Université de Montréal

# Automatic Taxonomy Evaluation

par

**Tianjian Lucas Gao**

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en Intelligence Artificielle

6 décembre, 2022

# Université de Montréal

Faculté des arts et des sciences

Ce mémoire intitulé

**Automatic Taxonomy Evaluation**

présenté par

## Tianjian Lucas Gao

a été évalué par un jury composé des personnes suivantes :

*Esma Aïmeur*

(président-rapporteur)

*Philippe Langlais*

(directeur de recherche)

*Guy Lapalme*

(membre du jury)

# Résumé

Les taxonomies sont une représentation essentielle des connaissances, jouant un rôle central dans de nombreuses applications *riches en connaissances*. Malgré cela, leur construction est laborieuse que ce soit manuellement ou automatiquement, et l'évaluation quantitative de taxonomies est un sujet négligé. Lorsque les chercheurs se concentrent sur la construction d'une taxonomie à partir de grands corpus *non structurés*, l'évaluation est faite souvent manuellement, ce qui implique des biais et se traduit souvent par une reproductibilité limitée. Les entreprises qui souhaitent améliorer leur taxonomie manquent souvent d'étalon ou de référence, une sorte de taxonomie bien optimisée pouvant service de référence. Par conséquent, des connaissances et des efforts spécialisés sont nécessaires pour évaluer une taxonomie.

Dans ce travail, nous soutenons que l'évaluation d'une taxonomie effectuée automatiquement et de manière reproductible est aussi importante que la génération automatique de telles taxonomies. Nous proposons deux nouvelles méthodes d'évaluation qui produisent des scores moins biaisés: un modèle de classification de la taxonomie extraite d'un corpus étiqueté, et un modèle de langue non supervisé qui sert de source de connaissances pour évaluer les relations hyperonymiques. Nous constatons que nos substituts d'évaluation corrèlent avec les jugements humains et que les modèles de langue pourraient imiter les experts humains dans les tâches riches en connaissances.

**Mots-clés: Taxonomie, Ontologie, Apprentissage de taxonomie, Évaluation d'ontologie, Extraction de connaissances, Représentation des connaissances, Extraction de l'information, Modélisation du langage, Découverte d'hyperonymes**

# Abstract

Taxonomies are an essential knowledge representation and play an important role in classification and numerous knowledge-rich applications, yet quantitative taxonomy evaluation remains to be overlooked and left much to be desired. While studies focus on automatic taxonomy construction (ATC) for extracting meaningful structures and semantics from large corpora, their evaluation is usually manual and subject to bias and low reproducibility. Companies wishing to improve their domain-focused taxonomies also suffer from lacking ground-truths. In fact, manual taxonomy evaluation requires substantial labour and expert knowledge.

As a result, we argue in this thesis that automatic taxonomy evaluation (ATE) is just as important as taxonomy construction. We propose two novel taxonomy evaluation methods for automatic taxonomy scoring, leveraging supervised classification for labelled corpora and unsupervised language modelling as a knowledge source for unlabelled data. We show that our evaluation *proxies* can exert similar effects and correlate well with human judgments and that language models can imitate human experts on knowledge-rich tasks.

**Keywords: Taxonomy, Ontology, Taxonomy Learning, Ontology Evaluation, Knowledge Representation, Knowledge Extraction, Information Retrieval, Information Extraction, Hypernym Discovery, Language Modelling**

# Contents

**Résumé** . . . . . . . . . . 5

**Abstract** . . . . . . . . . . 7

**List of tables** . . . . . . . . . . 13

**List of figures** . . . . . . . . . . 15

**List of abbreviations and acronyms** . . . . . . . . . . 17

**Acknowledgements** . . . . . . . . . . 19

**Introduction** . . . . . . . . . . 21

Motivation . . . . . . . . . . 22

Approach . . . . . . . . . . 22

Challenge . . . . . . . . . . 23

Presentation . . . . . . . . . . 23

**Chapter 1. Defining Taxonomy** . . . . . . . . . . 25

1.1. Taxonomy Learning . . . . . . . . . . 25

1.2. Taxonomy vs. Ontology vs. Knowledge Graph . . . . . . . . . . 26

Taxonomy . . . . . . . . . . 27

Topic taxonomy: a formal description . . . . . . . . . . 27

Ontology . . . . . . . . . . 28

Knowledge Graph . . . . . . . . . . 28

1.3. Taxonomy Applications . . . . . . . . . . 28

Bio-medicine and Clinical Care . . . . . . . . . . 29

Semantic Web . . . . . . . . . . 29

Product and Data Management . . . . . . . . . . 29

1.4. Taxonomy and Large-scale Classification Systems . . . . . . . . . . 30

1.5. Essential Ontologies in Research ..... 31
WordNet ..... 31
Yago ..... 31
Probase ..... 32

**Chapter 2. Automatic Taxonomy Construction** ..... 33

2.1. Pattern-based approach ..... 34

2.2. Distributional Approach ..... 35
2.2.1. Feature Representation ..... 35
2.2.2. Term Extraction ..... 37
2.2.3. Clustering ..... 39
2.2.4. Supervised Relation Classifier ..... 41

2.3. Tested Systems ..... 41
2.3.1. Description ..... 42
2.3.2. Observation ..... 43
2.3.3. Illustration ..... 44

2.4. Final Remarks ..... 46

**Chapter 3. Towards Automatic Taxonomy Evaluation** ..... 47

3.1. Taxonomy Errors ..... 47

3.2. Concept-pair and Coverage Analysis ..... 49

3.3. Comparing Taxonomies ..... 50

3.4. Intrinsic Evaluation ..... 52
Davies-Bouldin Index ..... 52
Silhouette Score ..... 53
Calinski-Harabasz Index ..... 53

3.5. Extrinsic Evaluation ..... 53
3.5.1. Gold Standard Method ..... 54
3.5.2. Data-driven method ..... 56
3.5.3. Application-based method ..... 57
3.5.4. Manual Evaluation ..... 58
3.5.5. Rank-based Metrics ..... 59

3.6. Towards Automatic Evaluation: a Dialogue ..... 59
"Which system works the best?" ..... 60
"Which taxonomy is the best?" ..... 60
Automatic Topic Coherence Evaluation ..... 61
Machine Learning as External Evaluators ..... 61

**Chapter 4. Supervised Taxonomy Evaluation with Text Classification..... 63**

4.1. Evaluation Targets ..... 64
4.1.1. Datasets ..... 64
4.1.2. Generated Taxonomies ..... 65

4.2. Feature Extraction ..... 69
4.2.1. Improving taxonomy-corpus matching ..... 70

4.3. Evaluation Procedure ..... 71

4.4. Perturbation Test ..... 71
Random Shuffle ..... 72
Random Deletion ..... 72
4.4.1. Results ..... 73

4.5. Evaluation Results and Ranking ..... 73

4.6. Conclusion ..... 76

**Chapter 5. Unsupervised Taxonomy Evaluation via MLM Prompting..... 77**

5.1. Masked Language Modelling ..... 77

5.2. Prompting ..... 79

5.3. Evaluation Procedure ..... 80
Example ..... 81

5.4. Evaluation Targets ..... 81

5.5. Improving Hypernymy Predictions ..... 84
5.5.1. Diversified Prompting ..... 84
5.5.2. Fine-tuning ..... 85

5.6. Results ..... 88

5.7. Conclusion ..... 91

**Chapter 6. Conclusion** . . . . . . . . . . 93

**References** . . . . . . . . . . 95

**Appendix A. MLM Proxy Evaluation Targets** . . . . . . . . . . 103

A.1. Evaluation Terms Added to the BERT Vocabulary . . . . . . . . . . 107

# List of tables

1.1 Overview of taxonomy, ontology and knowledge graph ........ 28

2.1 Count vectorization: an example ........ 36

4.1 Taxonomy topics and document labels ........ 64

4.2 Sample AG News data input ........ 65

4.3 Structures of taxonomies for ranking ........ 66

4.4 Neighbours (synonyms) selected by ATC programs as alternative topic labels .... 66

4.5 Categorical feature extraction ........ 69

4.6 Feature transformation with TF-IDF scoring ........ 70

4.7 Stemming vs. lemmatization ........ 70

4.8 Taxonomy coverage test ........ 75

4.9 Pairwise comparison of proxy classification scores ........ 75

5.1 Recalling Hypernymy via prompting ........ 80

5.2 Illustration of relation accuracy evaluation ........ 82

5.3 Main topics of MLM evaluation targets ........ 82

5.4 Evaluation parent-child samples ........ 83

5.5 Evaluation queries for "seafood-shrimp" of HiExpan1 ........ 84

5.6 Evaluation model configurations ........ 86

5.7 Masking Logic ........ 87

5.8 Fine-tuned vs. pre-trained models, "my favorite [MASK] is sirloin ." ........ 87

5.9 extended vs. base vocabulary, "[MASK] such as mozzarella sticks" ........ 88

5.10 Automatic relation accuracy scoring ........ 89

5.11 Taxonomy rankings of relation accuracy evaluation ........ 90

5.12 Sample relation accuracy evaluation results (Model2a) ........ 90

A.1 CoRel1 sampled clusters . . . . . . 103
A.2 CoRel2 sampled clusters . . . . . . 104
A.3 CoRel3 sampled clusters . . . . . . 104
A.4 CoRel4 sampled clusters . . . . . . 105
A.5 HiExpan1 sampled topics . . . . . . 105

# List of figures

1.1 An auto-constructed (and possibly erroneous) taxonomy ........ 25
1.2 Knowledge representation visualizations ........ 26
1.3 Excerpt of Google Content Categories ........ 30
1.4 A subset of WordNet topic "event" ........ 31

2.1 A typical taxonomy induction pipeline (Liu et al. 2012) ........ 33
2.2 A taxonomy of ATC methods ........ 34
2.3 AutoPhrase output with term integrity scores ........ 38
2.4 Compactness vs. connectivity based clustering ........ 40
2.5 Different CoRel runs generate different results ........ 43
2.6 Format of TaxoCom output (DBLP dataset) ........ 44
2.7 Sample CoRel subtopic clusters for topic "steak" (Yelp Dataset) ........ 44
2.8 Sample HiExpan output (arXiv dataset) ........ 45
2.9 Sample TaxoCom output (arXiv dataset) ........ 45

3.1 Noisy labels in an auto-generated taxonomy ........ 48
3.2 Erroneous parent label ........ 48
3.3 Cluster coherence problem ........ 49
3.4 Taxonomy structure problem ........ 49
3.5 Matrix visualization of taxonomy differences (Sancho-Chavarria et al. 2020) ..... 51
3.6 Subset of WordNet Synset ........ 56

4.1 "TaxoGen 1" with 3 levels and 4 children per topic (TG1) ........ 67
4.2 "TaxoGen 2" with 3 levels (TG2) ........ 67
4.3 "HiExpan 1" with 2 levels and 9 topics (HE1) ........ 67
4.4 "HiExpan 2" with 1 level and 19 topics (HE2) ........ 67
4.5 "TaxoCom 1" with 2 levels and 5 topics (TC1) ........ 68

4.6 "TaxoCom off-topic" generated from arXiv dataset (TC*) ........ 68

4.7 "HiExpan off-topic" generated from an aviation dataset (HE*) ........ 68

4.8 Reassigning topics to new positions to simulate a noisy taxonomy ........ 72

4.9 Randomly deleting taxonomy edges for reducing taxonomy coverage ........ 72

4.10 Perturbation test results ........ 74

5.1 Training a masked language model, an example ........ 78

5.2 Excerpt from HiExpan1 for topic "seafood" ........ 79

A.1 TaxoGen1 sampled topics ........ 106

A.2 TaxoGen2 topics ........ 106

# List of abbreviations and acronyms

| | |
|---|---|
| **ATC** | Automatic Taxonomy Construction |
| **ATE** | Automatic Taxonomy Evaluation |
| **BLEU** | BiLingual Evaluation Understudy |
| **BoW** | Bag of Words (for text sequence representation) |
| **CBOW** | Continuous Bag of Words |
| **DL** | Deep Learning |
| **GLUE** | General Language Understanding Evaluation benchmark |
| **IR** | Information Retrieval |
| **IS** | Information Systems |
| **KB** | Knowledge Base |
| **KG** | Knowledge Graph |
| **LM** | Language Model |

| | |
|---|---|
| **LDA** | Latent Dirichlet Allocation |
| **LSTM** | Long Short-Term Memory |
| **MAP** | Mean Average Precision |
| **ML** | Machine Learning |
| **MLM** | Masked Language Model(ling) |
| **MRR** | Mean Reciprocal Rank |
| **NER** | Named Entity Recognition |
| **NLP** | Natural Language Processing |
| **NP** | Noun Phrase |
| **PCA** | Principle Component Analysis |
| **POS** | Part-of-speech |
| **Q&A** | Question Answering |
| **SOTA** | State-of-the-art, most advanced |
| **TF-IDF** | Term Frequency - Inverse Document Frequency |

# Acknowledgements

*Philippe, je vous remercier pour votre soutien généreux, vos idées et vos feedbacks précieux pendant une période énormément difficile. Je connaissais très peu les modèles de langage masqué il y a juste un an! Merci* bien gros *à vous prof. Lapalme pour votre révision et vos commentaires. Merci à mes collègues du RALI et mes superviseurs de stages à IATA, surtout Olena, Hyuntae, Sorina et James, des gens amicaux et très professionnels. Cette thèse n'aurait pas été possible sans le soutien financier de MITACS/IATA. Merci à DIRO et FAS pour les bourses. Je suis reconnaissant à Jeffrey, Gary, Cindy, Céline, mes parents et mes amis. Kitty, tu me manques...*

I would like to especially thank my dear research director, Prof. Philippe Langlais for guiding me through the ordeal, for that I am grateful to have stood upon his shoulders and walked his footsteps. Thank you prof. Lapalme for the insightful comments, as well as to all of my brilliant colleagues at RALI and DIRO, and to my esteemed supervisors at IATA particularly Olena, Hyuntae, Sorina and James. I am grateful for the generous scholarships offered by DIRO and FAS, and I could not have done it without you, my wonderful parents and dear friends, Jeffrey, Gary, Cindy and Céline. Miss you Kitty...

# Introduction

Machine learning (ML) is in every respect *the* buzzword today, offering practical solutions to real-world problems by generalizing historical data to predict future data. Thanks to ML, natural language processing (NLP) is sought to automate innumerable manual workflows in processing large amounts of discrete texts with unprecedented efficiency.

Automatic taxonomy construction (ATC) is an important branch of NLP research that studies methods for extracting concepts and topics from domain-focused corpora, identifying entity relations and generating useful knowledge representations. As ML has ushered in the era of automatic classification, a myriad of data classification tasks are essentially managed through the use of multi-level (hierarchical) classification schemes in the form of a taxonomy. Although manually constructing and evaluating topic taxonomies can still be more flexible, comprehensive and often ineluctable, automated systems are expected to replace or supplement such manual efforts because they are time-consuming and require exceptional expertise in technical domains. As a result, a panoply of research has been conducted on ATC, leveraging various ML techniques, but the studies of machine learning and automatic taxonomy evaluation (ATE) have rarely crossed paths. This begs the question:

***Are machine learning and language modelling viable alternatives to the laborious manual taxonomy evaluation?***

In this work, we propose two evaluation strategies that bridge the gap between ML and automatic taxonomy scoring and ranking, while also guiding ATC and the optimization process. Specifically, we must address the situation in which obtaining a ground-truth or *gold-standard* is difficult and no external resources are available for validating concept relations.

*Science relies on the ability to reproduce and repeat results.* While most taxonomy assessments are marred with human judgements, we argue in this thesis that ATC evaluation should also shift towards a more systematic approach that prioritizes **quantifiability** and **reproducibility**.[1]

[1] Note that we may use the word taxonomy and ontology interchangeably throughout this work.

# Motivation

Although new automatic taxonomy extractors are constantly being proposed, currently there is no way to impartially answer two simple questions: *"which algorithm works best?"* and *"which taxonomy (among many) is the best?"*. For instance, changing the hyper-parameters of some parameterized ATC systems can result in drastically different outputs, but the optimal configuration remains unknown and it is unclear how to directly optimize the systems, since there are no automatic methods for assessing the structure and content of taxonomies. In other AI studies such as deep learning (DL), intrinsic measures such as loss functions enable artificial neural networks (ANN) to perform back-propagations and optimize their performance automatically, while external metrics can help data scientists evaluate and compare the performance of DL models in order to improve future results.

Consequently, the conundrum of ATC evaluation can stymie research progress and the call for quantitative taxonomy evaluation is justified. Because an ontology or taxonomy is typically developed in specific contexts to address problems or achieve goals, e.g. classification, its evaluation is evidently context-dependent, and many researchers believe that a task-independent automatic evaluation remains elusive, citing the lack of a general solution [**78**]. In fact, researchers have argued that objective evaluation metrics must be well available for significant progress in the development and deployment of taxonomies and ontologies [**13**].

# Approach

Automatic evaluation should focus on the structure, word relations and cluster coherence of a taxonomy. Our foremost problem in this thesis is the lack of external resources for assessing concept relations found in auto-generated topic taxonomies, as such resources are scarce and not publicly available for professional domains, which necessitates expert inputs and reduces the evaluation's reproducibility.

We discover that through training with documents from the evaluation domain, language models may be able to capture implicit concept relationships and become proxy domain experts, allowing us to evaluate semantic relations in a complex taxonomy without the use of external knowledge bases or search engines. For manually annotated corpora, we find that topic labels can be used to roughly estimate the ideal structure and topic distribution, so that we can convert text classification into a proxy application for taxonomy scoring. As such, we base our experiments with the following propositions that:

(1) Better taxonomies should result in higher scores on our evaluation tasks;
(2) Machine evaluation should correlate with human judgements;
(3) The evaluation procedures should be data and model-agnostic, with no reliance on a particular dataset or taxonomy/ontology extractor;

(4) The evaluation results should be reproducible.

To assess our proposed evaluation methods, we will attentively illustrate the structures of and topics in our extracted taxonomies, as well as creating low-biased (indicative) manual rankings for our evaluation targets. While it is nearly impossible to attempt comprehensive manual rankings on the vast majority of human-created taxonomies and ontologies, let alone cover their similarities and differences in depth, ranking automatically extracted taxonomies is far easier due to the experimental nature of ATC with many containing low-level noise and errors. We also design a *perturbation test* that gradually injects noise into a candidate taxonomy and alters the taxonomy through random deletions and swapping to simulate the degradation of taxonomy quality, as it is difficult to obtain ranked taxonomies with varying quality from the same dataset. Finally, we show that our evaluation results are consistent with the manual rankings and that our methods can generate scores independently for most datasets and ATC outputs.

## Challenge

Automatic taxonomy construction and evaluation are still in their infancy. To begin with, ATC systems involve a whole slew of moving parts and cannot always guarantee consistent results on any given datasets, which makes extracting topic taxonomies from texts other than those used in the original studies a challenge. Luckily, we were able to reproduce the results from research and extract taxonomy artifacts from other datasets and domains for our experiments.

Second, there have been few studies on the evaluation of auto-generated taxonomies, with most ontology evaluation research focusing on assessing Semantic Web ontologies and their components. While structural and semantic evaluations are prioritized, other critical aspects such as topic granularity, clarity and taxonomy usefulness are equally important but inherently difficult to be evaluated by humans and even more so by machines.

Our final hurdle is a lack of ground-truth taxonomies for our construction and evaluation tasks, especially for technical domains where existing resources and knowledge repositories are unavailable. Mining from a large repository, e.g. Wiki dump, is costly and time-consuming. Even if such ground-truth exists, manually or automatically mapping taxonomy concepts can pose another challenge. We hope that our work will provide insights and inspire future research in taxonomy evaluation, and that the methods can also be used in ontology and knowledge graph assessments and other information extraction tasks.

## Presentation

This thesis is presented largely in two parts: Chapter 1, 2, 3 will systematically introduce taxonomy, as well as state-of-the-art construction and evaluation methods; Chapter 4 and 5 will expand on the techniques discussed in Chapter 3 by proposing two new evaluation processes based on supervised classification and unsupervised language modelling, respectively. We thank IATA for providing the generous datasets and taxonomies, which have served as the foundation for our original research; however, the data will not be shown in this work, and instead public (albeit less interesting) data will be used.

# Chapter 1

# Defining Taxonomy

Integration of knowledge provides enormous opportunities for academics and industries and one of the most important knowledge representations is taxonomy. A subset of ontology, a topic taxonomy is a tree-like structure that not only helps with knowledge organization but also serves an integral part of many knowledge-rich applications including question answering, web search and recommendation systems. An ontology can also be regarded as a formal, explicit specification of a shared conceptualization [**33**].

This chapter introduces the concept of taxonomy as a knowledge representation, as well as the study of taxonomy learning, the significance of taxonomy and its relevance in practical applications.

## 1.1. Taxonomy Learning

Taxonomy learning is a comprehensive study in NLP that encompasses everything from automatically constructing taxonomies from texts to using taxonomies to improve a wide range of applications. In figure 1.1 we show a simple automatically constructed taxonomy from a scientific domain. A glut of applied research has been conducted on *hypernymy discovery and lexical entailment* [**82, 6, 91, 68, 17, 114**], instance-based *concept taxonomy construction* [**36, 94, 77, 112, 28, 58, 70**], clustering-based *topic taxonomy construction* [**56, 115, 87, 41**], *taxonomy expansion and enrichment* [**90, 89, 116, 43, 52**], *taxonomy*

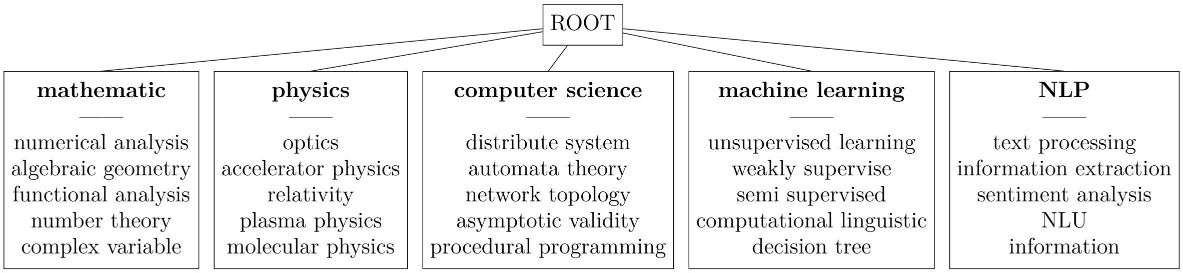


**Fig. 1.1.** An auto-constructed (and possibly erroneous) taxonomy

*applications*, e.g. improving recommender systems and information retrieval [**59, 120, 119, 109**], as well as *taxonomy evaluation* [**18, 2, 11, 10, 43**] [**88**].

