. doi: 10.1109/TKDE.2016.2571687. URL https://doi.org/10.1109/TKDE.2016.2571687.

[55] Masahiro Ito, Kotaro Nakayama, Takahiro Hara, and Shojiro Nishio. Association thesaurus construction methods based on link co-occurrence analysis for wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management,*

CIKM 2008, Napa Valley, California, USA, October 26-30, 2008, pages 817–826, 2008. doi: 10.1145/1458082.1458191. URL http://doi.acm.org/10.1145/1458082.1458191.

[56] Donald B. Johnson. Finding all the elementary circuits of a directed graph. *SIAM J. Comput.*, 4(1):77–84, 1975. doi: 10.1137/0204007. URL https://doi.org/10.1137/0204007.

[57] Rianne Kaptein, Pavel Serdyukov, Arjen P. de Vries, and Jaap Kamps. Entity ranking using wikipedia as a pivot. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, pages 69–78, 2010. doi: 10.1145/1871437.1871451. URL http://doi.acm.org/10.1145/1871437.1871451.

[58] Richard M. Karp. A simple derivation of edmonds' algorithm for optimum branchings. *Networks*, 1(3):265–272, 1971. doi: 10.1002/net.3230010305. URL https://doi.org/10.1002/net.3230010305.

[59] Laura Kassner, Vivi Nastase, and Michael Strube. Acquiring a taxonomy from the german wikipedia. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco*, 2008. URL http://www.lrec-conf.org/proceedings/lrec2008/summaries/544.html.

[60] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751, 2014. URL http://aclweb.org/anthology/D/D14/D14-1181.pdf.

[61] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980.

[62] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, 7-12 July 2003, Sapporo Convention Center, Sapporo, Japan.*, pages 423–430, 2003. URL http://aclweb.org/anthology/P/P03/P03-1054.pdf.

[63] Morton Klein. A primal method for minimal cost flows with applications to the assignment and transportation problems. *Management Science*, 14(3):205–220, 1967. URL https://EconPapers.repec.org/RePEc:inm:ormnsc:v:14:y:1967:i:3:p:205-220.

[64] Morton Klein. A primal method for minimal cost flows with applications to the assignment and transportation problems. *Management Science*, 14(3):205–220, 1967.

[65] Tomáš Kliegr, Václav Zeman, and Milan Dojchinovski. Linked hypernyms dataset-generation framework and use cases. In *3rd Workshop on Linked Data in Linguistics: Multilingual Knowledge Resources and Natural Language Processing*, page 82. Citeseer, 2014.

[66] Peter Kluegl, Martin Toepfer, Philip-Daniel Beck, Georg Fette, and Frank Puppe. UIMA ruta: Rapid development of rule-based information extraction applications. *Natural Language Engineering*, 22(1):1–40, 2016. doi: 10.1017/S1351324914000114. URL https://doi.org/10.1017/S1351324914000114.

[67] Sebastian Köhler, Sandra C. Doelken, Christopher J. Mungall, Sebastian Bauer, Helen V. Firth, Isabelle Bailleul-Forestier, Graeme C. M. Black, Danielle L. Brown, Michael Brudno, Jennifer Campbell, David R. FitzPatrick, Janan T. Eppig, Andrew P. Jackson, Kathleen Freson, Marta Gîrdea, Ingo Helbig, Jane A. Hurst, Johanna Jähn, Laird G. Jackson, Anne M. Kelly, David H. Ledbetter, Sahar Mansour, Christa L. Martin, Celia Moss, Andrew Mumford, Willem Ouwehand, Soo-Mi Park, Erin Rooney Riggs, Richard H. Scott, Sanjay Sisodiya, Steven Van Vooren, Ronald J. Wapner, Andrew O. M. Wilkie, Caroline F. Wright, Anneke T. Vulto-van Silfhout, Nicole de Leeuw, Bert B. A. de Vries, Nicole L. Washington, Cynthia L. Smith, Monte Westerfield, Paul N. Schofield, Barbara J. Ruef, Georgios V. Gkoutos, Melissa Haendel, Damian Smedley, Suzanna E. Lewis, and Peter N. Robinson. The human phenotype ontology project: linking molecular biology and disease through phenotype data. *Nucleic Acids Research*, 42(Database-Issue):966–974, 2014. doi: 10.1093/nar/gkt1026. URL https://doi.org/10.1093/nar/gkt1026.