While most taxonomy learning tasks are self-explanatory, hypernym discovery is the process of extracting relevant hypernyms or *super-types* from a text corpus given some candidate hyponyms or *sub-types*, whereas lexical entailment is concerned with identifying the semantic relations between two words. An important research area in information retrieval (IR), hypernym discovery can be regarded as the forerunner of automatic taxonomy construction (ATC) as it aims to detect potential hypernym-hyponym pairs without taking into account the taxonomy structure. At SemEval-2018 Task 9, researchers are asked to predict and rank selected hypernyms from a large pool of candidate vocabulary, e.g. attributing *athlete, sportsperson, person, competitor, sport, olympic sport...* to query subject "Suzy Favor Hamilton" [**6**].

In our literature review, we found that the assessment of taxonomy learning commonly makes use of existing ground-truths such as WordNet [**24**], Wikidata and ResearchCyc [**76**], accompanied by expert annotations and judgments. For hypernym discovery, *ranking measures* are used including mean reciprocal rank (MRR), mean average precision (MAP) and precision of top-$k$ retrieval.

## 1.2. Taxonomy vs. Ontology vs. Knowledge Graph

While knowledge graphs (KG) may have become the spotlight of AI, taxonomy, ontology and KG are all indispensable knowledge representations that manage concept entities and denote their underlying relations for later exploitation. Figure 1.2 showcases the three essential knowledge representations.

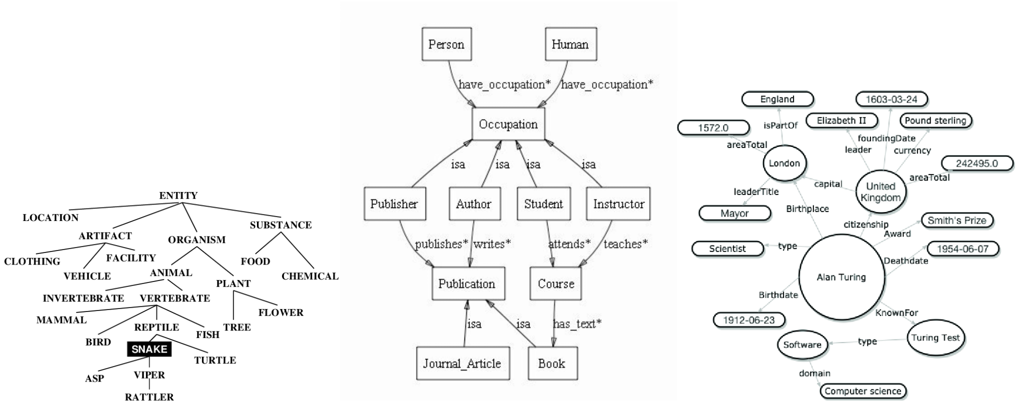


**Fig. 1.2.** Taxonomy, ontology and knowledge graph (from left to right)

## Taxonomy

From Latin *taxis*, meaning "order" and *nomos* meaning law, a taxonomy is a simple tree-like knowledge map, a semantic representation that uses controlled vocabulary and a static classification scheme for many applications. It can also be called a *dendrogram* in hierarchical clustering. Here we summarize the principle components of a typical topic taxonomy:

- **Node/entity**: otherwise known as a "taxa", a taxonomy entity is comprised of a concept or a topic;
- **Edge**: denotes the relation between a parent node and a child node. Subsumption or "is-a" relation is the most common relationship in topic taxonomies, i.e. a child topic *is a* type of the parent topic.

It is easier to include some key terminologies for denoting the main semantic relations when describing a taxonomy (and other knowledge representations):

- **Hypernym, hyponym**: describe the semantic relation between a "super-type" (parent) and a "sub-type" (child) from adjacent taxonomic levels, e.g. "fruit" and "apple";
- **Holonym, meronym**: contrary to hypernym and hyponym, meronym signifies a "part-of-relationship" while holonym denotes a "whole", e.g. a "core" is a meronym (part) of apple (holonym) while "macintosh" is a hyponym (type) of apple (hypernym);
- **Synonym**: words that share similar semantic meanings and are siblings under the same hypernym, e.g. "apple" and "orange".

Moreover, **granularity** is a critical concept that describes the level of detail in all knowledge representations. For instance, "city" can be a more detailed or *fine-grained* concept than "country"; A four-level taxonomy of *country, province, city and street* is finer-grained than a two-level one of *country and city.* Topic-wise, "Montreal" (city name) may share similar granularity with "Toronto" (city name) than with "Canada" (country name). A good taxonomy should host topics of the same granularity at each level (Montreal and Toronto) as compared to a mixed granularity (Montreal and Canada).

## Topic taxonomy: a formal description

We attempt to formally describe a *topic taxonomy* referring to the description of Shang et al. [**87**]. The target of automatic taxonomy construction and evaluation in our work is a tree-like structure $\mathcal{T}$ of topic nodes $c \in \mathcal{T}$ that ideally represents the domain of interest of document corpus $\mathcal{D}$, where each node $c$ is represented by an "anchor term" and may include a list of alternative representations that we designate otherwise as "neighbours", e.g. "software agents" and "multi-agent systems" for "intelligent agents". Parent-child pairs in $\mathcal{T}$

should follow a hypernym-hyponym relation, i.e. for each parent node $c$, its set of children $S_c = c_1, c_2, ..., c_n$ should all be subtopics of $c$ and share the same granularity.

### Ontology

From Greek *onto*, meaning *being* and *logia* meaning "logical discourse", ontology is a super-set of and is more structurally complex than a taxonomy. In contrast to the tree-like structure of a taxonomy, the typical structure of an ontology resembles a "web". The smallest unit of an ontology is called an *axiom*, which can either be a terminological axiom, a fact or an annotation, all of which are beyond the scope of this thesis.

Vrandečić in *Ontology Evaluation* [**104**] summarized that an ontology: (i) specifies a conceptualization, (ii) consists of a set of axioms, (iii) is expressed by an ontology document, and (iv) constrains the construction of models satisfying the ontology. Furthermore, web ontologies are typically encoded using ontology languages.

### Knowledge Graph

Knowledge graphs are an agglomeration of in-domain ontologies, usually created by intelligent systems (IS) and are frequently referred to as a large scale semantic network consisting of entities and concepts connected via various semantic relations. In table 1.1, we summarize the differences and similarities of taxonomy, ontology and KG.

| | **Taxonomy** | **Ontology** | **Knowledge Graph** |
|---|---|---|---|
| **Node** | Concept/topic | Individual/class | Object/event/ situation/concept |
| **Unit** | Taxa | Axiom | "Fact" |
| **Relationship** | Hypernym-hyponym (is-a) | Multiple relationships/ properties | Triplets:<br>e.g. Subject/Predicate/Object |
| **Topology** | Tree | Web | Graph (acyclic) |
| **Size** | Small | Medium | Large |
| **Example** | Amazon Product Categories | WordNet | Google Knowledge Panel |

**Table 1.1.** Overview of taxonomy, ontology and knowledge graph

## 1.3. Taxonomy Applications

Modern information systems are shifting from data processing towards concept processing [**11**]. Impressively, some of the most popular ontology products in NLP can range from web search to dialogue systems. To illustrate the potentials of automatically constructed taxonomies and ontologies, this section provides several examples of important taxonomy applications and how they are integrated in modern information systems.

## Bio-medicine and Clinical Care

In the realm of medicine, a medical ontology depicts the ideas behind medical terminologies and how they relate to one another. It also facilitates the storage and exchange of extensive medical knowledge. With the advancement of NLP, clinicians and medical researchers seek more effective methods of curating biomedical data and documenting care in the electronic health record (EHR) [**45**].

For example, some protein-protein interaction ontologies are created in bio-ontological studies for exploiting complex experimental data and excavate new information [**49**]; in patient care, clinicians used to document clinical findings and patient symptoms in a free-text format in EHR, which made it difficult to find information and optimize care. Using NLP, clinical texts can be mapped onto ontology concepts and stored in a more refined, structured and systematic manner that facilitates future retrieval.

## Semantic Web

Ontology is a vital component of the World Wide Web (WWW) and is used extensively for web content management. Resource description framework or RDF is used for representing web data and concepts and for exchanging of ontological information. In the context of the Semantic Web, an ontology is a formal explicit description of concepts in a domain of discourse with the *properties* of each concept describing features and attributes of the concept [**69**]. Ontology engineering focuses on searching, crawling, classifying and ranking Semantic Web ontologies.

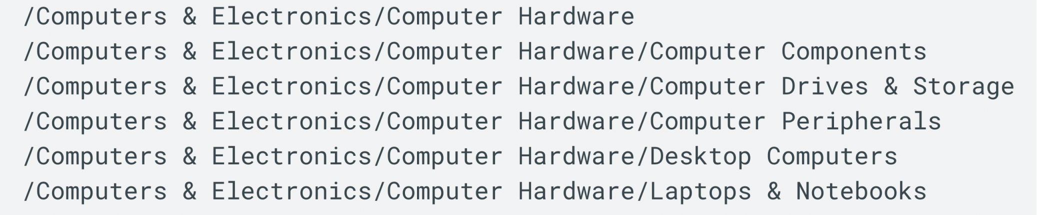
Below we summarize five points of usage of ontologies in the Semantic Web [**69**]:

- To share common understanding of the structure of information among people or software agents;
- To enable reuse of domain knowledge;
- To make domain assumptions explicit;
- To separate domain knowledge from the operational knowledge;
- To analyze domain knowledge.

## Product and Data Management

Taxonomies are inevitably used as business and product catalogs and for managing online sales. Notable ontology products in this domain include Amazon Category Taxonomy, Google Product Taxonomy, Yelp Business Category and Google Content Categories. Logistics use taxonomies for inventory and resource management while recommendation systems leverage taxonomies for suggesting similar or highly-related items. In addition, question answering (Q&A) systems use taxonomies for identifying relations and formulate queries.

Due to their technicality, domain ontologies are commonly handcrafted by domain experts and used for large-scale classification. The largest trade association of the world's airlines with member airlines from over 117 countries, IATA operates a complex multi-level ontology for classifying and analyzing a plethora of airline-submitted aviation incident reports and for detecting recurring aircraft defects.



```
/Computers & Electronics/Computer Hardware
/Computers & Electronics/Computer Hardware/Computer Components
/Computers & Electronics/Computer Hardware/Computer Drives & Storage
/Computers & Electronics/Computer Hardware/Computer Peripherals
/Computers & Electronics/Computer Hardware/Desktop Computers
/Computers & Electronics/Computer Hardware/Laptops & Notebooks
```

**Fig. 1.3.** Excerpt of Google Content Categories [1]

## 1.4. Taxonomy and Large-scale Classification Systems[2]

As mentioned earlier, taxonomies are conventionally used as a hierarchical classification scheme. The content of taxonomies, on the other hand, is rarely static: as more entities are added to a taxonomy, classification performance may begin to deteriorate due to overlapping topics, labelling errors, skewed label distributions and lack of samples.

Improving the *scalability* of text classifiers has since become an important research topic, typically through feature selection and adopting ensemble techniques that include *voting*, *bagging* and *boosting* [**32**]. Companies are regularly faced with the dilemma between training a single classifier per taxonomic level that needs to be discarded or retrained due to potential taxonomic changes, or maintaining a large swath of binary classifiers for each taxonomic node, which is practically impossible due to the number of local classifiers scaling up with taxonomy size increase. Recently, text-to-text approaches aimed at training and fine-tuning "text generation" models, e.g. Google's T5 and DeepMind's GPT, have gained attention for enabling hierarchical classification with large-scale taxonomies, in which simple textual descriptions containing a task prompt (in our case "multi-label classification:") and some contexts (e.g. "runway excursion") can be used directly as model input for the models to generate the corresponding taxonomic labels (e.g. "1001, 7003"). Moreover, only one text generation model needs to be trained per taxonomic level [**37**].

[1] https://cloud.google.com/natural-language/docs/categories

[2] Memo for my IATA internships about large-scale hierarchical classification using complex ontologies

## 1.5. Essential Ontologies in Research

Several ontologies have become indispensable and widely-adopted as ground-truths and sources of knowledge for taxonomy construction and evaluation, as well as other knowledge-rich NLP tasks. We illustrate three such ontologies reviewed in our research.

### WordNet

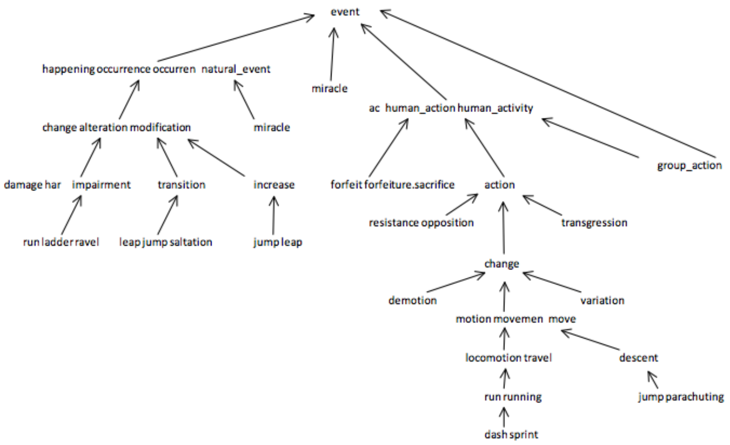


**Fig. 1.4.** A subset of WordNet topic "event"

A "taxonomy of words", WordNet [**24**] is meticulously curated by computational linguists and serves as a computer-readable lexical dictionary and memory for common-sense knowledge. WordNet's basic relationships are synonymy, antonymy, hyponymy, meronymy, troponymy and entailment (relations between verbs). In taxonomy construction, WordNet can be used for semantic look-up and disambiguation. In taxonomy evaluation, WordNet is used as the ground-truth for measuring concept similarities and as the target for testing a system's performance in a taxonomy reconstruction task.

WordNet is easily accessible via APIs such as NLTK. However, it is commonly restricted to generic usage due to its limited coverage of only 25,229 concepts [**109**], which also omits domain-specific concepts.

### Yago

YAGO (Yet Another Great Ontology) is a major ontology mining milestone published in 2007 that was automatically sourced from WordNet and Wikipedia using a combination of rule-based and heuristic methods, yielding 352,297 ontological concepts [**96**]. It is also designed to be easily extendable and represents a significant improvement over WordNet, which was created entirely manually due to ATC bottlenecks two decades ago. YAGO has achieved 95% accuracy assessed by human evaluators through sampling random facts from the ontology (since there are no computer-processable ground-truths available).

In the end, Yago is often used as a backend for knowledge-rich applications including entity-linking, question answering and conversational AI [**29**].

## Probase

Probase is a probabilistic framework harnessing 2.7 million concepts mined from 1.68 billion web pages [**109**]. Therefore when compared to the two previous ontologies, Probase has a higher chance of including a concept even if it comes from a domain-specific corpus, which is why it is used in numerous taxonomy learning tasks such as taxonomy enrichment. An outcome of its probabilistic nature, Probase can also provide a likelihood estimation for each concept pair by considering context evidence. For example, sentences like *"Spanish artists such as Pablo Picasso ..."* are considered evidence for the claim that *Pablo Picasso* is an instance of the concept *Spanish artist.* As such, Probase can theoretically be used for taxonomy evaluation if it possesses knowledge in the evaluation domain.

# Chapter 2

# Automatic Taxonomy Construction

Most existing taxonomy construction methods organize hypernym-hyponym entities into a tree-like structure to form an instance taxonomy [**41**]. It is at the forefront to understand the ATC paradigm and its essential steps in order to design appropriate evaluation strategies. To include all aspects of automatic taxonomy construction requires monumental effort. As such, this chapter will focus on the fundamentals of most ATC pipelines.

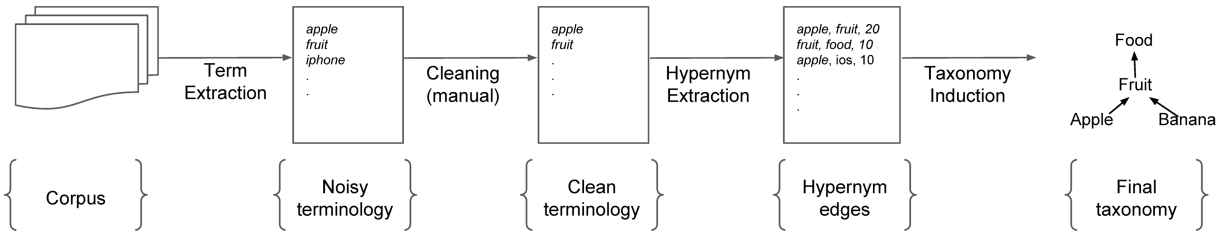


**Fig. 2.1.** A typical taxonomy induction pipeline (Liu et al. 2012)

With a goal of distilling unstructured data into structured knowledge, automatic taxonomy construction typically consists of: (1) terms and concept extraction from a text corpus and (2) a relation formation step that identifies concept relations based on the given corpus [**112**]. Unlike manually annotated taxonomies with more flexibility with topic naming (as they are primarily used by experts), ATC relies on rearranging existing topics found in a corpus into meaning hierarchies. Figure 2.1 demonstrates a typical pipeline of the taxonomy induction process. The concept terms, often hypernym-hyponym pairs introduced in Chapter 1, are either modelled using (i) an existing knowledge base (KB) or a lexical database such as WordNet Synsets (*portmanteau* for "synonym sets") and excerpts of Wikipedia [**94**] through *concept mapping*, or using (ii) *pattern-based* or *statistic-based* approaches that infer concept-pair relations from large text corpora. Recent SOTA taxonomy extractors have also experimented with machine learning and deep learning techniques such as hierarchical clustering that iteratively categorizes keywords into dendrograms (hierarchical clusters) [**56**] and *transfer learning* for detecting hypernymy pairs. The backbone of the SOTA approaches, however, is continuous numerical word representations processable by machine learning algorithms that capture the latent semantics and contexts of topics [**4, 62**].

The input of an ATC system are usually unstructured texts from a domain of interest, often accompanied by a list of relevant keywords that are curated manually or identified by some topic mining tools; The output of an ATC program is a structured document typically comprised of organized topic clusters of representative terms, indicating subsumption ("is-a") relations. While some systems also apply query expansion techniques to expand an extracted taxonomy by pulling related topics and concepts from a large lexical ontology or the web, some recent "seed-guided" methods make use of language models and a small seeded taxonomy for finding similar relations in context.

ATC algorithms can be roughly categorized into supervised, weakly supervised and unsupervised methods that are suitable for different use scenarios. Figure 2.2 provides a brief overview of automatic taxonomy construction methods. The following sections will introduce two principle ATC paradigms, an pattern-based approach and a distributional (statistical) approach, as well as SOTA extractors used for taxonomy building in this thesis.

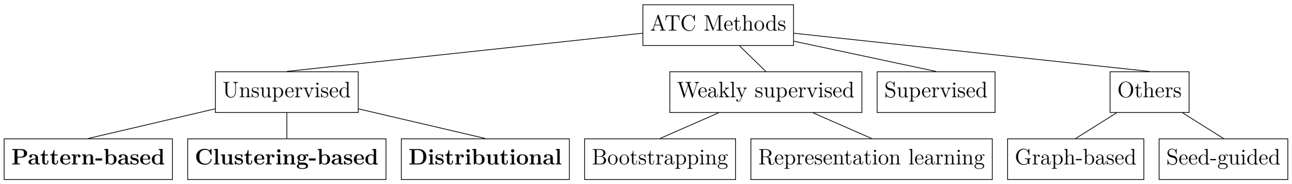


**Fig. 2.2.** A taxonomy of ATC methods

## 2.1. Pattern-based approach

Early ontology construction methods depended on pattern-based approaches that require handcrafted linguistic features and text filters, primarily Hearst patterns [**39**]. Hearst patterns are first developed as a set of simple heuristics and have since played a central role in extracting knowledge, thanks to their intuitive, easy-to-implement and human-comprehensible nature. Hearst patterns distill knowledge by exploiting *lexical-syntactic* patterns [**66, 47, 58, 5, 48**] that may indicate hypernymy relations in a corpus, i.e. locating sentences containing "such as", "including" etc., as exemplified below, where $NP$ denotes noun phrases:

- $NP_0$ **such as** $NP_1$, $NP_2$, ..., (and | or) $NP_n$
  e.g.: the *bow lute* such as the *bambara ndang.*

- $NP$ , $NP$* , **or other** $NP$
  e.g.: *Bruises, wounds, broken bones* or other *injuries.*
- $NP$ , **including** $NP$ ,* or | and $NP$
  e.g.: All *common-law countries*, including *Canada* and *England.*

Later pattern-based methods would include even more linguistic rules and generalized patterns such as "star-pattern", "meta-pattern" and "SOL pattern" [**58, 81, 47, 65, 42, 64, 115**]. Aside from using a lexical database such as WordNet to obtain quantifiable scores including term coverage, precision and recall for topic pairs extracted using Hearst patterns, queries containing the above lexical patterns can also be submitted to a search engine for computing an *evidence score* via the search results [**58**].

Pattern-based methods are fast and convenient for gathering potential hypernym-hyponym pairs in a domain corpus as they do not require preparatory steps such as key term extraction. With all its benefits, a pattern-based approach can achieve very high precision but relatively low recall and low coverage tradeoff since it is impossible to manually annotate all lexical patterns, and newly included lexical rules can also negatively affect certain existing rules. Some researchers opt to train the systems on massive corpora to offset the low recall [**109**], as studies have indicated that a pattern has a higher chance to explicitly appear in a corpus such as the web if the size of the corpus is nearly unlimited [**23, 48, 112**]. Further, research has also proposed weakly-supervised automatic bootstrapping methods to learn new relation instances and improve low recall [**39, 94, 112**].

## 2.2. Distributional Approach

Distributional or statistical ATC methods seek to improve the sparse coverage induced by using handcrafted linguistic features above, which also lead to significantly more complex systems. However, there is currently no standardized approach to statistical taxonomy extraction. New methods have been constantly proposed for generating meaningful topic clusters or recognizing hypernymy relations using machine learning, with some of them also leveraging meta-data and other text-rich features [**87**]. Generally, SOTA automatic taxonomy construction can be regarded as a complex hierarchical embedding generation and clustering problem with the current paradigm consisting of term extraction, generating distributed word representations, iterative clustering and structural optimization, which we will describe in detail.

### 2.2.1. Feature Representation

Raw texts must be converted into vectorized, numerical representations for machine clustering and classification. Classical machine learning before neural language models counted on *frequency-based* feature extraction methods, which use word occurrence to represent a feature in the vector space. The methods are based on the assumption that a textual sequence can be embodied by a simplified view also known as *bag-of-words* (BoW) that disregards word orders. For instance, the sentence *"Montreal is in Quebec."* can be represented by a list of four separate words: "Montreal", "is", "in", "Quebec", even though the same features can also represent *"Quebec is in Montreal"*. A canonical frequency-based feature transformation is count vectorization, by counting up words in a predefined vocabulary in each document, as illustrated in table 2.1.

| | "this" | "is" | "the" | "first" | "sentence" | "second" | "and" |
|---|---|---|---|---|---|---|---|
| "this is the first sentence" | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| "this is the second sentence" | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| "the first and the second sentence" | 0 | 0 | 2 | 1 | 1 | 1 | 1 |

**Table 2.1.** Count vectorization: an example

Note that in table 2.1, it might appear that "sentence" is the most representative word of the three examples as all phrases contain at least one occurence of it. However, article "the" also appears in all the sentences and is non-informative. To counter this, *pre-processing* techniques such as removing "stop-words" (e.g. "the", "would") are required. As such, frequency-based methods are efficient to compute by most hardware but cannot capture any word semantics. Another issue arises when the feature vectors become sparse (consisting of mostly zero values) when the number of words in the vocabulary increases, suffering the *curse of dimensionality*.

An improvement to the standard count vectorizer is the term frequency - inverse document frequency or TF-IDF method, in which a TF-IDF feature is computed by term frequency (TF) * inverse document frequency (IDF). TF-IDF is an indispensable tool in NLP with a goal of scoring the significance of each term to a document using a bag-of-word representation. The term frequency feature shares similarity with the count vectorizer by measuring how important a term $t$ is to a document $d$ :

```
TF(t,d) = count of term in document / number of words in document
```

While the inverse document frequency (IDF) feature measures how informative a term is in the document corpus of size $N$, i.e. if a word such as a stop-word appears in every document, then it does not convey any useful information due to low entropy:

```
DF(t) = number of documents in which the term occurs
IDF(t) = log(N/(DF(t)+1)), add one is used to prevent divide-by-zero
```

Combining TF and IDF features, the TF-IDF score for a document term is therefore:

```
  TF-IDF(t, d) = TF(t,d) * log(N/(df+1))
```

In the 2000s, distributional methods were developed for probabilistic language models that assign each term into a separate, continuous vector space and generate word representations called *word embeddings* in fixed-sized, low-dimensional vectors, based on the surrounding contexts [**4**]. Such models are trained with massive amount of data over various unsupervised language modelling objectives including Skip-gram and CBoW (Continuous Bag-of-Words) as described by Word2Vec [**63, 62**], producing *static* word vectors for the target vocabulary that allow words in similar contexts (surrounding words) to have closer distance. The word embeddings produced by methods such as Word2Vec are ubiquitous in most ATC systems for modelling topic similarities and relations from unlabelled texts.

For ATC, the discriminative power of word embeddings can significantly limit the accuracy of hierarchical clustering and its output, since clustering algorithms may struggle to segregate topics with close embedding distance with higher granularity. As a result, some studies propose to use "locally trained embeddings" [**115, 7, 22**] to calculate word vectors with more discriminative power by iteratively fine-tuning global embeddings with weighted documents containing fine-grained topics [**87**].

Another drawback of static word embeddings is that the vector value is averaged across all word senses. With the breakthrough in language modelling, newest ATC methods also leverage *contextualized embeddings* generated by the Transformers architecture. Transformers use *self-attention mechanism* [**102**] to generate *dynamic* embeddings through a feed-forward neural network, and produce feature vectors for words in different contexts, (e.g. homonymy "bat" as in "baseball bat" or "bat the mammal"), as opposed to single static word embeddings. ELMo [**75**] is the first method to follow such intuition to calculate contextualized embeddings, by using LSTM (Long Short-Term Memory) [**40**] and a language modelling task called *next word prediction*. The contextualized embeddings are then generated by concatenating the hidden layers of the language model. BERT [**21**] improved upon the ELMo process by using auto-encoders and *masked language modelling* that looks both ways for contexts. New ATC systems such as CoRel [**41**] use the values of the initial layer of BERT as word embeddings for clustering input. [1]

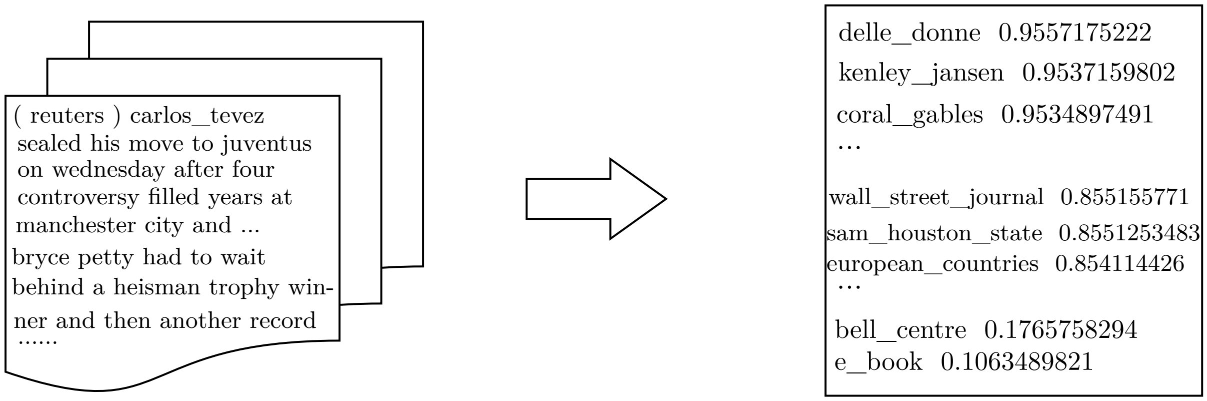
### 2.2.2. Term Extraction

A dedicated study in domain knowledge acquisition and information retrieval, automatic term or keyword extraction is essential in most ATC pipelines and involves a whole new set of

[1] The success of Transformer language models also enabled downstream tasks such as text classification and summarization.

challenges, including identifying domain terminologies and ranking a list of extracted terms based on certain criteria such as "relevancy" and "term integrity". With a readily identified corpus domain, a list of keywords can be manually curated or chosen from existing ontologies such as WordNet [**56**]. Some popular statistical models for corpus keyword extraction are Rake[2] [**83**] and YAKE![3]. These methods are used in acquiring knowledge from domain-specific texts, eclipsing the performance of previous linguistic analyzing approaches relying on techniques such as *dependency parsing.*

We decide to include in this section a SOTA automatic *phrase mining* tool called AutoPhrase that is used extensively in our tested ATC systems. AutoPhrase leverages a part-of-speech (POS) guided phrasal segmentation model that incorporates shallow syntactic information existing in POS-tags and as general large knowledge bases such as Wikipedia to train a performant identifier of quality phrases [**86**].


( reuters ) carlos_tevez sealed his move to juventus on wednesday after four controversy filled years at manchester city and ...
bryce petty had to wait behind a heisman trophy winner and then another record
......
delle_donne 0.9557175222
kenley_jansen 0.9537159802
coral_gables 0.9534897491
...
wall_street_journal 0.855155771
sam_houston_state 0.8551253483
european_countries 0.854114426
...
bell_centre 0.1765758294
e_book 0.1063489821

(Document Corpus) (AutoPhrase Output)

**Fig. 2.3.** AutoPhrase output with term integrity scores

AutoPhrase accelerates taxonomy construction in that it can generate quality keywords and provide a term integrity score for each extracted keyword as shown in figure 2.3 that is useful for ranking and for indicating keyword importance. It is also domain-independent, although providing a list of domain keywords as reference will guide the algorithm to extract better quality topics. Using the AutoPhrase output, a term integrity threshold can be set for selecting the top quality *noun phrases* [4] or multi-word keywords for tagging the corpus, a process known as *phrasal segmentation*, so that each multi-word noun phrase cannot be split when calculating word embeddings or as different topic labels in a taxonomy. In the following example, phrasal segmentation links two entities of interest "Chinese food" and "Mexican food" using underscores.

[2] `https://github.com/aneesha/RAKE`

[3] `https://github.com/LIAAD/yake`

[4] technically, a phrase is defined as a sequence of words that appear consecutively in the text, forming a complete semantic unit [**25**].

```
Original Sentence: "I love Chinese food and I love Mexican food."
Phrasal Segmentation: "I love Chinese_food and I love Mexican_food."
```

In this case, the problems we encounter with AutoPhrase lie twofold: (1) how to interpret the term integrity score and to configure the appropriate keyword threshold for taxonomy construction. (2) as a list of reference keywords is unprovided, the key terms found by the algorithm are often unfocused and can include words other than nouns and pronouns. Some of the good keyword entities, e.g. "e_book" in figure 2.3 can also have low scores.

### 2.2.3. Clustering

Hierarchical clustering is innate for ATC since it inherently generates a dendrogram (tree structure) of topics that is easy to interpret by humans [**56**]. Clustering-based ATC methods or iterative clustering in general uses two partition strategies: an *agglomerative method* is a bottom-up approach merging two most similar clusters at every step, while a *divisive* method follows a top-down principle that splits a cluster into two or more fine-grained clusters at each step. Recent clustering-based ATC methods often follow the divisive principle to first learn a representation space of word embeddings for the terms of interest, then separate the terms into an appropriate taxonomic structure, usually by identifying hypernymy-hyponymy relations [**56, 106, 112, 41**]. Besides, a "conceptual clustering" method for taxonomy construction has also came to popularity [**15**].

Clustering algorithms mainly focus on two types of data characteristics: algorithms namely k-means use *compactness* to measure the distances between data-points and cluster centers, where similar datapoints are assumed scattering compactly around their cluster centers; other algorithms such as spectral clustering emphasize *connectivity*, assuming similar data are connected to each other and form continuance on a graph. We can observe such traits in figure 2.4, in which partitions are created by clustering algorithms, denoted by colours, on four different 2D shapes comprised of datapoints (circles, half-moons, blobs and no-structure). The bottom-right number in each sub-figure denotes the algorithm's running time on the dataset for comparison purposes, where MiniBatchKMeans (a variation of the standard k-means) has demonstrated remarkable efficiency. In the circles and half-moons examples, MiniBatchKMeans and spectral clustering each produced two distinct partitions based on proximity to cluster centroids versus continuity, even though the latter appears to be more intuitive for humans. The bottom case depicts a "null" situation where the algorithms attempt to segregate on homogenous data, and thus no optimal strategies are guaranteed. Other clustering methods, e.g. affinity propagation [**41**] and agglomerative clustering, are used for obtaining optimal taxonomy subtopics without specification of the number of clusters. In this case, the number of clusters selected is manually reduced for improved visualization using hyper-parameters. Aside from using compactness and connectivity, DBSCAN, a *density-based* method, has demonstrated its effectiveness such as on the blobs example, detecting three rather than two cluster groups. It does not however work well on overlapping clusters where no drop in density is observed, nor does it work well with categorical features.

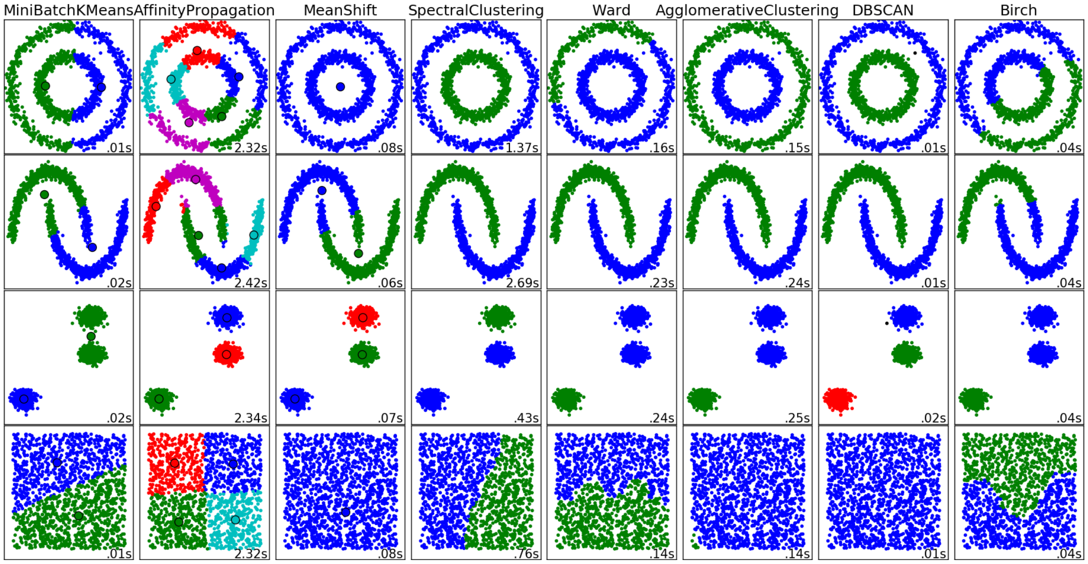


**Fig. 2.4.** Partitions of popular compactness and connectivity-based clustering algorithms on four different data shapes (Original figure from scikit-learn)

We want to specifically discuss k-means, which is perhaps the most well-studied clustering algorithm and unsupervised learning method. K-means starts with a predefined number $k$ of *centroids* with the objective of partitioning datapoints into $k$ different partitions. K-means follows the *within-cluster squared-error criterion* and does so by iteratively initializing $k$ centroids randomly then assigning and updating steps to minimize the distances of each datapoint to its closest centroid until convergence. In other words, the goal of k-means is to optimally position $k$ cluster centers so that the surrounding data are closest to the cluster centers. K-means aims to minimize the total *sum of squared error*. The equation for the k-means minimization problem is rather self-explanatory where $\mathcal{J}$ is the sum of all distances from the datapoints to the cluster centroids, as shown in equation 2.2.1:

$$\mathcal{J}(x) = \sum_{k=1}^{K} \sum_{x_i \in C_k} (x_i - \bar{x}_k)^2 \tag{2.2.1}$$

While effective and practical, k-means suffers from two problems where (1) it is often non-trivial to find and decide on the optimal $k$ for a particular solution. Determining the optimal value $k$, however, relies on proficiency in the domain knowledge, empirical study or using tools such as an elbow graph; (2) k-means algorithm may converge to a *local minima* instead of a *global minima* and may require multiple runs for the best solution.

Spherical k-means is a variation of the conventional k-means algorithm [**3**] and is used extensively in TaxoGen [**115**], by projecting the estimated centroids onto a unit sphere instead of a plane during each step update.

### 2.2.4. Supervised Relation Classifier

The last point we want to mention in this section is the novel use of language models for classifying hypernymy relationships. We use the example provided in CoRel [**41**] to illustrate the use of a relation classifier in ATC. Given a positive parent-child pair provided in a seeded taxonomy, in this case "dessert" and "ice cream", the ATC system assumes that a sentence implies their relation if it contains the two entities and gathers all sentences contained in a collection of texts with co-occurrences of the topic pair. The negative samples can be gathered by combining a topic with another non-sibling entity in the seed taxonomy, e.g. "dessert" and "salad". The training target of the classifier is therefore binary in that the topic pair is either valid or non-valid, although CoRel trains a model with three output targets, parent-child ("ice cream" is-a "dessert"), child-parent ("dessert" is-a "ice cream") and non-relation.

```
Training input: We don’t serve [MASK] today except for [MASK]
Masked entities: desserts, ice_cream
```

The supervised relation classifier is intuitive and can achieve high accuracy, provided with sufficient positive samples. However, the approach faces critical challenges in real-world usage because (1) it is difficult to find parent-child co-occurence in a moderately-sized corpus and many valid parent-child pairs are sparsely scattered in different sentences. (2) a technique to gather more positive samples is to experiment with various seeded taxonomies and choose the one that generates the most positive samples, but this is incredibly arduous and requires prior knowledge of the corpus, which defeats the purpose of automatic extraction.

## 2.3. Tested Systems

As ATC remains highly experimental, we are grateful to have discovered some recent state-of-the-art taxonomy extractors available publicly at a very early stage and that we were able to adopt the code and replicate the published results. It is shown that the current SOTA implementations can produce quality results on selected datasets and taxonomy domains. We describe in 2.3.1 four tested systems that have directly enabled our taxonomy research and illustrate in 2.3.3 a sample output for each algorithm.

### 2.3.1. Description

**TaxoGen** [**115**] is an adaptive text embedding and clustering algorithm leveraging various phrase-mining and clustering techniques including AutoPhrase [**86**], caseOLAP [**54**] and spherical k-means clustering [**3**]. TaxoGen iteratively refines selected keywords and chooses cluster representative terms based on two criteria: *popularity* which prefers term-frequency in a cluster and *concentration* which assumes that representative terms should be more relevant to their belonging clusters than their sibling clusters. The system can be configured with several hyper-parameters including the depth of the taxonomy, the number of children per parent term and the "representativeness" threshold. Experiments were conducted on DBLP and SP (Signal Processing) datasets and the system is quantitatively evaluated with relation accuracy and term coherency measures assessed by human evaluators (10 doctoral students).

**CoRel** [**41**] takes advantages of novel relation transferring and concept learning techniques and uses hypernym-hyponym pairs provided in a seeded taxonomy to help train a deep learning (BERT) relation classifier and expand the seeded taxonomy horizontally (width expansion) and vertically (depth expansion). Topical clusters are generated using pre-computed BERT embeddings and a discriminative embedding space is learned, so that each concept is surrounded by its representative terms. The clustering algorithms used by CoRel are *spectral co-clustering* [**46**] and *affinity propagation* [**27**], which automatically computes the optimal number of topic clusters. Compared to TaxoGen, CoRel does not require depth and cluster number specifications but a small seeding taxonomy as an input for enabling a weakly-supervised relation classifier as discussed in section 2.2.4 based on BERT [**21**]. Each pair of the provided hypernym-hyponym pair in the seeded taxonomy is used for gathering positive training examples by CoRel to find other similar topic pairs. CoRel is quantitatively evaluated with term coherency, relation f1 and sibling distinctiveness judged by 5 computer science students on subsets of DBLP and Yelp datasets. The system generates outputs in the form of large hierarchical topic word clusters.

**HiExpan** [**90**] differs from the aforementioned constructors in that it is a hierarchical tree expansion framework, which aims to dynamically expand a seeded taxonomy horizontally (width expansion) and vertically (depth expansion) and performs entity linking with Microsoft's Probase introduced in section 1.5 to iteratively grow a seeded taxonomy. As an entity is matched against a verified knowledge base, the accuracy of terms and concept relations is perceived to be higher than than CoRel and TaxoGen. For quantitative evaluation, HiExpan invited its authors and some volunteers to assess the taxonomy parent-child pair relations using ancestor-f1 and edge-f1.

**TaxoCom** [**52**] is a novel framework improved upon TaxoGen, leveraging local discriminative word embeddings that are more performant in clustering fine-grained topics. TaxoCom also defines and implements a new term significance metric and requires a seeded input for completion and discovering new cluster topics. Evaluation of TaxoCom is conducted on term coherency and topic completeness judged by 10 doctoral researchers on the NYT news and arXiv datasets.

### 2.3.2. Observation

Each of our tested systems faces its own set of advantages and drawbacks. TaxoGen and TaxoCom are the only parameterized systems in our experiments and only TaxoGen requires no seeded input for producing an output, which can be beneficial when prior knowledge of the corpus is lacking [5]. Additionally, TaxoGen and TaxoCom also generate alternative synonyms for each taxonomy topic, which increases the coverage and improves concept mapping between taxonomies and documents. However, TaxoGen and TaxoCom seem to depend on the keyword extraction quality and it is unclear how to determine the best hyper-parameter settings for the systems owing to the lack of evaluation methods.

CoRel uses the concept pairs provided in the taxonomy seeds for mining similar relations, but this has become its Achilles' heel because same-sentence co-occurrence of valid parent-child topics is rare in real-world data. As a result, CoRel may fail to produce any output at all due to insufficient training examples for the relation classifier. It is also resource-intensive for making use of neural networks for relation transferring and depth expansion. Anecdotally, the output of CoRel may also not be entirely exhaustive and deterministic. Below are two runs of CoRel that have produced two distinct results for topic "fuel" extracted from an aviation corpus:

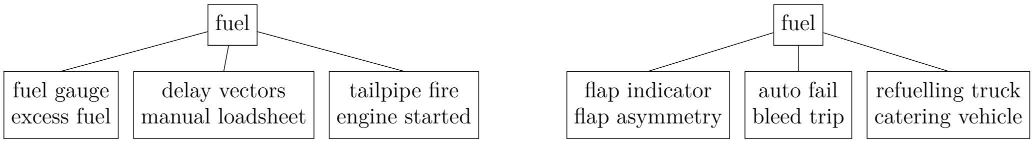


**Fig. 2.5.** Different CoRel runs generate different results

For our experiments, HiExpan is perceived to produce the most consistent taxonomies thanks to the use of Probase for measuring topic similarities and locating related concepts. However, the set-expansion mechanism of HiExpan often ignores topic granularity and adds hyponyms and hypernyms found in similar contexts to the exact same taxonomy level (hence most HiExpan taxonomies are two-level only). It also cannot differentiate word senses such as virus as in *computer virus* and a *viral disease*.

[5] This has inspired us to study using a "seed-less" system such as TaxoGen for bootstrapping systems that require seeded inputs (CoRel, HiExpan).