[68] Zornitsa Kozareva and Eduard H. Hovy. A semi-supervised method to learn and construct taxonomies using the web. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9-11 October 2010, MIT Stata Center, Massachusetts, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1110–1118, 2010. URL http://www.aclweb.org/anthology/D10-1108.

[69] Zornitsa Kozareva, Ellen Riloff, and Eduard H. Hovy. Semantic class learning from the web with hyponym pattern linkage graphs. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, pages 1048–1056, 2008. URL http://www.aclweb.org/anthology/P08-1119.

[70] Anton E Lawson. How do humans acquire knowledge? and what does that imply about the nature of knowledge? *Science & Education*, 9(6):577–598, 2000.

[71] Joël Legrand and Ronan Collobert. Joint rnn-based greedy parsing and word composition. *CoRR*, abs/1412.7028, 2014. URL http://arxiv.org/abs/1412.7028.

[72] Joël Legrand and Ronan Collobert. Deep neural networks for syntactic parsing of morphologically rich languages. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*, 2016. URL http://aclweb.org/anthology/P/P16/P16-2093.pdf.

[73] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 6(2):167–195, 2015. URL http://jens-lehmann.org/files/2015/swj_dbpedia.pdf.

[74] Douglas B. Lenat. CYC: A large-scale investment in knowledge infrastructure. *Commun. ACM*, 38(11):32–38, 1995. doi: 10.1145/219717.219745. URL http://doi.acm.org/10.1145/219717.219745.

[75] Xueqing Liu, Yangqiu Song, Shixia Liu, and Haixun Wang. Automatic taxonomy construction from keywords. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*, pages 1433–1441, 2012. doi: 10.1145/2339530.2339754. URL http://doi.acm.org/10.1145/2339530.2339754.

[76] Anh Tuan Luu, Jung-jae Kim, and See-Kiong Ng. Taxonomy construction using syntactic contextual evidence. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 810–819, 2014. URL http://aclweb.org/anthology/D/D14/D14-1088.pdf.

[77] Anh Tuan Luu, Siu Cheung Hui, and See-Kiong Ng. Utilizing temporal information for taxonomy construction. *TACL*, 4:551–564, 2016. URL https://transacl.org/ojs/index.php/tacl/article/view/954.

[78] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. YAGO3: A knowledge base from multilingual wikipedias. In *CIDR 2015, Seventh Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 4-7, 2015, Online Proceedings*, 2015. URL http://cidrdb.org/cidr2015/Papers/CIDR15_Paper1.pdf.

[79] Jean M Mandler and Laraine McDonough. Drinking and driving don't mix: Inductive generalization in infancy. *Cognition*, 59(3):307–335, 1996.

[80] John McCarthy. *Programs with common sense*. RLE and MIT Computation Center, 1960.

[81] John McCarthy. Epistemological problems of artificial intelligence. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence. Cambridge, MA, USA, August 22-25, 1977*, pages 1038–1044, 1977. URL http://ijcai.org/Proceedings/77-2/Papers/094.pdf.

[82] Olena Medelyan, Ian H. Witten, Anna Divoli, and Jeen Broekstra. Automatic construction of lexicons, taxonomies, ontologies, and other knowledge structures. *Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery*, 3(4):257–279, 2013. doi: 10.1002/widm.1097. URL https://doi.org/10.1002/widm.1097.

[83] Christian M. Meyer and Iryna Gurevych. What psycholinguists know about chemistry: Aligning wiktionary and wordnet for increased domain coverage. In *Fifth International Joint Conference on Natural Language Processing, IJCNLP 2011, Chiang Mai, Thailand, November 8-13, 2011*, pages 883–892, 2011. URL http://aclweb.org/anthology/I/I11/I11-1099.pdf.

[84] Peter Mika, Massimiliano Ciaramita, Hugo Zaragoza, and Jordi Atserias. Learning to tag and tagging to learn: A case study on wikipedia. *IEEE Intelligent Systems*, 23(5):26–33, 2008. doi: 10.1109/MIS.2008.85. URL https://doi.org/10.1109/MIS.2008.85.

[85] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119, 2013. URL http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.

[86] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, 1995. doi: 10.1145/219717.219748. URL http://doi.acm.org/10.1145/219717.219748.

[87] George A Miller, Claudia Leacock, Randee Tengi, and Ross T Bunker. A semantic concordance. In *Proceedings of the workshop on Human Language Technology*, pages 303–308. Association for Computational Linguistics, 1993.

[88] Dan I. Moldovan, Sanda M. Harabagiu, Marius Pasca, Rada Mihalcea, Richard Goodrum, Roxana Girju, and Vasile Rus. LASSO: A tool for surfing the answer net. In *Proceedings of The Eighth Text REtrieval Conference, TREC 1999, Gaithersburg, Maryland, USA, November 17-19, 1999*, 1999. URL http://trec.nist.gov/pubs/trec8/papers/smu.pdf.

[89] Ndapandula Nakashole, Gerhard Weikum, and Fabian M. Suchanek. PATTY: A taxonomy of relational patterns with semantic types. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 1135–1145, 2012. URL http://www.aclweb.org/anthology/D12-1104.

[90] Vivi Nastase and Michael Strube. Decoding wikipedia categories for knowledge acquisition. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pages 1219–1224, 2008. URL http://www.aaai.org/Library/AAAI/2008/aaai08-193.php.

[91] Vivi Nastase, Michael Strube, Benjamin Boerschinger, Cäcilia Zirn, and Anas Elghafari. Wikinet: A very large scale multi-lingual concept network. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta*, 2010. URL http://www.lrec-conf.org/proceedings/lrec2010/summaries/615.html.

[92] Roberto Navigli and Simone Paolo Ponzetto. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artif. Intell.*, 193:217–250, 2012. doi: 10.1016/j.artint.2012.07.001. URL https://doi.org/10.1016/j.artint.2012.07.001.

[93] Roberto Navigli and Paola Velardi. Learning word-class lattices for definition and hypernym extraction. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 1318–1327, 2010. URL http://www.aclweb.org/anthology/P10-1134.

[94] Roberto Navigli, Paola Velardi, and Stefano Faralli. A graph-based algorithm for inducing lexical taxonomies from scratch. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 1872–1877, 2011. doi: 10.5591/978-1-57735-516-8/IJCAI11-313. URL https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-313.

[95] Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *CoRR*, abs/1705.08039, 2017. URL http://arxiv.org/abs/1705.08039.

[96] Robert C. Nickerson, Upkar Varshney, and Jan Muntermann. A method for taxonomy development and its application in information systems. *EJIS*, 22(3):336–359, 2013. doi: 10.1057/ejis.2012.26. URL https://doi.org/10.1057/ejis.2012.26.

[97] Joel Nothman, Tara Murphy, and James R. Curran. Analysing wikipedia and gold-standard corpora for NER training. In *EACL 2009, 12th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, Athens, Greece, March 30 - April 3, 2009*, pages 612–620, 2009. URL http://www.aclweb.org/anthology/E09-1070.

[98] Michael P. Oakes. Using hearst's rules for the automatic acquisition of hyponyms for mining a pharmaceutical corpus. In *International Workshop Text Mining Research, Practice and Opportunities, Proceedings, Borovets, Bulgaria, 24 September 2005, held in conjunction with RANLP 2005*, pages 63–67, 2005.