### 2.3.3. Illustration

```
*/multi_agent
multi_agent, multiagent, multi_agent_systems, multi_agent_system
*/support_vector_machines
support_vector_machines, support_vector_machine, svm, svms
*/semantic_web
semantic_web, semantic_web_technologies, rdf, ontologies
*/multi_agent/mobile_robots
mobile_robots, robots, mobile_robot, robot, robotic
*/multi_agent/classical_planning
classical_planning, nondeterministic, logic_programs, propositional
*/multi_agent/logistics
logistics, manufacturing, transportation, supply_chain, production
*/support_vector_machines/neural_network
neural_network, neural_networks, artificial_neural_network, neural
```

**Fig. 2.6.** Format of TaxoGen output, where each topic comes with five alternatives and topic-subtopic is separated by slash (DBLP dataset)

```
rib_steak sirloin_steak tenderloin porterhouse flank_steak sirloin
rib_eye cooked_medium ribeye rib med_rare cooked_medium_rare
medium_rare filet ribeye_steak striploin bone boneless filet_mignon
bone_in_rib_eye flank fillet 6oz med prime_rib rare_steak

tomatos cucumbers green_peppers tomatoes romaine_lettuce diced peppers
jalape_os lettuce banana_peppers bell_peppers red_peppers onions
bell_pepper jalapenos shredded_lettuce shredded red_onion green_onions
diced_tomatoes iceberg avocados grilled_onions spring_mix basil_leaves

mash_potatoes peppercorn_sauce chilaquiles cornbread chimichurri
smothered shepherds_pie meatloaf grits red_beans_and_rice biscuits
mash grandmas baked_beans chile_verde po_boy bangers hush_puppies
frites gravy meat_loaf hushpuppies potato short_rib latkes yuca

sauteed_mushrooms fingerling_potatoes wild_mushrooms foie_gras
truffle truffles risotto agnolotti fois_gras gnocchi caramelized_onions
ravioli tagliatelle earthy aged budino fig truffled sherry burrata
mushroom rich marmalade manchego steak_tartare pecorino decadent
```

**Fig. 2.7.** Sample CoRel subtopic clusters for topic "steak" (Yelp Dataset)

```
computer_vision (eid=4061)
  object_recognition (eid=12014)
  object_detection (eid=20)
  speech_recognition (eid=9346)
  deep_learning (eid=1149)
  visual_recognition (eid=7899)
  ...
image_processing (eid=2952)
  image_restoration (eid=11812)
  image_analysis (eid=5249)
  knowledge (eid=717)
  daily (eid=1024)
  ontology (eid=2329)
  ...
datum_mining (eid=152)
  mining (eid=1481)
  pattern_mining (eid=11352)
  association_rule (eid=8105)
  sequential_pattern (eid=35385)
  frequent_pattern (eid=13284)
  ...
...
```

**Fig. 2.8.** Sample HiExpan output (arXiv dataset)

```
*/math
geometric_topology, spectral_sequence, quantum_group, symplectic_geometry
*/physics
photons, electrons, bunch, accelerator_physics, laser, ion, photonic
*/wireless_networks
two_hop, relaying, arq, backhaul, single_hop, multicast, power_allocation
*/classical
uniform_sampling, mle, probabilistic_inference, pomdps, streaming_model
*/math/algebraic_geometry
finite_difference, numerical_solution, multigrid, finite_element, galerkin
*/physics/accelerator_physics
readout, pixel, asic, tpc, tdc, prototypes, gems, gem, alice, rpc
*/wireless_networks/best_effort
traffic_engineering, control_plane, congestion_control, core_network
*/classical/np_complete
np_complete, csp, bipartite_graph, graph_coloring, undirected_graphs
```

**Fig. 2.9.** Sample TaxoCom output with similar format as TaxoGen (arXiv dataset)

## 2.4. Final Remarks

Current ATC approaches remain highly experimental. Although we had success in reproducing the published results, producing quality taxonomies in other domains using automatic taxonomy constructors is difficult. As ATC systems often entail various moving parts, the quality of auto-generated taxonomies can depend on numerous factors, e.g. the quality of keywords, the discriminative power of word embeddings, the clustering algorithms and the seeded input.

We find that ATC taxonomies are particularly vulnerable to lexical noise, mixed granularity and edge inaccuracy, possibly due to the fact that word embeddings alone can omit categorial lexical information critical to taxonomy construction such as hypernymy relations, as well as granularity and topical significance. The usefulness of a generated taxonomy also cannot be guaranteed, due to a mismatch between a user's interests and the actual topics induced by ATC. Further, the taxonomy construction process is complex and time-consuming as it may involve term extraction, real-time local embedding generation and clustering.

We summarized in this chapter pattern-based and statistical-based taxonomy generation strategies, where pattern-based methods using lexical-syntactic patterns may suffer from *low coverage* despite having high precision, whereas statistical methods may have higher recall and can generate a complete taxonomy without supervision but at the same time suffer from data sparsity and lowered precision, when the discriminative power of word embeddings is inadequate to separate noise from useful topics.

Theoretically, ATC can help in the understanding of highly focused and rapidly changing corpus domains when compared to manual taxonomy construction; however, text corpora that can characterize the domains are hard to find and often do not contain sufficient samples of new topics detectable by current ATC methods [**56**]. Besides, ATC taxonomies lack flexibility in that topics and concepts must come from the corpus. In future research, we will study ATC methods that can also manage and integrate abstract topic descriptors into a meaningful taxonomy.

# Chapter 3

# Towards Automatic Taxonomy Evaluation

Many researchers reinstate the importance of having clear objectives for taxonomy and ontology creation, as well as developing a set of evaluation methodologies for assessing the structure, word relations and cluster coherence of an automatically generated topic taxonomy. Because quantitative evaluation methods are lacking, research on taxonomy and ontology construction heavily relies on qualitative descriptions for comparing ATC output from the various perspectives of ontology engineers, system users or domain experts [**30, 35**]. Quantitative measures, on the other hand, require either calculating intrinsic measures such as perplexity and cluster distance, or depend on extrinsic measures such as comparing target taxonomies to a reference, also known as a *gold-standard*.

Because an ontology or taxonomy is frequently used within a targeted application to achieve a specific goal, many researchers believe that a task-independent automatic evaluation remains elusive, citing the lack of a general solution [**78**]. Current studies on taxonomy and ontology evaluation, to our knowledge, are primarily concerned with the assessment of web ontologies. Nevertheless, we find that ontology evaluation strategies shall remain mostly applicable to ATC taxonomies, despite the fact that web ontologies are far more structurally complex. In the next sections we will introduce these evaluation strategies and discuss approaches that may ultimately lead to quantifiable and reproducible taxonomy scoring.

## 3.1. Taxonomy Errors

Automatically constructed taxonomies are often exposed to basic errors that are not commonly found in manually constructed taxonomies, from selecting terms that best characterize a topic cluster to assigning a node to its ideal position. In Semantic Web applications, Vrandečić in *Ontology Evaluation* argued that mistakes and omissions in ontologies can lead to the inability of applications to achieve the full potential of exchanged data [**104**]. Below we use a machine generated aviation taxonomy to summarize the various types of errors identified in our research.

**Noisy and trivial topics** lack significance and reduce the taxonomy's usefulness and representativeness of the knowledge domain. This is the most common error in an automatically generated taxonomy as word embeddings struggle to capture term significance. In figure 3.1, topics such as "25r", "27r", "7l" and "9l" (taxi path names) are included in a subset of an auto-generated aviation incident taxonomy concerning taxiway issues. A simple improvement can be implementing filters or Regex syntaxes for removing such lexical patterns, however this approach can be laborious and may not eliminate all potential noise.

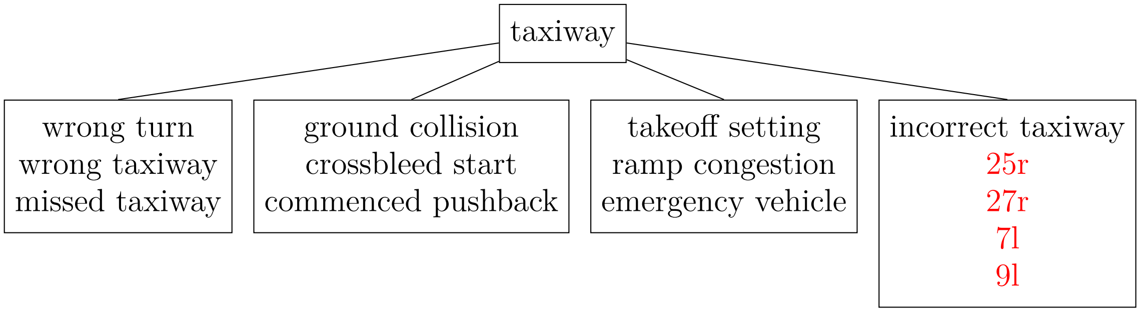


**Fig. 3.1.** Noisy labels in an auto-generated taxonomy

Some ATC algorithms also heuristically select or favour terms that appear with the highest frequency or are more "unique" to one topic cluster than others in order to represent the topic of a word cluster, producing incorrect or noisy cluster labels. For example, figure 3.2 showcases a taxonomy subset where the parent node is labelled simply "e" for taxa "traffic", "thrust" and "altitude". In other instances we have observed erroneous parent labels such as "white", "red" and "bit".

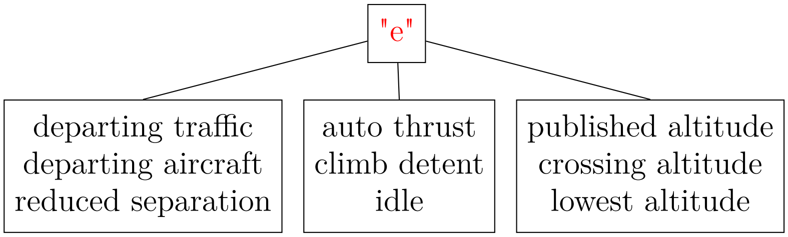


**Fig. 3.2.** Erroneous parent label

**Incoherent clusters** decrease the uniformity, interpretability and accuracy of the taxonomy. In figure 3.3, "incorrect taxiway" is a much closer topic to "wrong taxiway" (in fact, they are semantically identical) as compared to "missed taxiway".

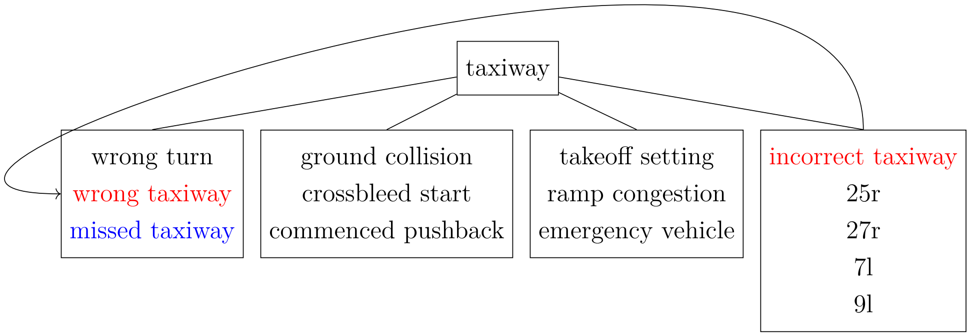


**Fig. 3.3.** Cluster coherence problem

**Structural problems** are highly complex and harder to define since there is no clear standard that indicates the optimality of a taxonomy's structure (number of nodes and edges, depth, granularity) in a knowledge domain. Figure 3.4 depicts a simple case of this problem, in which the sibling nodes have different granularities and should be clearly displaced at different levels. Another way to consider the structural error is whether nodes at a certain ontology level contain a subsumption relation, e.g. if incursion and runway/taxiway, or turbulence and weather should appear at the same ontology level as shown in figure 3.4.

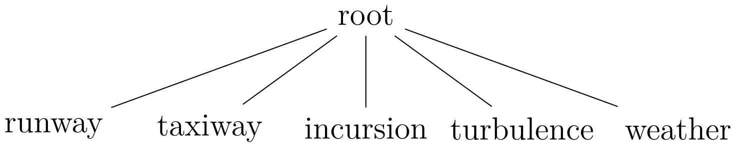


**Fig. 3.4.** Taxonomy structure problem

Other taxonomy errors can occur before or after the ATC process. For instance, **source errors** can occur when an ontology extraction process learns from incorrect data, e.g. a person born in 1802 but featured in a Wikipedia category for people born in 1805. **Philosophical issues** can be exemplified with the following question: "is an economist working in France a French Economist, even if he was born in Ireland?"

## 3.2. Concept-pair and Coverage Analysis

Our first goal in this master's project was evaluating a multi-level large aviation incident ontology that is comprised of specialized aviation terminologies. Foremost, we intended to validate the topic relations that exist in the ontology. As ground-truth aviation incident ontologies are rarely found, we instead sought to find evidence for the concept-pair coexistence in text. Specifically, in a multi-level ontology and a massive aviation incident corpus, we attempted to extract all topic-subtopic identifier pairs in the adjacent ontology levels, e.g. *weather* as the topic and *hail, lighting strike, heavy rain* as the subtopics:

```
        weather: {hail, lighting strike, heavy rain}
                                   ⇓
(weather, hail), (weather, lighting strike), (weather, heavy rain)
```

Next, we aimed to retrieve all reports that contain an event pair in the ontology and use the contexts as evidences for topic relation evaluation. For example, one such report could be *"bad weather... heavy rain... return to gate"*. If no reports mentioning the two events can be found, the ontology *edge* may also be redundant or obsolete and should be removed.

The problem of this approach is that the names of valid events may not appear *as-is* in the corpus documents, as abstract event descriptions and umbrella terms are ubiquitous in ontologies created for manual classification, e.g. "Dangerous Goods Carried by Passenger and Crew" and "Threat & Interference". Another challenge arises when different event descriptions can refer to the same concept (e.g. "birdstrike" & "bird strike"; "intoxication" & "intoxicated"), which may be resolved using text preprocessing and approximate string matching. Our preliminary results show low recall with less than 50% of concept pairs found to coexist in all returned reports and as such, assuming topic coexistence in corpus as a proof of positive ontology relations can result in *false negatives*.

For coverage analysis and estimating the number of missing topics in an ontology, domain experts and external references are commonly required. However, Brewster et al. [**11, 13**] proposed to extract domain-specific terms from the document corpus using latent semantic analysis (LSA) and measure the overlap of the LSA extraction and the ontology topics. In sum, it is difficult to evaluate ontology concepts especially those from professional domains.

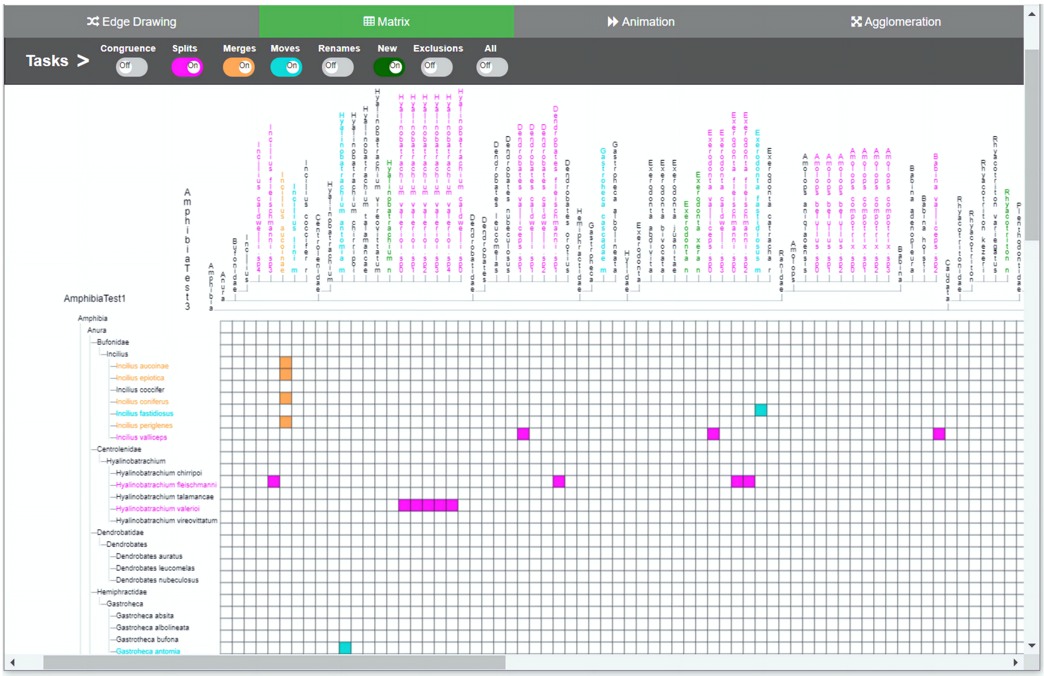
## 3.3. Comparing Taxonomies

Studies have been conducted to methodically describe, compare and visualize the differences and similarities between two taxonomy artifacts, or to highlight changes between two different versions of a taxonomy. In the biological taxonomy domain, such comparisons are critical for the "reconciliation of alternative versions of a taxonomic classification" [**85**] and are mostly carried out by domain experts with the help of visualization tools due to the complexities of taxonomy's structures, taxa and usage.

In biology, for example, characterization of a taxonomy and its different versions involves multi-national experts, expert interviews, literature reviews, as well as multiple sets of manual tasks done by taxonomists from identifying splits and merges to locating repeated names. Few researchers detail the procedures for systematically comparing two taxonomies without some sort of expert agreement, with the exceptions of some methods that compare labelled trees and tree alignment, compute "tree edit distance" and measure taxonomy matching rates [**8, 117, 2**]. Consequently, the current methodology of taxonomy evaluation is largely confined by expert discretion. Even if external evaluations can help determining which taxonomy is more useful for a specific task, the results may not correlate at all to the actual observed quality of many taxonomies.

Below we list four popular visualization software comparing two taxonomy-like hierarchies in the literature, including *edge drawing*, *matrix representation*, *animation* and *agglomeration* [**85, 31**]:

- **Edge drawing** uses colour-coded edges to visualize changes between two versions of a taxonomy;
- **Matrix representation** arranges two taxonomies along the horizontal and vertical axes of a matrix, similar to a Levenshtein distance matrix. Differences and changes are coloured and highlighted at the cross-sections of the matrix cells;
- **Animation** uses animations to illustrate the differences between two taxonomies;
- **Agglomeration** merges two taxonomies into a single hierarchical structure (e.g. sub-trees with the same parent name will be merged together) and uses colours to distinguish parts from different taxonomies and changes.



Tasks >

Congruence: Off
Splits: On
Merges: On
Moves: On
Renames: Off
New: On
Exclusions: Off
All: Off

**Fig. 3.5.** Matrix visualization of taxonomy differences (Sancho-Chavarria et al. 2020)

Finally, the advancement of neural language models enables visualization of the relatedness of words in a corpus and manually identifying overlapping and duplicating topics using word embeddings. T-SNE [**101**] and PCA (Principle Component Analysis) are two principle methods that can be deployed to project an n-dimensional word embedding (for example, Word2Vec embeddings typically have $n = 300$ and for BERT $n = 768$) to a two-dimensional or three-dimensional space through *dimensionality reduction* while preserving data variation. During topic visualization, one can observe that highly correlated topics are clustered closely as their embeddings share closer distances. Supposedly, the methods can be used directly for projecting the topics of any taxonomy or ontology onto a 3D topic space, if their surface-forms are found in-corpus, but this is rarely the case and would require domain experts to make sense of the projected topic clusters.

## 3.4. Intrinsic Evaluation

In their simplest form, ATC-generated topic taxonomies can also be viewed as a collection of topic clusters calculated using word vectors and clustering algorithms. External evaluation for cluster quality is rarely used due to the lack of dependent variables or class labels for each cluster object. For evaluating clustering results with reference labels, simple measures such as *homogeneity score*, *completeness score* and *v-measure* can be used.

As a result, *intrinsic* metrics may have become indispensable for obtaining the optimal topic clustering for some ATC systems. However, such metrics cannot be reused for evaluating ATC outputs as the clusters are directly optimized using these methods and the scores are difficult to interpret by humans. Below we list three widely-used cluster evaluation methods for ATC clusters.

### Davies-Bouldin Index

The Davies-Bouldin index (DBI) is an intrinsic metric based on the principle of, in short, *within-cluster* and *between cluster* distances, or the proximities between clusters and their nearest neighbours [**61, 19**].

$$DB = \frac{1}{k}\sum_{i=1}^{k}\max_{j\neq i} R_{ij} \tag{3.4.1}$$

With:

$$R_{ij} = \frac{s_i + s_j}{d_{ij}} \tag{3.4.2}$$

In Equation 3.4.2, $s_i$ denotes the average distance between each cluster data point and the cluster centroid and $d_{ij}$ is the distance between cluster centroids $i$ and $j$. As the clusters become more compact and separated, their Davies-Bouldin index will decrease. In some ATC research, Davies-Bouldin is used to quantify cluster quality [**115**].

### Silhouette Score

A higher silhouette score [**84**] indicates how well an object belongs to its own cluster as compared to all other clusters. Given variables $a$ and $b$, where:

- **a** is the mean intra-cluster distance measured by the mean distance between each data point and all other data points in the *same* cluster.
- **b** denotes the mean nearest-cluster distance, which is measured by the mean distance between a data point and all other points in the *next nearest* cluster.

For each datapoint, the Silhouette Coefficient can be computed as:

$$s(i) = \frac{b(i) - a(i)}{\max a(i), b(i)} \tag{3.4.3}$$

The silhouette score can be a good indicator for cluster quality without target variables and the score is bound between -1 and +1, with closer to 1 indicating strong affiliation, 0 indicating overlapping clusters and -1 suggesting an object could be incorrectly assigned, which can be superior to that of Davies-Bouldin as its scores are unbounded and less interpretable.

### Calinski-Harabasz Index

Also known as the variance ratio criterion as it is based on the principle of variance, Calinski-Harabasz index [**14**] is a heuristic device defined as ratio of the *between-cluster dispersion* and *within-cluster dispersion.* It is fast to compute, and the denser and more well separated the clusters, the higher the score. However it generally favours *convex clusters* over *density based clusters.*

$$S_{CH} = \frac{B}{W} \times \frac{N-k}{k-1} \tag{3.4.4}$$

In which $B$, $W$, $N$, $k$ denote the between-cluster variance, within-cluster variance, total number of datapoints and the number of clusters.

## 3.5. Extrinsic Evaluation

Brank et al. [**11**] summarized four principle *extrinsic* ontology evaluation methods, by (1) comparing the target ontology to a "gold standard" (ground-truth) ontology [**60**]; (2) using the target ontology in an application and evaluating the application results [**78**]; (3) conducting coverage analysis comparing the target with a source of data (eg., a collection of documents) about a specific domain [**13**]; (4) manual reviews done by human experts that assess how well the target ontology meets a set of predefined criteria, standards, and requirements [**57**].