[99] James B. Orlin. A polynomial time primal network simplex algorithm for minimum cost flows. *Math. Program.*, 77:109–129, 1997. doi: 10.1007/BF02614365. URL https://doi.org/10.1007/BF02614365.

[100] Sebastian Padó and Mirella Lapata. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199, 2007. doi: 10.1162/coli.2007.33.2.161. URL https://doi.org/10.1162/coli.2007.33.2.161.

[101] Alexander Panchenko, Olga Morozova, and Hubert Naets. A semantic similarity measure based on lexico-syntactic patterns. In *11th Conference on Natural Language Processing, KONVENS 2012, Empirical Methods in Natural Language Processing, Vienna, Austria, September 19-21, 2012*, pages 174–178, 2012. URL http://www.oegai.at/konvens2012/proceedings/23_panchenko12p/.

[102] Alexander Panchenko, Stefano Faralli, Eugen Ruppert, Steffen Remus, Hubert Naets, Cédrick Fairon, Simone Paolo Ponzetto, and Chris Biemann. TAXI at semeval-2016 task 13: a taxonomy induction method based on lexico-syntactic patterns, substrings

and focused crawling. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 1320–1327, 2016. URL http://aclweb.org/anthology/S/S16/S16-1206.pdf.

[103] Marco Pennacchiotti and Patrick Pantel. Ontologizing semantic relations. In *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006*, 2006. URL http://aclweb.org/anthology/P06-1100.

[104] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543, 2014. URL http://aclweb.org/anthology/D/D14/D14-1162.pdf.

[105] Simone Paolo Ponzetto. Creating a knowledge base from a collaboratively generated encyclopedia. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, April 22-27, 2007, Rochester, New York, USA*, pages 9–12, 2007. URL http://www.aclweb.org/anthology/N07-3003.

[106] Simone Paolo Ponzetto and Roberto Navigli. Knowledge-rich word sense disambiguation rivaling supervised systems. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 1522–1531, 2010. URL http://www.aclweb.org/anthology/P10-1154.

[107] Simone Paolo Ponzetto and Michael Strube. Knowledge derived from wikipedia for computing semantic relatedness. *J. Artif. Intell. Res.*, 30:181–212, 2007. doi: 10.1613/jair.2308. URL https://doi.org/10.1613/jair.2308.

[108] Simone Paolo Ponzetto and Michael Strube. Deriving a large-scale taxonomy from wikipedia. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pages 1440–1445, 2007. URL http://www.aaai.org/Library/AAAI/2007/aaai07-228.php.

[109] Simone Paolo Ponzetto and Michael Strube. Wikitaxonomy: A large scale knowledge resource. In *ECAI 2008 - 18th European Conference on Artificial Intelligence, Patras, Greece, July 21-25, 2008, Proceedings*, pages 751–752, 2008. doi: 10.3233/978-1-58603-891-5-751. URL https://doi.org/10.3233/978-1-58603-891-5-751.

[110] Simone Paolo Ponzetto and Michael Strube. Taxonomy induction based on a collaboratively built knowledge repository. *Artif. Intell.*, 175(9-10):1737–1756, 2011. doi: 10.1016/j.artint.2011.01.003. URL https://doi.org/10.1016/j.artint.2011.01.003.

[111] Hoifung Poon and Pedro M. Domingos. Unsupervised ontology induction from text. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 296–305, 2010. URL http://www.aclweb.org/anthology/P10-1031.

[112] Tye Rattenbury, Nathaniel Good, and Mor Naaman. Towards automatic extraction of event and place semantics from flickr tags. In *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007*, pages 103–110, 2007. doi: 10.1145/1277741.1277762. URL http://doi.acm.org/10.1145/1277741.1277762.

[113] Matthew Richardson and Pedro M. Domingos. Building large knowledge bases by mass collaboration. In *Proceedings of the 2nd International Conference on Knowledge Capture (K-CAP 2003), October 23-25, 2003, Sanibel Island, FL, USA*, pages 129–137, 2003. doi: 10.1145/945645.945665. URL http://doi.acm.org/10.1145/945645.945665.

[114] Eleanor Rosch. Principles of categorization. *Concepts: core readings*, 189, 1999.

[115] Patrick Schone, Gary M. Ciany, R. Cutts, Paul McNamee, James Mayfield, and Thomas Smith. Qactis-based question answering at TREC 2005. In *Proceedings of the Fourteenth Text REtrieval Conference, TREC 2005, Gaithersburg, Maryland, USA, November 15-18, 2005*, 2005. URL http://trec.nist.gov/pubs/trec14/papers/dept-o-defense.qa.pdf.

[116] Lenhart K. Schubert. Turing's dream and the knowledge challenge. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, pages 1534–1538, 2006. URL http://www.aaai.org/Library/AAAI/2006/aaai06-244.php.

[117] F. Sclano and P. Velardi. Termextractor: a web application to learn the shared terminology of emergent web communities. In *Enterprise Interoperability II - New Challenges and Industrial Approaches, Proceedings of the 3th International Conference on Interoperability for Enterprise Software and Applications, IESA 2007, March 27-30, 2007, Funchal, Madeira Island, Portugal*, pages 287–290, 2007. doi: 10.1007/978-1-84628-858-6_32. URL https://doi.org/10.1007/978-1-84628-858-6_32.

[118] Julian Seitner, Christian Bizer, Kai Eckert, Stefano Faralli, Robert Meusel, Heiko Paulheim, and Simone Paolo Ponzetto. A large database of hypernymy relations extracted from the web. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016.*, 2016. URL http://www.lrec-conf.org/proceedings/lrec2016/summaries/204.html.

[119] Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada]*, pages 1297–1304, 2004. URL http://papers.nips.cc/paper/2659-learning-syntactic-patterns-for-automatic-hypernym-discovery.

[120] Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. Semantic taxonomy induction from heterogenous evidence. In *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics,*

Proceedings of the Conference, Sydney, Australia, 17-21 July 2006, 2006. URL http://aclweb.org/anthology/P06-1101.

[121] Rion Langley Snow. *Semantic Taxonomy Induction*. PhD thesis, Stanford University, 2009.

[122] Yangqiu Song, Shixia Liu, Xueqing Liu, and Haixun Wang. Automatic taxonomy construction from keywords via scalable bayesian rose trees. *IEEE Trans. Knowl. Data Eng.*, 27(7):1861–1874, 2015. doi: 10.1109/TKDE.2015.2397432. URL https://doi.org/10.1109/TKDE.2015.2397432.

[123] Michael Strube and Simone Paolo Ponzetto. Wikirelate! computing semantic relatedness using wikipedia. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, pages 1419–1424, 2006. URL http://www.aaai.org/Library/AAAI/2006/aaai06-223.php.

[124] Rudi Studer, V. Richard Benjamins, and Dieter Fensel. Knowledge engineering: Principles and methods. *Data Knowl. Eng.*, 25(1-2):161–197, 1998. doi: 10.1016/S0169-023X(97)00056-6. URL https://doi.org/10.1016/S0169-023X(97)00056-6.

[125] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, pages 697–706, 2007. doi: 10.1145/1242572.1242667. URL http://doi.acm.org/10.1145/1242572.1242667.

[126] Bongwon Suh, Gregorio Convertino, Ed H. Chi, and Peter Pirolli. The singularity is not near: Slowing growth of wikipedia. In *Proceedings of the 5th International Symposium on Wikis and Open Collaboration*, WikiSym '09, pages 8:1–8:10, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-730-1. doi: 10.1145/1641309.1641322. URL http://doi.acm.org/10.1145/1641309.1641322.

[127] Bongwon Suh, Gregorio Convertino, Ed H. Chi, and Peter Pirolli. The singularity is not near: slowing growth of wikipedia. In *Proceedings of the 2009 International Symposium on Wikis, 2009, Orlando, Florida, USA, October 25-27, 2009*, 2009. doi: 10.1145/1641309.1641322. URL http://doi.acm.org/10.1145/1641309.1641322.