### 3.5.1. Gold Standard Method

Studies of gold standard methods focus on comparing and measuring the similarity of two ontologies.[1] Maedche and Staab [**60**] proposed a two-level comparison framework to measure ontology similarities on lexical and conceptual levels. Lexical level comparison aims at measuring the similarity between two target strings, e.g. "TopHotel" and "Top_Hotel" and producing a string matching score using a modified Levenshtein's edit distance. Conceptual level comparison focuses on the similarity of concept by computing the *semantic cotopy*, which given a single term from two ontologies, compares the *intersection* or *overlapping* of all the super- and sub-concepts of the term in each ontology to estimate the conceptual similarity of the term's role in the two ontologies. The evaluation for Maedche and Staab's method has shown promising results. Brewster et al. [**13**], however, argued that it is still difficult to construct automated tests to automatically compare two or more taxonomies, as it depends on external semantics to perform the evaluation capable only by human beings, although recent work has been investigating in contextualized word embeddings as a proxy for human judgements of word semantics [**118**].

Nevertheless, the gold standard method has remained popular as given a collection of is-a and not-is-a term pairs as ground truth, standard relation classification metrics such as precision (P), recall (R) and f-score (F) can be employed for quantitative evaluation [**92, 114**]. Further, the gold-standard evaluation can also be helpful for parameter tuning [**100**]. The generic precision and recall metrics are defined as below:

$$precision = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{3.5.1}$$

$$recall = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{3.5.2}$$

For a gold-standard evaluation, a precision score can be calculated as the percentage of taxonomy nodes that also appear in the gold standard, divided by the total number of nodes in the taxonomy; a recall score is defined as the percentage of topic nodes in the gold standard that also appear in the taxonomy, in relation to the total number of entries in the gold standard [**11**]. SemEval-2015 [**9**] further detailed the definitions of several quantifiable gold-standard metrics by comparing:

$$common\ nodes = |V_S \cap V_G| \tag{3.5.3}$$

[1] As ontologies are in fact far more structurally complex than taxonomies, we draw comparisons solely on the evaluation of taxonomic relations in an ontology and forgo evaluations of all other ontology components and functionalities such as formal language, syntax, XML and RDF.

$$\textit{vertex coverage} = \frac{|V_S \cap V_G|}{|V_G|} \tag{3.5.4}$$

$$\textit{common edge} = |E_S \cap E_G| \tag{3.5.5}$$

$$\textit{edge coverage} = \frac{|E_S \cap E_G|}{|E_G|} \tag{3.5.6}$$

$$\textit{edge precision (P)} = \frac{|E_S \cap E_G|}{|E_S|} \tag{3.5.7}$$

$$\textit{edge recall (R)} = \frac{|E_S \cap E_G|}{|E_G|} \tag{3.5.8}$$

$$\textit{F1 score} = 2\frac{(P * R)}{(P + R)} \tag{3.5.9}$$

Constructing a gold standard is time-consuming and typically involves: (1) selecting fine-grained concepts using a domain reference, e.g. Wikipedia or ACM classification that categorizes subjects in Computer Science; (2) choosing a depth or the number of classification levels of the taxonomy for a desired granularity; (3) choosing a scope of coverage based on the subject of interest. However, even the optimality of manually constructed gold-standards cannot be guaranteed.

Lastly, much semantic work including topic modelling and ontology comparison depends on WordNet for gold-standard evaluation. Many similarity metrics based on WordNet are therefore proposed for estimating and comparing topic similarities using the depths and subsumer information of WordNet. The *least common subsumer* (LCS) is defined as the deepest node in the hierarchy that subsumes both of the synsets under question [**67**], i.e. the most specific concept which is an ancestor of both WordNet concept A and concept B [**73**]. In the case of a subset of WordNet in figure 3.6, the LCS of "lion" and "cat" is "feline". It is also safe to assume from the example that a lion is closer to a cat than it is to a wolf, as the LCS of "lion" and "cat" is "feline" and the LCS of "lion" and "wolf" is "carnivore", which is a further concept.

Similarly, the similarity of two ATC concepts can also be measured if they can be identified in WordNet or other structured databases. Below we include three widely used WordNet semantic similarity metrics including Wu-Palmer (WuP), Leacock-Chodorow($LC_H$) and Lin.

Wu-Palmer [**110**] aimed to tackle a task of English-Chinese machine translation by accurately selecting fine-grained lexemes from WordNet or a comparable thesaurus. The following metric 3.5.10 quantifies the concept similarity in a hierarchical structure:

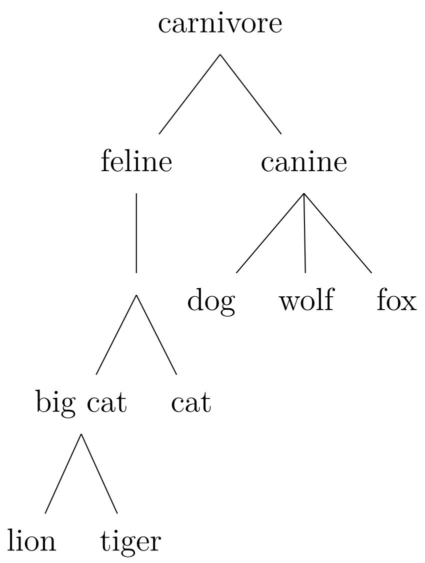


**Fig. 3.6.** Subset of WordNet Synset

$$WuP(c_1, c_2) = \frac{2 \cdot depth(LCS_{c_1,c_2})}{depth_{c_1} + depth_{c_2} + 2 \cdot depth(LCS_{c_1,c_2})} \tag{3.5.10}$$

Leacock-Chodorow ($LC_H$) [**51**] attempted word sense identification, specifically to disambiguate a noun, a verb and an adjective through knowledge acquisition via WordNet. The $LC_H$ measure finds the shortest path $sp(c_1, c_2)$ between two WordNet synsets and scales by the maximum depth of WordNet ($\mathbf{D}$) [**67**]. The final score takes the log likelihood as:

$$LC_H(c_1, c_2) = -\log \frac{sp(c_1, c_2)}{2 \cdot \mathbf{D}} \tag{3.5.11}$$

Lin [**55**] uses an *information theoretic* approach introduced as the Resnik Information Content [**80**], which weights the edges of WordNet using their frequency of occurence in a text corpus to measure the Information Content of WordNet concept nodes, avoiding the assumption that all edges in WordNet possess equal importance [**67**]. As such, the Information Content is defined by $IC(c) = -\log p(c)$ and the Lin measure can be calculated as:

$$Lin_{c_1,c_2} = \frac{2 \times \log p(LCS_{c_1,c_2})}{\log p(c_1) + \log p(c_2)} \tag{3.5.12}$$

### 3.5.2. Data-driven method

Brewster et al. [**13**] presented the namesake approach that provides a more automatized evaluation strategy to the more approachable and more popular gold-standard method. Fundamentally, it proposes an evaluation task to select the most structural *fit* ontology among a set of candidates to a target corpus, or to select an ontology that best represents the knowledge domain of the text corpus. Theoretically, the data-driven approach attempts to derive the conditional probabilities of ontologies given a corpus: the ontology that maximizes the conditional probability of the ontology $\mathcal{O}$ given a corpus $\mathcal{C}$ is hence deemed the best fit ontology $\mathcal{O}^*$, where *Bayes' Theorem* is used to derive the fit score *if* a valid method would exist for estimating $P(C|O)$.

$$O^* = \arg\max_o P(O|C) = \arg\max_o P(C|O)P(O)/P(C) \tag{3.5.13}$$

However, the authors have expressed uncertainty about how to approximate such conditional probability and no experiment results were shown in the research. One solution to this is to extract all domain-specific terms using an external reference, e.g. performing a two step WordNet hyponym lookup, then measure the overlap between the extracted terms and the target ontology so as to estimate the "fit" or the coverage of the ontology. An ontology-tagged corpus by for example WordNet may also help estimate such *coverage.* Although the data-driven approach shows great potentials, it is fairly complex to implement. A detailed pipeline for converge analysis is shown as follows:

(1) **Keyword extraction**. Identify and extract keywords and terms of interest from a selected corpus;

(2) **Query expansion**. Use existing ontologies such as WordNet and other Information Retrieval techniques to find and expand the representations of the terms from (1). The authors suggest to add two levels of hypernyms to each term using WordNet, to ensure that a term in an ontology can be located via multiple lexical variations to facilitate concept mapping;

(3) **Ontology mapping**. Match the ontology terms and their lexical variance to the keywords extracted directly from the domain corpus to obtain a rough estimate of the *coverage* of the ontology on the textual data.

### 3.5.3. Application-based method

The idea behind application-based ontology evaluation is straightforward in that one might infer the quality of some ontologies by plugging them into an application that would benefit from the use of taxonomies and comparing the application results (performance) [**11**]. However, this approach does not necessarily exclude the use of gold-standards but instead concentrates on producing a quantifiable score for ontology comparisons. Porzel et al. [**78**] who pioneered the application-based approach proposed several possible applications for inferring ontology quality such as "speech recognition concept tagging" and concept-pair relation classification. Large ontological applications such as ONTOSCORE that converts an ontology to a directed graph can also be used for scoring. Specifically, the authors sought to compute three types of performance measures typical in automatic speech recognition, the target application and compare the results to a gold-standard to obtain *error rates*:

(1) **insertion errors** indicate *superfluous concepts*, isa- and semantic relations;
(2) **deletion errors** indicate *missing concepts*, isa- and semantic relations, and;
(3) **substitution errors** indicate off-target or *ambiguous concepts*, isa- and semantic relations.

Another objective of the proposed application-based evaluation can be improving an existing ontology by reducing the errors above. As a result, the application-based approach is a viable alternative to gold-standard evaluation but is complex and non-scalable, best suited for comparing a set of taxonomies with an existing gold-standard on some particular usage of an ontology application. Brank et al. [**11**] summarized that it is in fact hard to correlate ontology quality with the application performance and to measure the extent of an ontology's contribution to the application outcome. It might also be infeasible to plug in all ontologies into the sample application for evaluation.

### 3.5.4. Manual Evaluation

In the end, manual evaluation is still desired due to the lack of gold-standards. It can also be the only option when a taxonomy is designed following certain strict design principles. It is difficult to create comprehensive machine-evaluation protocols that can cover the entire manual evaluation spectrums, including lexical vocabulary and concept, hierarchy structure, context and application, syntactic, structure and design and other semantic relations [**11**]. Manual evaluation can also revolve around *philosophical* evaluation criteria that can guide ontology design (after all, ontologies and taxonomies are created for and used by humans) some of which are listed below [**95, 34, 1, 104**]:

(1) **Accuracy** measures the precision and recall of which the asserted knowledge in the ontology agree with an expert's knowledge;
(2) **Adaptability** measures how "easy or difficult" to use an ontology given different contexts or applications;
(3) **Clarity** measures how effectively the ontology communicates the intended meaning of the defined terms;
(4) **Completeness** measures if the domain of interest properly covered via coverage analysis;
(5) **Conciseness** measures if the ontology includes irrelevant elements regarding the domain to be covered;
(6) **Consistency** measures the number of terms with contradicting meanings.

In sum, a manual approach can shed valuable insights about a taxonomy but is still not advisable as human evaluators inevitably judge from different standpoints, creating discrepancies during evaluation. To replicate the success of manual construction and evaluation, however, ATC and ATE must possess a vast amount of real-world knowledge or be able to accurately derive facts from domain corpora.

### 3.5.5. Rank-based Metrics

Lastly, we mention ranked-based metrics commonly used for evaluating recommender systems, whose aims are to provide a list of most relevant recommendations or user-interested items. These criteria measure a system's ability to retrieve relevant information, given preferred rankings of a list of items that should appear in the system output. In hypernymy discovery, external rank-based metrics are used to measure the effectiveness of a system at retrieving all relevant hypernyms of a concept. These measures include mean reciprocal rank (MRR), precision@K (P@K), recall@K (R@K) and mean average precision (MAP), which can be deployed with human-created references/gold standards.

To compute MRR, we first generate for each user $u$ a list of recommendations and find **rank** $k_u$ of its first recommendation (the first recommendation has rank 1), then the reciprocal rank is:

$$\mathrm{MRR}(O, U) = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{k_u} \tag{3.5.14}$$

For P@K and R@K, we first set a rank threshold $K \in 1, 2,.., n$, and the scores are therefore:

$$\text{P@K} = \frac{\text{number of relevant items}}{\text{number of items recommended}} \tag{3.5.15}$$

$$\text{R@K} = \frac{\text{number of relevant items}}{\text{number of all possible relevant items}} \tag{3.5.16}$$

With P@K we can compute mean average precision using the procedure as follows:

(1) Consider rank position of each relevant document $K_1, K_2, ..., K_R$;

(2) Compute P@K for each $K_1, K_2, ..., K_R$;

(3) Average precision is the mean of all P@K for a single ranking;

(4) Mean Average precision is the mean of all P@K for multiple rankings.

## 3.6. Towards Automatic Evaluation: a Dialogue

*"Which system works the best? Which taxonomy is the best?"* In sections 3.4 and 3.5 we discussed a wide gamut of intrinsic and extrinsic methods for taxonomy and ontology

evaluation, which suffer the trade-off between reproducibility (for extrinsic measures involving manual inputs) and interpretability (for intrinsic measures). The data-driven and application-based methods have demonstrated the most potential towards automatic taxonomy evaluation among the four principle ontology evaluation methods, but are difficult to implement because taxonomy coverage is hard to estimate and many ontologies do not have a specific application. Hence for many researchers, a general solution for task-independent automatic evaluation does not exist [**78**].

## "Which system works the best?"

As our first problem in this thesis concerns the ranking of ATC systems, it is possible to design universal benchmarks similar to that of GLUE for Natural Language Understanding (NLU) [**105**], through gathering domain-focused corpora from various sources and hand-crafting gold-standard taxonomies, to compare and measure ATC system's ability to find common edges and nodes. We can also simply test the systems on reconstructing or expanding a subset of WordNet using WordNet annotated corpora such as SemCor3.0 for English. It is unclear, however, whether reconstructing WordNet is a good indicator of a system's performance and whether the same performance can be translated to other domains. Benchmark-making is also a lengthy process that is beyond the scope of this thesis. As such, we decided to concentrate on our second challenge, which is taxonomy scoring and ranking.

## "Which taxonomy is the best?"

Although automatic taxonomy and ontology evaluation has gained more traction recently [**72, 103, 12, 20, 18**], a gold standard from an authoritative source is still much of a necessity for calculating node and edge precision and recall. In the namesake research of automatic ontology evaluation [**12, 103**], however, ontology researchers began to develop methods for automatically evaluating a few selected aspects of web ontologies such as *rigidity, unity, identity and dependence* or "similarity measures on partitions". De Knijff et al. [**20**] proposed an auto-evaluation measure based on a score that is a trade-off between a term frequency measure, but the measure is also used for calculating the depth of an "optimal" taxonomy during taxonomy construction, which led to some researchers arguing that the score is difficult to interpret [**18**]. Patel et al. [**72**] used the names of concepts and relations of a collection of web ontologies as direct input to text classification models with the goal of identifying ontology topics. Nevertheless, the study only measured the accuracy of topic classification rather than using it towards ontology ranking.

## Automatic Topic Coherence Evaluation

Newman et al. [**67**] described a novel intrinsic measure for *topic coherence* scoring, which we find potentially advantageous for quantifying the quality of a taxonomy cluster, using a symmetric word-similarity measure $\mathcal{D}(w_i, w_j)$ to calculate the arithmetic mean or median score for component words $(w_1, \ldots, w_10)$ under a given topic:

$$\text{Mean-D-Score}(\mathbf{w}) = \text{mean}\{\mathcal{D}(w_i, w_j), ij \in 1 \ldots 10, i < j\} \tag{3.6.1}$$

$$\text{Median-D-Score}(\mathbf{w}) = \text{median}\{\mathcal{D}(w_i, w_j), ij \in 1 \ldots 10, i < j\} \tag{3.6.2}$$

To calculate $\mathcal{D}(w_i, w_j)$, the authors proposed three scoring methods including WordNet similarity (Leacock-Chodorow, Wu-Palmer, Lin etc.) as introduced in section 3.5.1, Wikipedia similarity and search engine-based similarity. The search engine-based similarity include "Google title matches" and "Google log hit matches" and works by querying the cluster topics, e.g. *"space earth moon science scientist light nasa mission planet mars ..."*, and matching their occurrences in the top results returned by the search engine. The research shows that both Wikipedia and search engine-based approaches outperform WordNet-based methods and are close to the results from manual annotations. We believe however that search engine-based similarity, due to its broader coverage, may better suit our needs for universal taxonomy evaluation.

## Machine Learning as External Evaluators

Finally, we draw attention to the success of machine learning in non-classification tasks such as text readability assessment and essay scoring [**26, 98**]. The key to the unparalleled success lies in the engineering of hand-coded lexical, syntactic and semantic features, e.g. average sentence lengths, mean syllables per word, word frequencies and part-of-speech (POS), with some novel studies propose to learn such features automatically. To train the machine evaluators, human annotators will provide reference readability or essay scores for the evaluation models to reproduce human gradings.

In the study of *text generation*, which aims to produce texts in natural languages using text generation models or *casual language models*, with important applications including machine translation and paraphrasing, evaluation also entails comparing the generated texts to some annotated references for quantifying their semantic similarities. As exact-matching evaluation methods such as the widely-adopted BLEU metric [**71**] can only compare the surface-form similarity of two sentences through token matching, Zhang et al. [**118**] proposed a novel BERTScore that can better measure sentence semantic similarity using contextualized BERT embeddings. The results of BERTScore show that neural-based evaluation can in fact correlate better with human judgements than most lexical similarity methods, broadening the horizon of using language models for automatic semantic evaluation.

Inspired by the aforementioned studies, we propose in chapter 4 and 5 two new evaluation proxies towards automatic ATC taxonomy assessments, leveraging supervised classification and domain-adapted language models for producing reproducible and interpretable results.

# Chapter 4

# Supervised Taxonomy Evaluation with Text Classification

We explore in this chapter an application-based (or rather, a mixed data-driven and application-based) ontology evaluation approach, using text classification as a supervised machine learning task for quantifying and comparing the quality of taxonomies generated from labelled datasets, e.g. reviews annotated by genre and news articles categorized by topic. Unlike previous methods that use the names of ontology nodes and edges directly for classifying ontology topics [**72**], we propose to extract feature vectors from labelled documents by "tagging" a document with taxonomy entities and obtain a simplistic view of the observation, e.g. representing the sentence *"massive protests erupted in China"* with taxonomy words "protest" and "China". The manually assigned topic labels, e.g. "politics", usually coming from a ground-truth, are then used as target variables for training an auto-taxonomy evaluator, implying the true class distribution of the observation. An ML algorithm is used to learn the ideal taxonomy-transformed document representation and attempt to map it against the topic labels using a training set, and the learning accuracy is obtained from a testing set to complete the supervised learning task. We will dive into the details as we progress through the chapter.

Since we assume a strong correlation can be found between document classes and topic clusters of a quality taxonomy, our method can be regarded as a measurement of cluster coherence or at least as an automatic coverage analysis. We hypothesize that a topic taxonomy with better structure, higher cluster coherence and topic coverage will outperform its "noisier" counterparts in topic classification because:

(1) documents represented by noisy taxonomy topics will not be consistent with the distribution of their manual labels;

(2) trivial taxonomy topics with low *entropy* are not indicative as classification features for the document topics and the corpus domain, therefore do not increase classification accuracy;

(3) documents tagged by a taxonomy with less coverage may lead to fewer features or can produce a sparser feature vector/matrix, resulting in lower recall.

To demonstrate the intuition behind our proposed approach, table 4.1 depicts five sample news articles and taxonomy topics extracted from the AG News corpus, as well as the articles' class labels. The taxonomy topics found in the documents display a high correlation with the corresponding document label and a lower correlation with other document topics. Although word sense disambiguation should be applied here to better determine the sense of a taxonomy concept and to improve concept mapping accuracy (e.g. computer virus and viral disease), we assume the percentage of ambiguous topics is low and identical across all taxonomies generated on the same corpus.

| | Document | Taxonomy topics in document | Doc. label |
|---|---|---|---|
| 1 | dutch retailer beat apple download... | "market", "retail", "music", "free" | technology |
| 2 | afghan army dispatch calm violence.. | "capital", "government" | politics |
| 3 | dollar briefly hit wk low v euro ... | "economy", "euro", "oil" | business |
| 4 | natalie coughlin win backstroke ap... | "olympic", "gold" | sports |
| 5 | china red flag linux focus enterprise | "company", "linux", "software" | technology |

**Table 4.1.** Document corpus annotated with entities from a quality ATC taxonomy, which strongly correlate with the manually created document labels. Keywords appearing in both the documents and the taxonomy are highlighted in yellow.

Thus, our problems are twofold: (1) how to acquire feature vectors to best represent the semantics and the structure of a taxonomy evaluation target, and (2) whether the classification scores will correlate with the observed taxonomy quality, a common pitfall in application-based evaluation. For the first problem, we experiment in section 4.2 with a bag-of-words (BoW) approach by associating each labelled document with relevant taxonomy entities via string matching and then using entity counts as features to train a text classifier; For the second problem, we design a *perturbation test* in section 4.4 to simulate the degradation of taxonomy quality by reassigning a percentage of taxonomy topics to different topic clusters via random swapping and random deletion [**107**].

## 4.1. Evaluation Targets

### 4.1.1. Datasets

Our evaluation procedure is based on the assumption that manually created topic labels exist for defining the topics of interest in some documents. To our best knowledge, public datasets for topic classification are uncommon, with most focusing on other NLP tasks such as sentiment analysis. However, topic classification is in high demand on a commercial level and many companies possess private annotated data for topic classification and taxonomy construction. We summarize two datasets for experimentation and illustration below.

(1) **AG news** is a widely-used news classification dataset collecting over a million news articles. The documents are categorized into four topic categories, including world (1), sport (2), business (3) and science/technology (4). Our training and test set contain 120,000 and 7,600 news samples respectively, each containing a short title and a description. Additionally, the dataset has a balanced class distribution across all four categories.[1]

(2) **IATA dataset** is a private large multi-label and multilingual dataset, collecting over millions of aviation incident and accident reports and containing frequent jargon, acronyms, misspellings and noise. For our evaluation, we use ten distinctive level-one categories as ground truths.

| | Title | Description | Class Index |
|---|---|---|---|
| 1 | El Salvador Jail Riot Kills at Least 31 (AP) | AP - Rival inmates fought each other with... | World (1) |
| 2 | Reversal Gives Peirsol Gold; Phelps Win... | Despite being disqualified shortly after his... | Sport (2) |
| 3 | Cash-rich, commodity-starved mainland ... | SINGAPORE: The mainland #39;s plan to... | Business (3) |
| 4 | IBM Extends Chip Performance with Ger... | IBM today said it has demonstrated a tec... | Sci/Tech (4) |

**Table 4.2.** Sample AG News data input

### 4.1.2. Generated Taxonomies

Because it is difficult to include every component of a large taxonomy, we attempt to illustrate five taxonomies (TG1, TG2, HE1, HE2, TC1) generated on the AG News Dataset using three different ATC systems introduced in section 2.3.3: TaxoGen (TG1, TG2), HiExpan (HE1, HE2) and TaxoCom (TC1). We also selected two off-topic taxonomies (TC*, HE*) generated on other datasets to study their effects on proxy classification. In a large taxonomy, we use ellipsis to represent one or more subtopic clusters under the parent topic.

From left to right, table 4.3 compares the maximum vertical levels, number of main topics, number of "neighbours" for the main topics , total number of concepts in the taxonomy, and the average number of children (sans neighbours) per taxonomy parent. We also display the level 1 topics, the percentage of documents in the training and testing samples that corresponded to at least one taxonomy topic, and a manual ranking for the evaluation targets. Specifically, the percentage of matched documents describes the proportion of documents that contain at least one word from a taxonomy, which to some extent indicates taxonomy coverage. We also attempted a manual ranking based on topic quality, taxonomy structure and convergence to the real topic labels (i.e. world, sports, business and science/technology). Due to the primitive nature of ATC, none of the extracted taxonomies could compete with

[1] `https://www.kaggle.com/datasets/amananandrai/ag-news-classification-dataset`

|  | Depth | Nodes in taxo. | Neigh-bours | Total vocab. in taxonomy | Avg. children per parent (no neighbours) | Main topics | Matched documents (%) | Manual Ranking |
|---|---|---|---|---|---|---|---|---|
| TaxoGen1 (TG1) | 3 | 79 | 621 | 700 | 4 | "opposition", "sharply", "field", "interface" ... | 90% | 3 |
| TaxoGen2 (TG2) | 3 | 33 | 244 | 277 | 4 | "interface", "sharply" | 51% | 4 |
| HiExpan1 (HE1) | 2 | 399 | 0 | 399 | 26.8 | "business", "politics", "sports", "technology"... | 84% | 1 |
| HiExpan2 (HE2) | 1 | 19 | 0 | 19 | 19 | "business", "politics", "sports", "technology" | 30% | 5 |
| TaxoCom1 (TC1) | 2 | 38 | 323 | 361 | 6.3 | "sports", "politics", "business", "arts", "case" | 30% | 2 |
| TaxoCom-OT (TC*) | 2 | 42 | 366 | 408 | 7 | "math", "physics", "classical", "databases" | 23% | 7 (tie) |
| HiExpan-OT (HE*) | 2 | 490 | 0 | 490 | 22.9 | "unruly passenger", "emergency equipment" | 54% | 7 (tie) |

**Table 4.3.** Structures of taxonomies for ranking

| Taxo | Main topic | Neighbours (ATC generated alternative titles) |
|---|---|---|
| TG1 | opposition | opposition, parliament, election, elections, parliamentary, political, vote, prime minister, opposition leader, coalition |
| TG1 | interface | interface, desktop, features, server, digital, integrated, tool, version, platform, compatible |
| TG2 | memory | memory, size, cards, supercomputer, chip, breakthrough, processors, drives, flash, memory, dual |
| TG2 | robot | robot, solar, researchers, science, orbit, spacecraft, mouse, tiny, craft, brain |
| TC1 | politics | affordable care act, abortion, immigration, law enforcement, gay rights, president obama, health care, surveillance, advocates, congress |
| TC1 | tennis | rafael nadal, federer, djokovic, roger federer, novak djokovic, serena williams, fed cup, andy murray, nadal, wimbledon |
| TC* | number theory | hardy space, banach spaces, composition operators, functional analysis, linear operators, hilbert transform, boundedness, pseudo differential, besov, riesz |

**Table 4.4.** Neighbours (synonyms) selected by ATC programs as alternative topic labels

the overall quality of their handcrafted counterparts. TaxoGen taxonomies are more structurally rigorous, but they capture a lot of noise and the selected topics lack significance and interpretability. HiExpan taxonomies, on the other hand, include higher quality keywords but are structurally homogeneous with poor granularity. We rank both TaxoCom and HiExpan "off-topic" taxonomies last for their irrelevancy. TaxoGen and TaxoCom taxonomies (TG1, TG2, TC1, TC*) also include alternative titles with varying consistency for each main taxonomy node, as illustrated in table 4.4 and in section 2.3.3, which can be used for improving document matching.

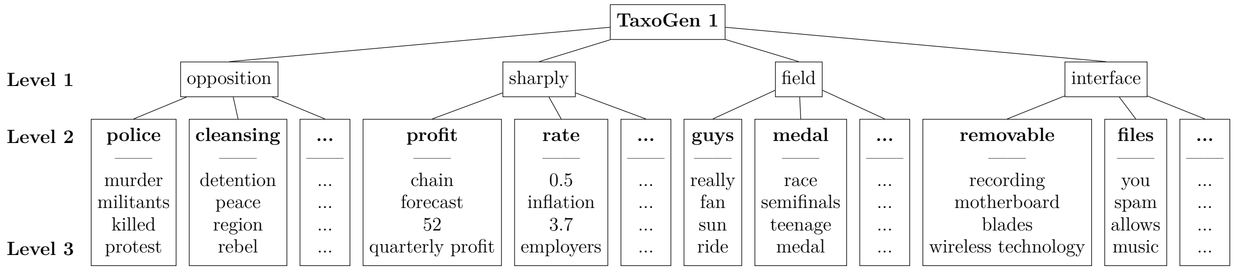


**Fig. 4.1.** "TaxoGen 1" with 3 levels and 4 children per topic (TG1)

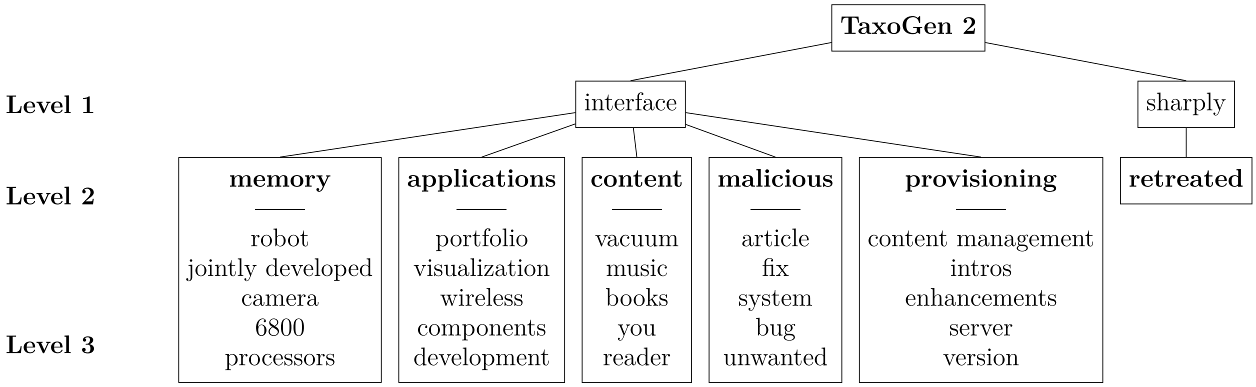


**Fig. 4.2.** "TaxoGen 2" with 3 levels (TG2)

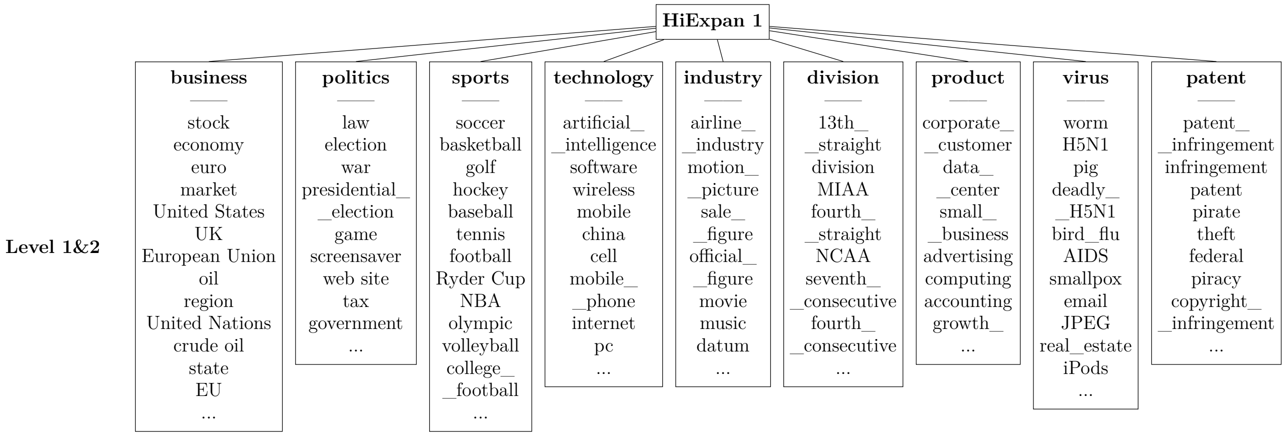


**Fig. 4.3.** "HiExpan 1" with 2 levels and 9 topics (HE1)

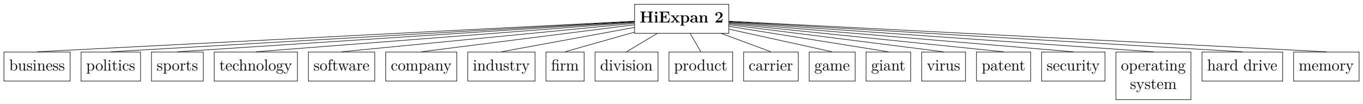


**Fig. 4.4.** "HiExpan 2" with 1 level and 19 topics (HE2)

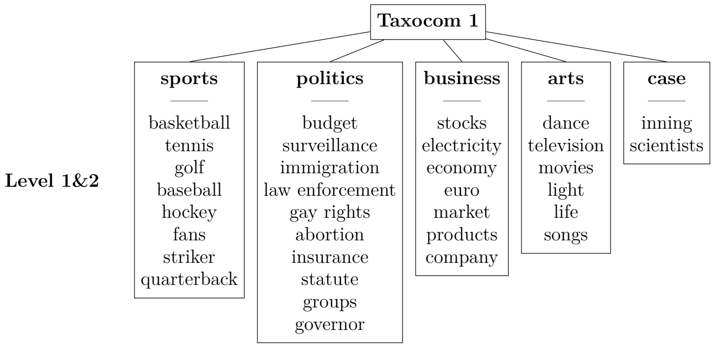


**Fig. 4.5.** "TaxoCom 1" with 2 levels and 5 topics (TC1)

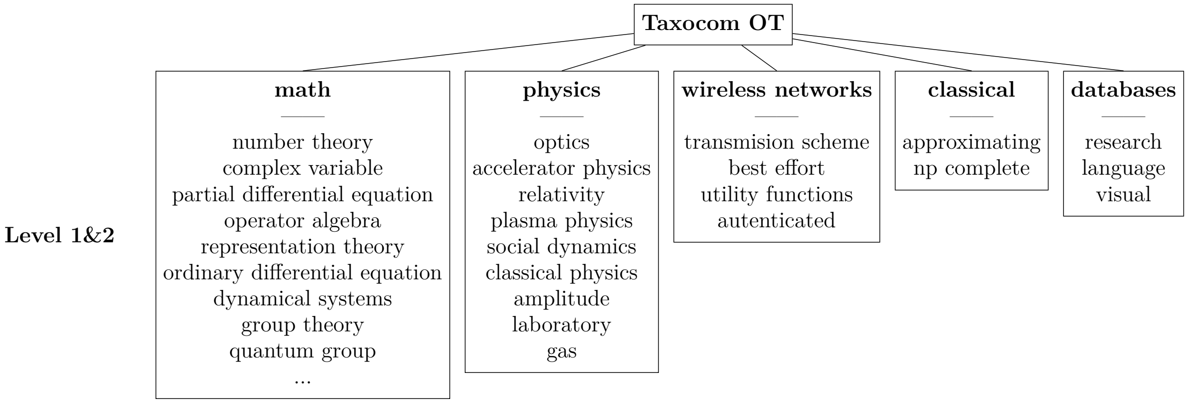


**Fig. 4.6.** "TaxoCom off-topic" generated from arXiv dataset (TC*)

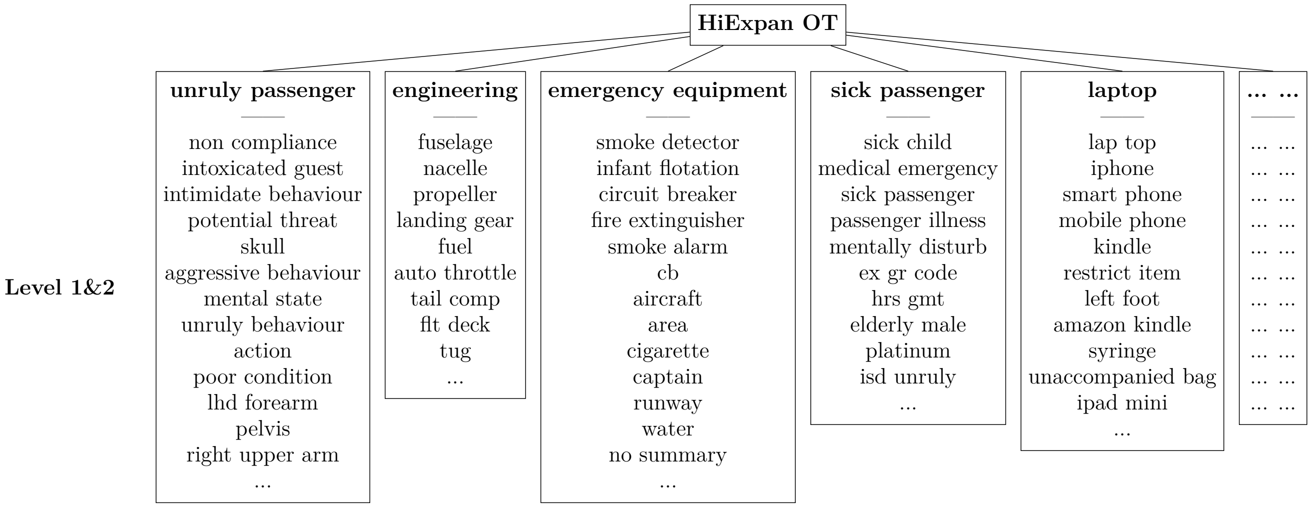


**Fig. 4.7.** "HiExpan off-topic" generated from an aviation dataset (HE*)

## 4.2. Feature Extraction

Text feature extraction primarily relies on count vectorization and TF-IDF discussed in chapter 2 for capturing word frequencies and feature significance to a document category. To convert an ATC taxonomy to categorical features for classification, we started with a one-hot representation that considers each taxonomy entity a categorical feature with equal importance, regardless of the depth of the entity in the taxonomy, i.e. a feature value is 1 if a document contains the entity or its optional neighbours (both associated with a single categorical feature), and 0 otherwise. We then attempted to improve upon the one-hot encoding by assuming that the more a topic word is mentioned in a document, the more significant the topic is to the document, i.e. instead of binary features, the sum of a taxonomy topic's occurence and alternative forms in-text is used as a new feature value to represent how likely a document is to be associated with the topic.

| AG News Data | | Feature Vectors | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Class | Normalized documents | business | politics | sport | technology | company | ... | firm | industry |
| Business | carlyle look toward commercial aerospace... | 1 | 1 | 0 | 0 | 0 | ... | 1 | 3 |
| Business | oil economy cloud stock outlook reuters... | 6 | 0 | 0 | 0 | 0 | ... | 0 | 0 |
| World | pakistan musharraf say quit army chief... | 0 | 1 | 0 | 0 | 0 | ... | 0 | 0 |
| Sports | phelps thorpe advance freestyle ap ap m... | 0 | 0 | 2 | 0 | 0 | ... | 0 | 0 |
| Tech | group propose new high speed wireless f... | 0 | 0 | 0 | 3 | 1 | ... | 0 | 0 |

**Table 4.5.** Categorical feature extraction with HiExpan1 features from all levels. The value of "technology" is greater than 1, being the aggregate score of "company" and other child topics among the classification features.

Transforming taxonomy structure as numerical and classifiable features, however, is complicated. In an initial setup, we encode taxonomic relations by adding the count values of all child topics to the values of their respective parent features from the immediate upper taxonomic level, e.g. adding the counts of "baseball", "basketball", "football" in a document to "sports"; a parent is thus a non-leaf node from the taxonomy of evaluation and inherits all counts from its descendants or direct and indirect subtopics. Aside from concatenating all feature topics into the embedding matrix. We also consider obtaining and comparing classification scores per taxonomic level, such as comparing the level 2 performance of TaxoGen and TaxoCom by embedding level two features only in the feature matrix.

A potential improvement to the count feature transformation is using TF-IDF scores to represent feature importance. Specifically, using a TF-IDF vectorizer configured for unigrams and bigrams for efficiency, we assign the TF-IDF value for each unigram and bigram entity in a taxonomy for evaluation instead of the string count. For efficiency, we assign the TF-IDF value for each unigram and bigram topic in an evaluation target instead of the string count, while taxonomy entities with higher ngram numbers are assigned a score of 0.5 if they are found in the documents but do not have a TF-IDF score. However, trigrams and higher are uncommon during our testing (only 7 counts for AG News). For non-leaf entities, the TF-IDF scores become the maximum between the parents' TF-IDF scores and those of their immoderation children (subordinates) as TF-IDF scores are non-additive. However, currently we find little to no effect to the classifiers between the use of TF-IDF features and occurence features. However, we currently find that using normalized TF-IDF features has little to no improvement for our classifiers.

| AG News Data | | Feature Vectors | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Class | Normalized documents | business | politics | sport | technology | company | ... | firm | industry |
| Business | carlyle look toward commercial aerospace... | 0.064694 | 0.070737 | 0 | 0 | 0 | ... | 0.073609 | 0.117641 |
| Business | oil economy cloud stock outlook reuters... | 0.186722 | 0 | 0 | 0 | 0 | ... | 0 | 0 |
| World | pakistan musharraf say quit army chief... | 0 | 0.067618 | 0 | 0 | 0 | ... | 0 | 0 |
| Sports | phelps thorpe advance freestyle ap ap m... | 0 | 0 | 0.115337 | 0 | 0 | ... | 0 | 0 |
| Tech | group propose new high speed wireless f... | 0 | 0 | 0 | 0.1457255 | 0.051533 | ... | 0 | 0 |

**Table 4.6.** Feature transformation with TF-IDF scoring. The documents are represented by TF-IDF scores of taxonomy entities instead of count features.

### 4.2.1. Improving taxonomy-corpus matching

To better match taxonomy concepts with their associated documents and increase recall, we apply text preprocessing techniques to *normalize* the taxonomies and the corpus.

In boolean retrieval, *lemmatization* and *stemming* are common techniques for reducing word variations including inflections and homonymies. A taxonomy topic "politics", for example, should ideally match documents containing the words "politic", "political" and "politically". While "stemmers" are a set of fast and efficient algorithms for removing word inflections without using morphological rules, "lemmatizers" aim to reduce words to their proper *lemmas* via morphological analysis. The WordNet lemmatizer, for example, refers to WordNet Synsets to ensure a word would fall back to its correct position using the word's part-of-speech tag. We chose lemmatization for our experiments and discovered a 10 to 15% increase in matching rate between all taxonomies of evaluation and the AG News corpus.

| | Stemming | Lemmatization |
|---|---|---|
| 1 | studi_es → studi | studying (verb) → study |
| 2 | studi_ed → studi | studies (noun) → study |
| 3 | studying → studi | studying (verb) → study |

**Table 4.7.** Stemming vs. lemmatization

**Approximate string matching** is an umbrella term for a collection of techniques that can also improve the match between taxonomy keywords and documents containing noise and misspellings. There are two principle fuzzy string matching implementations with *Edit Distance* (ED) and *Word Embeddings*. The Levenshtein distance [**53**], or canonically just *edit* *distance*, is a complex recursive algorithm that can be aided by the Levenshtein matrix to solve the surface-level similarity of two strings. Another metric is the Jaro-Winkler distance [**108**], which takes the average of matching characters and the number of *transpositions* with respect to the lengths of two strings of interest.

New embedding-based similarity metrics are computed otherwise by the cosine distance of two word embeddings generated with Word2Vec [**62**] or a pre-trained language model. As our goal is to associate a taxonomy concept with all relevant corpus documents, as opposed to broadening the scope of a taxonomy (for example, linking taxonomy topic "basketball" with documents containing "baseball"), surface-form edit-distance methods may be more appropriate for our task. However, determining the optimal similarity threshold for edit distance can be tricky in order to avoid false positives. Moreover, Jaro-Winkler scores can also be disproportionately high, even for unrelated concepts. We found that the effect of approximate string matching was minimal and as a result, we relied solely on text preprocessing such as lemmatization in our experiment.