[128] Idan Szpektor, Ido Dagan, Roy Bar-Haim, and Jacob Goldberger. Contextual preferences. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, pages 683–691, 2008. URL http://www.aclweb.org/anthology/P08-1078.

[129] Paola Velardi, Stefano Faralli, and Roberto Navigli. Ontolearn reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics*, 39(3):665–707, 2013. doi: 10.1162/COLI_a_00146. URL https://doi.org/10.1162/COLI_a_00146.

[130] Luis von Ahn. Games with a purpose. *IEEE Computer*, 39(6):92–94, 2006. doi: 10.1109/MC.2006.196. URL https://doi.org/10.1109/MC.2006.196.

[131] Luis von Ahn and Laura Dabbish. Designing games with a purpose. *Commun. ACM*, 51(8):58–67, 2008. doi: 10.1145/1378704.1378719. URL http://doi.acm.org/10.1145/1378704.1378719.

[132] Chengyu Wang, Xiaofeng He, and Aoying Zhou. A short survey on taxonomy learning from text corpora: Issues, resources and recent advances. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1201–1214, 2017. URL http://aclanthology.info/papers/D17-1124/d17-1124.

[133] Dominic Widdows. Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2003, Edmonton, Canada, May 27 - June 1, 2003*, 2003. URL http://aclweb.org/anthology/N/N03/N03-1036.pdf.

[134] Wikipedia. Wikipedia community. https://en.wikipedia.org/wiki/Wikipedia_community, 2017. [Online; accessed 11-October-2017].

[135] Wikipedia. Wikipedia:manual of style. https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style, 2017. [Online; accessed 11-October-2017].

[136] Wikipedia. Editing guidelines for wikipedia categories. ttps://en.wikipedia.org/wiki/Wikipedia:Categorization, 2017. [Online; accessed 27-July-2017].

[137] Wikipedia. Watson (computer). https://en.wikipedia.org/wiki/Watson_(computer), 2017. [Online; accessed 7-October-2017].

[138] Wikipedia. Wikipedia. https://en.wikipedia.org/wiki/Wikipedia, 2017. [Online; accessed 11-October-2017].

[139] Wikipedia. List of wikipedias — wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=List_of_Wikipedias&oldid773693902, 2017. [Online; accessed 9-April-2017].

[140] YAGO. Yago: A high-quality knowledge base. http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/, 2017. [Online; accessed 26-July-2017].

[141] Hui Yang and Jamie Callan. A metric-based framework for automatic taxonomy induction. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*, pages 271–279, 2009. URL http://www.aclweb.org/anthology/P09-1031.

[142] Shuo Yang, Lei Zou, Zhongyuan Wang, Jun Yan, and Ji-Rong Wen. Efficiently answering technical questions - A knowledge graph approach. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 3111–3118, 2017. URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14576.

[143] Torsten Zesch, Christof Müller, and Iryna Gurevych. Using wiktionary for computing semantic relatedness. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pages 861–866, 2008. URL http://www.aaai.org/Library/AAAI/2008/aaai08-137.php.

[144] Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657, 2015. URL http://papers.nips.cc/paper/5782-character-level-convolutional-networks-for-text-classification.

[145] Yuchen Zhang, Amr Ahmed, Vanja Josifovski, and Alexander J. Smola. Taxonomy discovery for personalized recommendation. In *Seventh ACM International Conference on Web Search and Data Mining, WSDM 2014, New York, NY, USA, February 24-28, 2014*, pages 243–252, 2014. doi: 10.1145/2556195.2556236. URL http://doi.acm.org/10.1145/2556195.2556236.

[146] Xingwei Zhu, Zhaoyan Ming, Xiaoyan Zhu, and Tat-Seng Chua. Topic hierarchy construction for the organization of multi-source user generated contents. In *The 36th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '13, Dublin, Ireland - July 28 - August 01, 2013*, pages 233–242, 2013. doi: 10.1145/2484028.2484032. URL http://doi.acm.org/10.1145/2484028.2484032.