## 4.3. Evaluation Procedure

The input for our evaluation task are (1) an annotated corpus $\mathcal{D}$ of size $k$ and (2) a taxonomy of topics $c_1, c_2, ..., c_n \in \mathcal{T}$ extracted using an ATC program from $\mathcal{D}$. The evaluation target $\mathcal{T}$ is a taxonomy that resembles a tree of topics, or rather some multi-level topic clusters with optional "neighbouring" terms selected by the ATC program as alternative representations for $c_1, c_2, ..., c_n$, e.g. *"agents", "software agents"* and *"multi agent system"* for *"intelligent agents"*. We aim to obtain a proxy score associated with the quality of $\mathcal{T}$, by counting the occurrences of the main taxonomy topics $c_1, c_2, ..., c_n$ and their neighbours if available for each document $d_1, d_2, ..., d_k \in \mathcal{D}$ and obtaining a feature matrix $\mathcal{D}'$ of shape ($n$,$k$) as input for a classification task. Therefore, a document containing no words from the evaluation target will be represented by all zero features. We use the topic labels $l_1, l_2, ..., l_k \in \mathcal{L}$ as target variables, compared to which an optimal $\mathcal{T}$ and topics $c_1, c_2, ..., c_n$ should have a similar class distribution. We set the train-test split to 70/30 for training and testing an ML classifier on unseen data and use $\mathcal{D}'$ and $\mathcal{L}$ to compute a proxy evaluation score, which in our case is the classification accuracy.

## 4.4. Perturbation Test

To test if taxonomy-transformed topic classification can be used for predicting taxonomy quality, we propose gradually increasing the randomness in and generating a set of altered versions of an original taxonomy, to seek correlations between feature randomness and classification accuracy. For our tests, we create feature vectors of the same size as the number of main (level one) taxonomy topics, and whose values are the number of appearances of the main topics and their respective descendants (from level 2, 3 etc.). We then randomly reassign a proportion of the descendants to different parents or remove a percentage of taxonomy edges to obtain a new taxonomy, with which we repeat the process and obtain a modified classification score. The classification outcomes are obtained using a logistic regression model.

## Random Shuffle

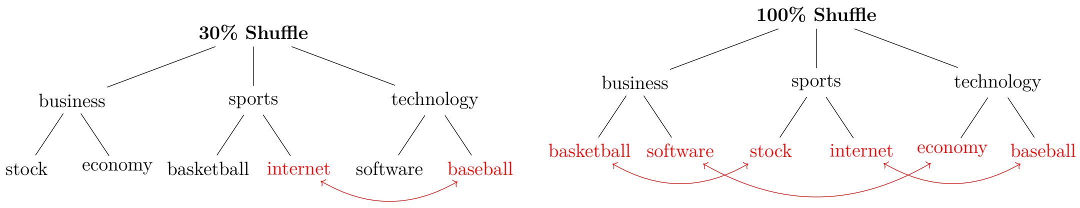


**Fig. 4.8.** Reassigning topics to new positions to simulate a noisy taxonomy

Random shuffle recreates situations in which ATC programs often wrongfully assign an incorrect child topic to a parent. Figure 4.8 depicts two randomly shuffled taxonomies in which the leaf nodes are switched to different topics, resulting in inferior versions of the original topic hierarchies, with the 30% shuffled taxonomy (30% child nodes displaced) still observably more coherent than the 100% shuffled taxonomy (100% leaf nodes displaced from original positions). For our experiment, we create ten randomness settings, increasing the proportion of nodes reassigned to a new topic by 10% per randomness level. Although it is simple to ensure that a subtopic is moved under a new topic, it is however difficult to assert that the old sibling nodes are not also assigned to the same new topic, reducing the effect of random shuffling. As a result, we take the average of five different runs with different random seedings per randomness level. Since the taxonomy structure and feature vector size remain unchanged, we can conclude that incoherency in a taxonomy has a negative impact on classification accuracy, if the scores decrease as we increase random shuffling.

## Random Deletion

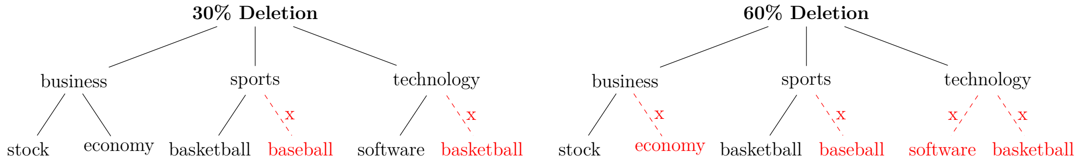


**Fig. 4.9.** Randomly deleting taxonomy edges for reducing taxonomy coverage

In random deletion, taxonomy edges are randomly removed to produce taxonomies with less completeness and lower coverage. To achieve this, we sample and delete child nodes from the main topics, gradually increasing the deletion rate by 10% over time. In figure 4.9 we show two deformed taxonomies of varied levels, with 30% and 60% of the edges or leaf nodes removed respectively (illustrated with dashed lines). As the size of the feature vector is unreduced, lower taxonomy coverage may increase feature sparsity, which negatively influences classification performance.

### 4.4.1. Results

Figure 4.10 demonstrates the results of random shuffle and deletion on the IATA and AG News datasets. Evidently, we can see a clear, near-linear correlation between the level of perturbation in a taxonomy (number of nodes displaced or removed) or decrease in observed taxonomy quality and classification accuracy. For the IATA dataset, we generated two large aviation incident taxonomies CoRel1 and CoRel2 and classified them using ten incident labels.[2] Some jittering in random shuffle with AG News can also be observed because of the small size of the TaxoGen and TaxoCom taxonomies (no neighbours are used for classification) and imprecision in our shuffling procedure.

In addition to random shuffling and deletion, we find in other experiments that attaching noisy subtopics, e.g. random words sampled from the corpus or domain-related entities to the main topics results in nearly identical classification scores or does not increase or decrease classification accuracy. This demonstrates that the proxy score can be at least a good predictor of a coherent and comprehensive taxonomy and is correlated with the observed taxonomy quality.

## 4.5. Evaluation Results and Ranking

We present in this section two distinct taxonomy evaluation results and rankings using our evaluation procedure, with the first focusing on taxonomy (topic) coverage and the second on feature quality. Table 4.8 compares the feature vector size or the number of classification features after conversion, taxonomy coverage and classification accuracy of all evaluation targets generated across **the entire taxonomy-transformed training and testing samples**, which may include a large number of zero-value feature vectors as the documents do not contain any taxonomy topics. We used logistic regression and a 10-fold cross validated decision tree as classification models.

Table 4.8 shows that, while the classification scores have a weak correlation with our manual ranking, they are largely driven and influenced by the number of classification features representing the documents, as well as the percentage of non-zero feature vectors or

[2] Due to copyright of the dataset, the taxonomies are not shown.

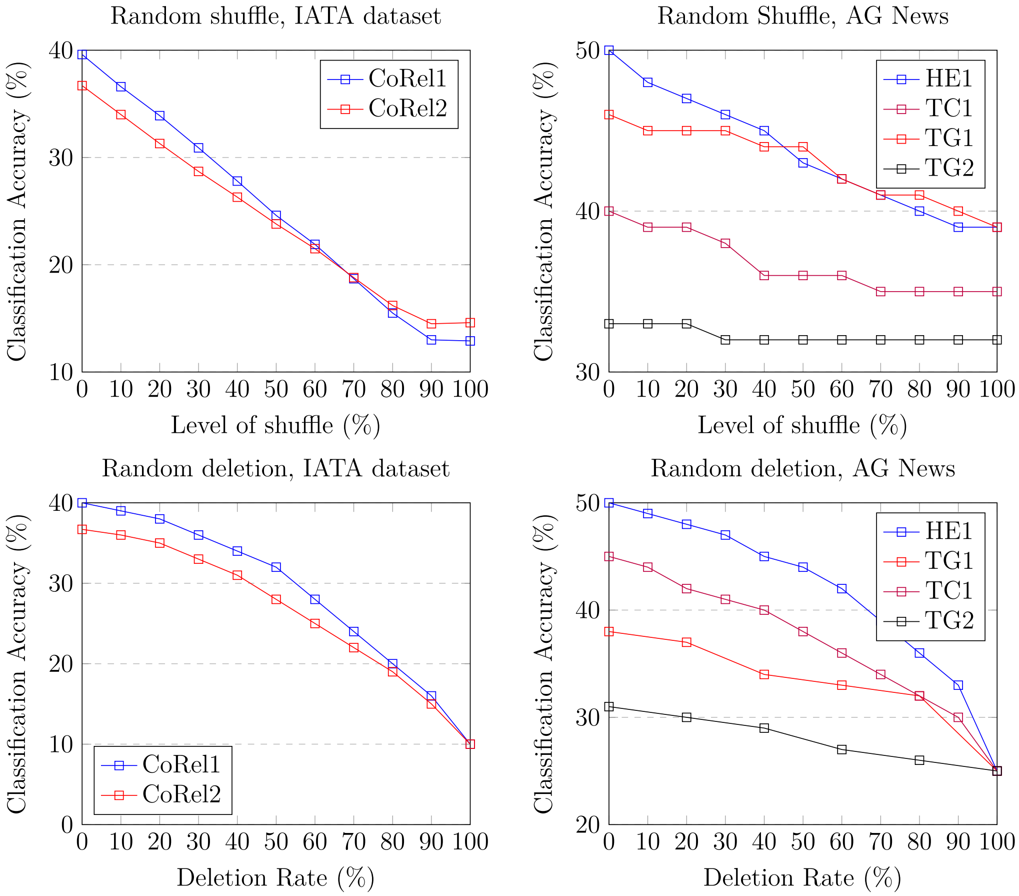


**Fig. 4.10.** Perturbation tests show that the more incoherent or incomplete a taxonomy is, the lower the classification accuracy. Through random shuffling and deletion, we create *corrupted* versions of an original taxonomy with observed quality degradation. For TaxoGen/TaxoCom, we only used the main topics (without neighbours) for generating classification features. Because TaxoGen2 contains so few topics, the effect on classification score is less pronounced.

documents containing at least one taxonomy word for training and testing the classifiers, with the exception of TaxoGen2, TaxoCom1 and HiExpan OT, which have a similar ratio of non-zero features or taxonomy coverage, but higher classification scores for TaxoGen2 and TaxoCom1. As such, our preliminary results can be reliably used for taxonomy ranking with an emphasis on taxonomy coverage. Despite having poor interpretability, TaxoGen1 is given the highest scores among the evaluation targets as its first level topics correspond exactly with the manually assigned news labels, i.e. "opposition" (politics), "sharply" (business), "field", (sports) and "interface" (sci/tech).

To reduce the effect of taxonomy coverage on classification scores, we conduct in table 4.9 a pairwise comparison of classification accuracy that completely disregards documents containing no keywords from either taxonomy target, i.e. for two taxonomies, we only consider the **intersection of non-zero feature rows or records matching both taxonomies**

|  | Classif.<br>Features | Taxonomy<br>Coverage (%) | Logistics<br>Regression | D-tree<br>10-fold CV | Classif.<br>Ranking | Manual<br>Ranking |
|---|---:|---:|---:|---:|---:|---:|
| TaxoGen1 | 79 | 90% | **0.71** | **0.67** | 1 | 3 |
| TaxoGen2 | 33 | 51% | 0.51 | 0.51 | 3 | 4 |
| HiExpan1 | 392 | 84% | 0.68 | 0.66 | 2 | 1 |
| HiExpan2 | 19 | 30% | 0.41 | 0.42 | 6 | 5 |
| TaxoCom1 | 38 | 47% | 0.50 | 0.49 | 4 | 2 |
| TaxoCom OT | 42 | 23% | 0.35 | 0.34 | 7 | 7 (tie) |
| HiExpan OT | 478 | 54% | 0.45 | 0.45 | 5 | 7 (tie) |

**Table 4.8.** Taxonomy scores and ranking calculated by using all training and testing samples, resulting in a large number of zero-value feature rows from documents containing no taxonomy topics. As a result, the classification scores are highly associated with taxonomy coverage or the ratio of corpus documents tagged by a taxonomy. However, both TaxoGen2 and TaxoCom1 comparably show higher scores despite having less taxonomy coverage than HiExpan off-topic (OT).

for classification. This way, large taxonomies are not penalized yet feature importance can play a larger role in classification performance. As only 2,500 out of 127,600 (2%) news articles contain a topic word from each of the seven evaluation targets, pairwise comparison also ensures sufficient data support (at least 5,000 testing samples after 70/30 split) for more accurate results. Our second results show a nearly perfect match with the manual ranking, with minor discrepancies for TaxoGen1 (TG1) and TaxoCom1 (TC1), as even though the former taxonomy has lower lexical quality, the latter omits the entire branch of science and technology topics. Furthermore, we discover that our procedure can also accurately reflect the taxonomy quality on the IATA dataset, but we are unable to show the results here.

|  | TG1 | TG2 | HE1 | HE2 | TC1 | TC* | HE* | Ranking | Manual |
|---|---|---|---|---|---|---|---|---|---|
| TG1 |  | +8% | -2% | +10% | +3% | +27% | +14% | 2 | 3 |
| TG2 | -8% |  | -10% | +8% | -3% | +19% | +5% | 3 | 4 |
| HE1 | +2% | +10% |  | +11% | +5% | +27% | +13% | 1 | 1 |
| HE2 | -10% | -8% | -11% |  | -1% | +20% | +1% | 5 | 5 |
| TC1 | -3% | -8% | -5% | +1% |  | +23% | +7% | 4 | 2 |
| TC* | -27% | -19% | -27% | -20% | -23% |  | -12% | 7 | 7 (tie) |
| HE* | -14% | -5% | -13% | -1% | -7% | +12% |  | 6 | 7 (tie) |

**Table 4.9.** Pairwise classification accuracy comparison between taxonomies A (left column) and B (top column), using only intersections of non-zero feature rows, or texts containing at least one topic from both A and B, therefore reducing the bias towards large taxonomies that associate with more records. The intersections of non-zero feature rows are then split 70/30 to calculate and compare classification scores. The results show strong correlations with manual ranking.

## 4.6. Conclusion

Although using text classification as an application for taxonomy evaluation is quick and simple, and has been shown to some extent to reflect taxonomy quality, particularly topic coherence and coverage as demonstrated in the perturbation tests, its limitations are also self-evident, in that annotated corpora are scarce and modelling topic relations and taxonomy structure is difficult. The most prominent case in our evaluation proxy is that classification accuracy clearly scales with the size or number of concepts in a taxonomy, which can be attributed to either good taxonomy coverage or the model overfitting against noisy features. Prominent classification features are well separated and are unlikely to be found under other topics, e.g. "cleansing" in TaxoGen1 (which is most likely related to politics than business, sports or technology), but they may not translate to meaningful taxonomy topics and our procedure ignores term significance.

In future work, we would like to better integrating taxonomies with classification for an application-based taxonomy evaluation: (1) similar to data augmentation methods that use WordNet [**24**] for synonym lookup [**107**], we can use ATC taxonomies to augment a document corpus through synonym replacement, then use the hypothetical improvement in classification accuracy as a proxy measure for taxonomy quality; (2) using taxonomic features as additional features in classification or leveraging ensemble methods such as stacking, measuring how much a taxonomy can help improve the classification score; (3) integrating taxonomy as external knowledge and a semantic loss function for a ML/DL classifier [**111**].

# Chapter 5

# Unsupervised Taxonomy Evaluation via MLM Prompting

In this chapter, we propose a novel procedure for reproducible and automatic taxonomy evaluation without topic labels or external knowledge using masked language modelling (MLM). Specifically, we seek to automate the scoring of a *relation accuracy* metric, which aims at measuring the portion of true-positive parent-child relations in a given taxonomy. In TaxoGen, this is done via majority voting by asking at least three human evaluators if a taxonomy concept pair contains a hypernymy relationship [**115**], but we found that such relationship can also be learned through adapting language models (LM) to a corpus domain and later exploited via *prompting*, i.e. asking the LM "what are the possible parent types of a topic?" (In fact, a critical part of taxonomy/ontology evaluation is knowledge about the subsumptions, e.g. "is *fluorescence spectroscopy* a type of *fluorescence technology*?" or "is *CRJ200* a *Bombardier*?") We show that instead of relying on expert judgments, using a single or multiple fine-tuned language models with prompting is another viable option for approximating the relation accuracy metric, even in more technical domains.

## 5.1. Masked Language Modelling

Transformer-based language models pre-trained with massive data are shown to excel in many downstream NLP tasks such as machine translation and question answering, or even predicting missing texts in ancient languages [**50**]. The key to transformer's success can be largely attributed to the use of the attention mechanism [**102**], which replaces the context vectors of recurrent neural networks (RNN) with *query* (Q), *keys* (K) and *values* (V) vectors to attend to both global and local contexts. Such attention function can be summarized in equation 5.1.1, where $d_k$ signifies the dimension of keys.

$$\text{Attention (Q, K, V)} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}V\right) \tag{5.1.1}$$

To enable unsupervised language modelling, several learning objectives can be deployed including next word prediction, next sentence prediction and masked language modelling, which is derived from the Cloze test [**99**], originally developed for assessing individuals' language ability. Analogous to "fill in the blanks", the MLM objective aims to randomly mask a sentence token and train the LMs to restore the masked token using surrounding contexts. This would require a bidirectional language model such as BERT [**21**] that can attend to contexts both to the left and to the right of the mask instead of unidirectional models such as OpenAI GPTs [**79**].

```
Sentence: "State of the art Transformer"
MLM objective: "State of the [MASK] Transformer"
Input: "[CLS]", "state", "of", "the", "[MASK]", "transform", "##er", "[SEP]"
Predictions: ("art", 0.988), ("union", 0.002), ("market", 0.001)
```

**Fig. 5.1.** Training a masked language model, an example

In Figure 5.1, a pre-trained WordPiece sentence tokenizer first splits the sentence into words from a predefined vocabulary, which also breaks longer and less common words into two or multiple sub-tokens, e.g. "transformer" to "transform" and "##er", for greater model and tokenizer efficiency. Three of the special tokens of a typical BERT tokenizer include "[CLS]" and "[SEP]", which mark the beginning and ending of a sentence, as well as "[MASK]". The sentence tokens are converted to numerical token (input) IDs and padded or truncated to a fixed length as model input. To complete MLM training, a percentage of sentence tokens are replaced with the mask token and passed to the masked language model. A transformer model for MLM has a classification layer on top of the encoder output layer where *logits* are computed by multiplying the encoder outputs by the embedding matrix and then reshaping them into the dimension of the vocabulary. We can henceforth obtain the model's predictions of the most probable masked tokens as illustrated in figure 5.1 by passing the *logits* of the model vocabulary dimension into the softmax function, which can be disputably regarded as a probability distribution.

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} = \hat{y}_i \tag{5.1.2}$$

The output loss can be calculated using cross-entropy, a divergence measure for quantifying the difference between two distributions, given the real masked token as the real label. In this case, $m_i = 1$ if the token at $i$-th position is masked and 0 otherwise and $y_i$ is the one-hot encoding of the original token [**44**]:

$$Loss = -\sum_{i=1}^{n} m_i \times y_i \log \hat{y}_i \tag{5.1.3}$$

The original BERT implementation chose a 15% probability of masking each token pretraining, where the token is replaced with the "[MASK]" token 80% of the time, a random token 10% of the time, and unchanged 10% of the time [**21**].

## 5.2. Prompting

In hypernymy discovery, a typical form of prompting is simply using MLM to predict the sentence "$x$ is a type of ___" (fill in the blank) so as to discover the attributes or entity types of $x$, e.g. *"crème brûlée is a type of ___ ?"* From section 5.1 we assumed that an optimized language model can always output a list of plausible hypernyms (e.g. "dessert") or highly associated terms with rapport to $x$ given contexts such as Hearst ("is-a") patterns.[1] As a result, we can regard language models as a proxy for domain knowledge and a good taxonomy parent of $x$ should be predictable among the top machine predictions.

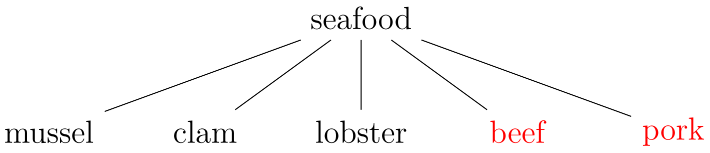


**Fig. 5.2.** Excerpt from HiExpan1 for topic "seafood"

Next, we illustrate our intuition for using prompting with MLM to evaluate a taxonomy's hypernymy relations. Figure 5.2 showcases a subset of an ATC taxonomy with topic *"seafood"* and its five subtopics. It is easy for humans to assert (using common-sense) that the concepts of *beef* and *pork* highlighted in red are incongruous with that of *seafood* as they are in fact subtopics of *"meat"*, for instance. In other words, a potential improvement to this taxonomy subset is dividing the subtopics into *seafood* (for mussel, clam and lobster) and *meat* (for beef and pork). Although it remains inconceivable for machines to fully grasp the true, implicit relationships underlying the above concepts that are apparent to humans only, language models may still capture and preserve high-order contextual word co-occurrence statistics [**93**], statistically associating seafood more closely with mussels and clams given the "is-a" context. This can be nonetheless beneficial for taxonomy construction and evaluation, even if the models are only memorizing parent-child pairs from training data [**38**] or words of sentences containing Hearst-like patterns. Another advantage of using language models is that the models may learn all Hearst patterns automatically and implicitly from the corpus without much specification.

[1] Nevertheless, we find that other types of ontological relations may also be inserted when using prompting, e.g. "$x$ is a part of ___". In fact, this has also opened the door for assessing knowledge representations that include other types of semantic relations including "has-a" and "part-of" for ontologies and knowledge graphs, which we will further examine in future works.

| Query | Prediction 1 | Prediction 2 | Prediction 3 | Prediction 4 | Prediction 5 | Rank |
|---|---|---|---|---|---|---|
| "Mussel is a type of [MASK] ." | fish (0.227) | dish (0.144) | seafood (0.140) | meat (0.037) | soup (0.033) | 3 |
| "Clam is a type of [MASK] ." | fish (0.203) | dish (0.095) | seafood (0.076) | crab (0.030) | thing (0.027) | 3 |
| "Lobster is a type of [MASK] ." | seafood (0.222) | dish (0.145) | lobster (0.131) | food (0.052) | sauce (0.052) | 1 |
| "Chicken is a type of [MASK] ." | dish (0.167) | meat (0.110) | chicken (0.079) | thing (0.058) | sauce (0.052) | 73 |
| "Beef is a type of [MASK] ." | meat (0.274) | beef (0.161) | dish (0.063) | food (0.027) | thing (0.024) | 57 |

**Table 5.1.** Recalling Hypernymy via prompting

In Table 5.1, we create five queries with each containing the subtopics from figure 5.2 and use a pre-trained BERT model (Bert-large-uncased-whole-word-masking) for unmasking. As shown, the language model correctly predicted the taxonomy parent "seafood" for mussel, clam and lobster and "meat" for chicken and beef in the top 5 predictions. Although not every prediction of the BERT-large model is factually correct (e.g. mussels are neither fish nor meat), and it remains evidently unreliable to depend solely upon pre-trained language models as ground-truths for all knowledge domains, we can regard the rankings of MLM predictions as a likelihood of a subsumption relation between the subject and the object of a query, as we show in our results that the model is significantly more likely to predict "seafood" for *mussel, clam* and *lobster* (rank 3,3,1) than for *chicken* and *beef* (rank 73,57).