Amit Gupta

Ph.D.
Natural Language Processing, Machine Learning

☎ *+41 78 67 19 118*
✉ *amitgupta151@gmail.com*
🖱 *http://amitgupta.co*
Nationality: Indian

Professional Experience

Sept 2013 - Dec 2017 **Doctoral Assistant**, *École Polytechnique Fédérale de Lausanne*, Switzerland.
Worked on the problem of automated taxonomy induction and its applications under the supervision of Prof. Karl Aberer. The thesis is nominated for the EPFL's **best dissertation award**, which is awarded to top 5% theses in EPFL per year. Results of the award awaited in 2018.

Jul 2015 - Dec 2015 **Intern**, *Google Inc.*, Zurich.
Worked on taxonomy induction from Wikipedia and learning generalization templates for Wikipedia entities under the supervision of Dr. Daniele Pighin. Internship work is published at COLING'16.

Apr 2013 - Aug 2013 **Senior Data Scientist**, *Zlemma.com*, Pune.
Conceptualized and implemented a parser for extracting structured data from unstructured resumes using advanced machine learning techniques. The system outperformed all the competitors in the market.

Sep 2012 - Mar 2013 **Founder**, *Squareft.in*, New Delhi.
Designed and developed a next generation map-based portal for the real estate market. Performed full stack web development using technologies such as RubyOnRails, HTML, js, coffee, Amazon EC2.

Jul 2010 - Aug 2012 **Strategist**, *Tower Research Capital Inc.*, New York and Gurgaon.
Worked on mathematical modeling of signals and high-frequency trading strategies for European and Canadian markets. Involved in all aspects of trading including development, testing, deployment, and monitoring of an end-to-end trading system.

Jan 2009 - Apr 2009 **Research Assistant**, *Rice University*, Houston.
Worked on the problem of automated taxonomy induction and its applications under the supervision of Prof. Karl Aberer.

May 2008 - July 2008 **Summer Intern**, *INRIA*, Sophia Antipolis.
Worked on poisson reconstruction of 2D and 3D surfaces using polygon soups.

May 2007 - July 2007 **Summer Intern**, *Vanderbilt University*, Nashville.
Worked on a collaborative project between Stanford University and Vanderbilt University funded by the Department of Education, USA to promote learning and reasoning skills in middle school students. Analyzed behavior patterns of students using hidden markov models.

Education

Sep 2013 - Dec 2017 **École polytechnique fédérale de Lausanne**, *Switzerland*, *5.63 / 6.*
Ph.D. in Computer Science.

Jul 2005 - Jun 2010 **IIT Bombay**, *Powai*, *9.27 / 10.*
B.Tech. + M.Tech. in Computer Science.

Jan 2009 - Apr 2009 **Rice University**, *Houston*, *4.15 / 4*.
Exchange student for a semester in the Department of Computer Science.

Achievements and Awards

EPFL 2017 Ph.D. thesis is nominated for EPFL's **best dissertation award** (awarded to top 5% thesis in EPFL).

EPFL 2013 Awarded the EPFL EDIC fellowship for pursual of doctoral studies.

IIT Bombay 2010 Awarded **Dr. George B. Fernandez fellowship** for academic excellence.

ACM ICPC 2008 Selected to represent IIT-Bombay in South Asia Regionals, ACM-Inter Collegiate Programming Competition.

IIT-JEE 2005 Ranked **6th** (All India Rank) among 200,000 students in IIT–Joint Entrance Screening Examination.

AIEEE 2005 Ranked **37th** (All India Rank) among 200,000 students in CBSE All India Engineering Entrance Examination. Awarded the CBSE AIEEE Scholarship.

INMO 2004 Ranked **17th** (All India Rank) in Indian National Mathematics Olympiad, hence, selected for the International Mathematics Olympiad Training Camp.

NSEP 2004 Ranked among top **1%** (All India) students in National Standard Examination in Physics.

NTSE 2003 Awarded National Talent Search scholarship.

Publications

AAAI 2018 **Gupta, A.**, Lebret, R., Harkous, H., & Aberer, K. (2017). 280 Birds with One Stone: Inducing Multilingual Taxonomies from Wikipedia using Character-level Classification. In proceedings of the 32nd AAAI Conference on Artificial Intelligence *(accepted to appear in AAAI 2018)*.

CIKM 2017 **Gupta, A.**, Lebret, R., Harkous, H., & Aberer, K. (2017). Taxonomy Induction using Hypernym Subsequences. In Proceedings of the 26th Conference on Information and Knowledge Management (No. EPFL-CONF-230205)

LDOW 2017 Smeros, P., **Gupta, A.**, Catasta, M., & Aberer, K. (2017). deepschema. org: An Ontology for Typing Entities in the Web of Data. In 10th Workshop on Linked Data on the Web (LDOW 2017) (No. EPFL-CONF-227993).

COLING 2016 **Gupta, A.**, Piccinno, F., Kozhevnikov, M., Pasca, M., & Pighin, D. (2016). Revisiting Taxonomy Induction over Wikipedia. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, December 11-17 2016 (No. EPFL-CONF-227401, pp. 2300-2309).

Greedy Algorithms 2008 Bellur, U., Vadodaria, H., & **Gupta, A.** (2008). Semantic Matchmaking Algorithms. In Greedy Algorithms. InTech.

ITS 2008 Jeong, H., **Gupta, A.**, Roscoe, R., Wagster, J., Biswas, G., & Schwartz, D. (2008). Using Hidden Markov models to characterize student behaviors in learning-by-teaching environments. In Intelligent Tutoring Systems (pp. 614-625). Springer Berlin/Heidelberg.

Relevant Courses

Mathematics	Introduction to Probability, Linear Algebra, Random Processes and Statistical Inference, Combinatorics, Game Theory (Rice University)
Theoretical Computer Science	Algorithms and Data Structures, Design and Analysis of Algorithms, Graph theory, Formal Methods in Computer Science, Theory of Computation, Linear Optimization, Distributed Algorithms (EPFL)
Machine Learning	Artificial Intelligence, Statistical Foundations of Machine Learning, Data Mining, Information Retrieval and Mining for web, Adaptive Systems (Rice University), Foundations of Imaging Science (EPFL)
Databases	Database and Information systems, Big Data (EPFL)
Miscellaneous	Operating Systems, Software Systems, Computer Networks, Network Security I & II, Principles of Programming Languages, Language Processors, Advanced Compilation in Parallel processors (Rice University)

Other Projects

May 2009 - Jun 2010 **Master's Thesis**, *Topic: Financial Forecasting*, IIT Bombay, Powai.

Defined new mathematical properties for time series, which helped in highly accurate clustering. Used the clustering information to construct a new forecasting system, which achieved an improvement of **12%** over existing methods.

Jan 2009 - Apr 2009 **Course Project**, *Topic: Netflix Recommender System Challenge*, Rice University, Houston.

Designed and developed a system for recommending movies to users based on Netflix user preferences data. Achieved more than **9%** improvement over the existing Netflix solution.

Jan 2007 - Apr 2007 **Course Project**, *Topic: Email Client*, IIT Bombay, Powai.

Worked in a team of two to build a fully-featured email client solution (similar to Thunderbird) using Java swing Library.

Teaching Assistantship

EPFL	Programmation I, Programmation II, Distributed Information Systems, Software Engineering (project coach).
IIT Bombay	Linear Optimization.

Technical Skills

Programming Languages	Python, C/C++, JAVA, Lua, Ruby.
Libraries & Software Packages	Torch, MySQL, Postgresql, MongoDB, ElasticSearch, Hadoop, MATLAB, RubyonRails.
Scripting Languages	Perl, Awk, Bash.
Other	LaTeX, SQL, HTML.

Languages

English (Fluent), Hindi (Native), French (Basic)