As such, we have successfully converted automatic Cloze tests into a reproducible, comparable and interpretable hypernym retrieval task for taxonomy evaluation using a certain recall threshold, which we set to the top-10 predictions for our experiment. However, the model's prediction scores are not yet suitable for measuring and comparing the likelihood of taxonomic relations since the softmax is distributed over the entire tokenizer vocabulary (30k), resulting in correct relations having abysmally low scores (e.g. 0.076 for seafood-clam). This could change however if the model can generate probabilities for only selected evaluation topics.

## 5.3. Evaluation Procedure

Using the intuition from section 5.2, we hope to elicit with language models canonical hypernyms pertained to the taxonomy domain for each child node of an evaluation target, and use them as potential ground truths to test whether the taxonomy hypernyms overlap with the LM predictions, thereby automatically examining taxonomic or "is-a" relations and revealing better taxonomies.

To do so, we first flatten the taxonomy of evaluation into parent-child pairs from adjacent taxonomy levels linked by single edges, denoted as $(p,c) \in \mathcal{T}$. For each unique parent-child pair $(p_i,c_i), i \in 1,...,n$, with $n$ being the total number of edges in $\mathcal{T}$, we insert $c_i$ and the "[MASK]" token into prompts containing "is-a" patterns, then use MLMs to unmask $p'_1(c_i), p'_2(c_i), ..., p'_k(c_i) \in p'(c_i)$ per query as proxy parent terms of $c_i$, where $k$ is a recall threshold (e.g. $k = 10$). Ultimately, an optimized language model should be able to generate an accurate sequence of the most canonical hypernyms for a given topic, similar to domain experts. A good taxonomy concept pair is therefore if the real taxonomy parent $p_i$ can also be found among the machine predictions $p'(c_i)$. In the case of table 5.1, we have $p'_1(c_i), p'_2(c_i), ..., p'_5(c_i)$ equal *fish, dish, seafood, meat, soup* for $c_i =$ *mussel*, in which we find the real taxonomy parent $p_i =$ *seafood* $= p'_3(c_i)$.

Our evaluation procedure takes two inputs: $c_i \in c, i \in 1, \ldots, n$ for querying and $p_i \in p, i \in 1, \ldots, n$ for validation, with $(p,c) \in \mathcal{T}$; the output of our procedure is intuitively a relation accuracy score as formulated in equation 5.3.1. Therefore, for a taxonomy with no parent-child pairs, i.e. a single-level taxonomy, our evaluation score is 0. [2] The input for fine-tuning the masked language models for better hypernym retrieval, which we will discuss in section 5.5, is a collection of documents $\mathcal{D}$ related to the evaluation domain as well as a list of user-interested or taxonomy topics that can be found in $\mathcal{D}$. For our experiments, we consider a parent-child relation *positive* if and only if the parent term is recalled one or more[3] times in the *top_k* predictions.

$$Score_{MLM}(\mathcal{T}) = \frac{\text{number of positive parent-child pairs}}{\text{number of all parent-child pairs in the taxonomy}} \tag{5.3.1}$$

### Example

In table 5.2, we sampled 10 parent-child topics from each of the three automatically generated taxonomies (Taxo 1-3), measured the concept pairs using the above method and highlighted the positive relations in red. Due to the complexity of taxonomies, the results of our evaluation are rather debatable. Here we only consider a strict "is-a" relation, i.e. "child" is a type of "parent" for a positive pair of concepts, although other relations may also be technically correct, e.g. a tamale (salad) can be a type of salad in Taxo1. We show the relation accuracy score of each taxonomy in the rightmost column.

## 5.4. Evaluation Targets

We use a subset of the Yelp dataset for reproducing research results [**41**] and extracting restaurant and food-themed taxonomies. The dataset contains 1.9 million mostly-English uncategorized reviews collected from the Yelp Dataset Challenge[4] which includes relevant topics and reviews about food, location, service, and so on. We extract noun phrases and entities from the dataset with AutoPhrase [**86**], a phrase mining and keyword extraction

[2] One solution for evaluating a single-level taxonomy or taxonomy structure in general is to check if the single-level topics contain any pair-wise subsumption relationships using the exact same evaluation procedure, which we will further investigate in future work.

[3] A parent word can be predicted multiple times in singular and plural forms, misspellings, and so on, e.g. "dessert", "desserts" and "desert".

[4] Author: Jiaxin Huang. `https://drive.google.com/drive/folders/13DQ0II9QFLDhDbbRcbQ-Ty9hcJETbHt9`

|  | Sample taxonomy concept pairs (p, c) | Score |
|---|---|---|
| Taxo1 | ('seafood', 'fried oysters'), ('appetizer', 'pakoras'), ('seafood', 'caprese'), ('dessert', 'kung fu tea'), ('salad', 'caesar salad'), ('dessert', 'corned beef hash'), ('dessert', 'pancake'), ('spicy', 'yakitori'), ('salad', 'tamale'), ('burger', 'applewood smoked bacon') | 5/10 |
| Taxo2 | ('burger', 'grilled onions'), ('dessert', 'breading'), ('soup', 'pho'), ('roll', 'sauce'), ('dessert', 'pear'), ('roll', 'carrots'), ('roll', 'chilli'), ('salad', 'goat cheese salad'), ('bread', 'muffins'), ('salad', 'walnuts') | 3/10 |
| Taxo3 | ('salad', 'mustard green'), ('beer', 'french wine'), ('seafood', 'crab'), ('music', 'background music'), ('pizza', 'calzone'), ('sushi', 'cucumber'), ('pizza', 'thin crust'), ('soup', 'prime steak'), ('beer', 'game'), ('seafood', 'chicken') | 4/10 |

**Table 5.2.** Illustration of relation accuracy evaluation

framework from section from section 2.2.2, and generate and select the following ATC taxonomies using our tested systems, including CoRel (Corel 1-4) [**41**], HiExpan (HiExpan 1) [**90**] and TaxoGen (TaxoGen 1-2) [**115**]. The extracted taxonomies are mostly two-levels except those of TaxoGen, which have three levels and a fixed number of child topics for each parent. In table 5.3 we list the main topics, i.e. non-leaf nodes of each taxonomy for evaluation. Since taxonomies are complex structures, in table 5.4 we randomly sampled 10 concept pairs from each taxonomy for qualitative display. An illustration of each generated taxonomy artifact can be found in Appendix A.

| Taxonomy | Top level (main) topics |
|---|---|
| CoRel1 | steak, veggies, beef, cheese, crispy, fish, rice, salad, shrimp, spicy, pork, bacon, burger, appetizer, bread, dessert, seafood |
| CoRel2 | bacon, bread, fries, roll, soup, burger, dessert, salad, shrimp |
| CoRel3 | chinese, seafood, dessert, steak |
| CoRel4 | dinner, food, location, lunch, service |
| HiExpan1 | seafood, salad, dessert, appetizer, food, sushi, "desert", pizza, coffee, bread, pasta, beer, soup, wine, cheese, cocktail, taco, water, music |
| TaxoGen1 | main_dish, south_hills, high_ceilings, "était pas" |
| TaxoGen2 | chest, tempe, amaretto, pepper_jelly, relies, travis, free_admission, exposed_brick |

**Table 5.3.** Main topics of MLM evaluation targets

| | Sampled parent-child pairs (p, c) |
|---|---|
| CoRel1 | ('rice', 'basil leaves'), ('dessert', 'blood orange'), ('appetizer', 'chicken parmigiana'), ('burger', 'skinny fries'), ('veggies', 'basil'), ('spicy', 'fried rice'), ('seafood', 'tuna tartar'), ('beef', 'kung pao chicken'), ('fish', 'tuna'), ('appetizer', 'frites') |
| CoRel2 | ('fries', 'dill'), ('bread', 'cilantro'), ('burger', 'filet mignon'), ('fries', 'melted cheese'), ('bacon', 'ice cream'), ('dessert', 'strawberries'), ('fries', 'cabbage'), ('bacon', 'blueberry'), ('fries', 'mayo'), ('burger', 'mix') |
| CoRel3 | ('dessert', 'sprinkles'), ('dessert', 'marshmallow'), ('dessert', 'alcoholic'), ('seafood', 'taco'), ('dessert', 'smoothie'), ('seafood', 'carne asada'), ('dessert', 'eggs'), ('seafood', 'tortilla'), ('seafood', 'crawfish'), ('chinese', 'sauteed') |
| CoRel4 | ('food', 'overpriced'), ('location', 'refuse'), ('food', 'horrible'), ('food', 'hamburger'), ('service', 'bed'), ('food', 'gravy'), ('location', 'walmart'), ('dinner', 'porterhouse'), ('food', 'bacon'), ('location', 'strip mall') |
| HiExpan1 | ('bread', 'ciabatta bread'), ('sushi', 'yellow tail'), ('food', 'zero star'), ('desert', 'strawberry shortcake'), ('pizza', 'white'), ('music', 'loud music'), ('wine', 'prosecco'), ('salad', 'mixed green'), ('dessert', 'waffle'), ('desert', 'cassoulet') |
| TaxoGen1 | ('main dish', 'dinners'), ('wood', 'comfortable chairs'), ('wood', 'lanterns'), ('buffalo chicken wrap', 'portabello'), ('main dish', 'korma'), ('wood', 'sofas'), ('sashimi', 'octopus'), ('lounging', 'outdoor patio'), ('lasagna', 'brother'), ('carne asada', 'enchiladas') |
| TaxoGen2 | ('relies', 'remain'), ('short ribs', 'carnitas'), ('short ribs', 'mussels'), ('chest', 'numb'), ('chest', 'chewed'), ('free admission', 'dinner'), ('travis', 'friendly'), ('relies', 'execution'), ('travis', 'rude'), ('pepper jelly', 'onion straws') |

**Table 5.4.** Evaluation parent-child samples

## 5.5. Improving Hypernymy Predictions

In this section, we discuss two methods for increasing the hit rate of hypernymy predictions of taxonomy subjects and reducing false negatives by (1) creating various prompts and (2) fine-tuning MLMs with different masking procedures using expanded vocabulary.

### 5.5.1. Diversified Prompting

Studies on MLM prompting universally find that differences in prompts used can actively impact a model's performance in hypernymy retrieval [**74, 38**], which can lead to trivial mask predictions such as stop-words (e.g. "**this** is a kind of seafood"), expressions and collocations found frequently in training samples (e.g. "seafood is a kind of **joke/disappointment**") and coarse-grained topics (e.g. "seafood is a kind of **dish**"). Hanna and Mareček [**38**] discovered that hypernymy discovery by prompting BERT can actually outperform other unsupervised methods even in an unconstrained scenario, but the effectiveness of it depends on the actual queries [**38**]. For example, query "A(n) $x$ **is a** [MASK]" outperformed "A(n) $x$ **is a type of** [MASK]" in P@1 and MRR ranking metrics during the authors' diagnostics on the Battig dataset. However, we argue that the claim varies for different datasets.

| | Prompt | Pred1 | Pred2 | Pred3 | Pred4 | Pred5 | Rank |
|---|---|---|---|---|---|---|---|
| p1a | {shrimp} [MASK] | salad | cocktail | pasta | soup | rice | 359 |
| p1b | [MASK] {shrimp} | fried | no | garlic | coconut | fresh | 117 |
| p2a | {shrimp} is a [MASK] | joke | must | winner | favorite | hit | 959 |
| p2b | {shrimp} is an [MASK] | option | issue | experience | art | order | 4407 |
| p3a | {shrimp} is a kind of [MASK] | joke | thing | dish | treat | disappointment | 146 |
| p3b | {shrimp} is a type of [MASK] | dish | thing | food | sauce | seafood | 5 |
| p3c | {shrimp} is an example of [MASK] | that | this | shrimp | food | seafood | 5 |
| p4a | [MASK] such as {shrimp} | sides | food | seafood | fish | shrimp | 3 |
| p4b | A [MASK] such as {shrimp} | lot | variety | side | combination | protein | 40 |
| p4c | An [MASK] such as {shrimp} | ingredient | item | option | order | animal | 197 |
| p5a | My favorite [MASK] is {shrimp} | dish | thing | part | item | roll | 16 |

**Table 5.5.** Evaluation queries for "seafood-shrimp" of HiExpan1

As a result, instead of relying on a single query, we design five pattern groups (p1-p5) of hypernymy tests for pooling unmasking results and preventing trivial predictions such as collocations from lowering the rank of the true taxonomy parent and producing false negatives. In the case of table 5.5, we use the tests to validate a parent-child pair (seafood-shrimp) in HiExpan1 and expect to find the taxonomy parent "seafood" among the top 5 predictions for all queries. However, only prompts p3b, p3c and p4a returned "seafood" in the top predictions (with rank 5,5,3), as other prompts all produced completely different predictions based on the mask token's functions in those sentences. While p2 to p4 follow standard Hearst-like patterns [**39**], p5(a) employs the "my favourite is" prompt that has demonstrated high P@1 and MRR in [**38**]. The different prompt types are also required for evaluating other types of parent-child relationships: p1(a,b) are created specifically for noun phrases, which have a tendency to be split and considered as good taxonomy edges by ATC systems. For instance, while "shrimp" is not a subtype of "salad" or "cocktail" per se in the example of g1a, the two terms are still considered true taxonomy parents in our experiment since "shrimp salad" and "shrimp cocktail" are both valid food items. A topic pair has therefore a score of 1, as in the seafood-shrimp example, if the parent term is among the top-k machine predictions for any inquiries containing the child topic, and 0 vice versa.[5]

$$Score(p,c) = \begin{cases} 1 & \text{if } p \text{ is retrieved in the top-k predictions of p1-p5} \\ 0 & otherwise \end{cases}$$

### 5.5.2. Fine-tuning

To improve hypernymy predictions, we must also address two issues with pre-trained language models: (1) the models are untrained on the evaluation domain; (2) the default BERT tokenizer and model vocabulary are oblivious of some taxonomy topics, resulting in zero recall. We find that most research on MLM prompting only assessed the performance of pre-trained models. Kawintiranon and Singh [**44**], however, showed that fine-tuning can be used to enhance MLM on political stance classification. Peng et al. [**74**] found an improvement in accuracy and mean rank of retrieval (MRR) when using FinBert models [**113**] pre-trained with massive financial corpora in retrieving financial hypernyms such as *equity* and *credit* for *"S&P 100 index is a/an ___ index"*, compared to using BERT-base [**21**]. Dai and Wang [**16**] generated ultra-fine entity typing labels, e.g. "person, soldier, man, criminal" for *"**he** was confined at Dunkirk, escaped, set sail for India"* through inserting hypernym extraction patterns and training LMs to predict such patterns. Analogously, we can compare fine-tuning MLMs to training domain experts via the Cloze test objective [**99**]. Although assessing hypernym predictions is challenging without gold standards, we created six fine-tuned models with various masking protocols, model vocabulary and training sizes, in the hopes of observing qualitative differences in the predictions.

[5] Another implementation may include a normalized likelihood score between 0-1, calculated by *(number of positive queries) / (number of total queries)* per taxonomy child, e.g. 7/10 if 7 queries out of 10 returned the parent topic for a child topic.

| Model Name | Base Model | Vocabulary | Training Size | Masking Protocol | Fine-tuned |
|---|---|---|---|---|---|
| Model1a | Bert-base-uncased | Bert-base-uncased<br>+31 food terms | Entire dataset<br>1.9m reviews | Entity masking<br>15% tokens | 2 epochs |
| Model1b | Bert-base-uncased | Bert-base-uncased<br>+31 food terms | Entire dataset<br>1.9m reviews | Entity masking<br>one entity only | 2 epochs |
| Model2a | Bert-base-uncased | Bert-base-uncased<br>+31 food terms | Entire dataset<br>1.9m reviews | Random masking<br>15% tokens | 2 epochs |
| Model2b | Bert-base-uncased | Bert-base-uncased<br>+31 food terms | 70% dataset<br>1.3m reviews | Random masking<br>15% tokens | 2 epochs |
| Model0 | Bert-base-uncased | Bert-base-uncased | 70% dataset<br>1.3m reviews | Random masking<br>15% tokens | 2 epochs |
| Model0e | Distilbert-base-uncased | Distilbert-base-uncased | 70% dataset<br>1.3m reviews | Entity masking<br>15% tokens | 2 epochs |
| BERT-large | Bert-large-uncased-<br>whole-word-masking | Bert-large-uncased-<br>whole-word-masking | N/A | N/A | No |
| BERT-base | Bert-base-uncased | Bert-base-uncased | N/A | N/A | No |

**Table 5.6.** Configurations of the fine-tuned evaluation models (model 0,1,2), with Model0 serving as baselines for training with the base tokenizer and a smaller sample size (for faster training time). For our experiment, we also select two pre-trained models (BERT-large and base) for comparisons.

First, we experiment with *entity masking* while fine-tuning model 1a, 1b and 0e, which emphasizes masking task-relevant tokens and is shown to be more effective than *random masking* in studies [**97, 44**]. Because we want the language models to concentrate on the taxonomy entities, particularly the parent terms and their surrounding contexts and word relationships, we prioritize therefore masking the main topics (shown previously in table 5.3) and parent terms of the evaluation targets, then other taxonomy entities (e.g. leaf nodes), followed by AutoPhrase entities if no taxonomy entities are present in the sentence and other random tokens from our training samples. In addition, we test entity masking by only masking *one* taxonomy entity rather than 15% of sentence tokens to gain more sentence contexts. Table 5.7 illustrates the entity and random masking procedures.

Next, we enrich the vocabulary of model 1 and 2, by adding the lemmas or singular forms of parent terms from table 5.3 that were not previously included in the base tokenizer, such as "sushi", "appetizer" and "carne asada", and resizing the models' token embedding matrices to match the size of the new tokenizer. The new tokens are initialized randomly for fine-tuning, although it is possible to manually assign them with the weights of the closest terms in the original vocabulary. This ensures that domain-specific words such as food items can be predicted as a whole word rather than being overlooked by the language models. By adding only a small number of new tokens to the model and tokenizer, we also ensure similar model and tokenizer efficiencies. We believe that vocabulary extension will become a necessary step for effective hypernymy prediction in most specialized domains, though the exact optimal strategies remain to be discussed. The newly added parent terms can be found in Appendix A.

| | |
|---|---|
| Sample Review | "Everything was pretty good but the beef in the mongolian beef was very chewy and had a weird texture." |
| Taxonomy Entities | beef (CoRel1-4, HiExpan1), mongolian (CoRel1-4) |
| AutoPhrase Entities | beef, chewy, mongolian, weird texture |
| Masking Priority | 1. beef (CoRel1 main topic); 2. mongolian (taxo.); 3. chewy (AutoPhrase); 4. weird texture (AutoPhrase) |
| Entity Masking (15%) | "Everything was pretty good but the [MASK] in the [MASK] [MASK] was very chewy and had a weird texture." |
| Entity Masking (one) | "Everything was pretty good but the [MASK] in the mongolian [MASK] was very chewy and had a weird texture." |
| Random Masking (15%) | "Everything was pretty [MASK] but the beef in the mongolian beef [MASK] very chewy and had a [MASK] texture." |

**Table 5.7.** Comparison between entity and random masking. Here we show a sample Yelp review with entities from the evaluation targets (CoRel1-4, HiExpan1, TaxoGen1-2) and entities extracted by AutoPhrase found in the review. We prioritize masking the taxonomy entities, AutoPhrase entities and random tokens, in that order.

To highlight the qualitative differences between our evaluation models, we provide a simple prompt "my favorite [MASK] is sirloin" for the models to predict the taxonomy hypernym "steak" in CoRel1. The results are shown in table 5.8, where 5 out of 6 fine-tuned models and none of the pre-trained models correctly predicted the taxonomy parent in the top 5 predictions. (In fact, it may seem absurd for the base models to associate terms such as "fruit" or "drink" with "steak".) Further, all fine-tuned models returned "steak" in the top ten predictions.

| | Model | Pred1 | Pred2 | Pred3 | Pred4 | Pred5 | Rank |
|---|---|---|---|---|---|---|---|
| Fine-tuned Models | Model1a | burger | dish | sandwich | steak | beer | 4 |
| | Model1b | dish | burger | beer | sandwich | pizza | 10 |
| | Model2a | steak | dish | meat | cut | one | 1 |
| | Model2b | steak | dish | burger | meat | here | 1 |
| | Model0 | dish | burger | steak | meat | sandwich | 3 |
| | Model0e | cut | steak | meat | beef | burger | 2 |
| Pre-trained Models | Large Whole Word | fruit | flavor | food | color | herb | 69 |
| | Bert Base | food | drink | color | dessert | fruit | 71 |

**Table 5.8.** Fine-tuned vs. pre-trained models, "my favorite [MASK] is sirloin ."

Lastly, we show the positive effects of tokenizer extension in table 5.9. In this example, we wish to recall the parent term "appetizer" for the concept pair "appetizer-mozzarella sticks" in CoRel1, where token "appetizer" would be split into *'app', '##eti' and '##zer'* by the standard tokenizer. Both Model1a and Model1b trained with entity masking and an expanded vocabulary correctly predicted "appetizer" in their top five predictions; Model2(a,b) also recalled the term, albeit with a very low rank whereas other models are completely oblivious to it. Nevertheless, we find that expanding the model's vocabulary in conjunction with entity masking may introduce bias into the models when training with limited training samples, i.e. always predicting the added tokens.

| | Model | Pred1 | Pred2 | Pred3 | Pred4 | Pred5 | Rank |
|---|---|---|---|---|---|---|---|
| extended vocabulary | Model1a | sides | foods | food | apps | appetizer | 5 |
| | Model1b | sides | food | appetizer | foods | sandwiches | 3 |
| | Model2a | sides | items | food | dessert | things | 6089 |
| | Model2b | things | items | foods | props | sides | 3111 |
| base vocabulary | Model0 | sides | extras | items | dessert | staples | N/A |
| | Model0e | sides | apps | foods | snacks | extras | N/A |
| | BERT-large | foods | items | products | food | snacks | N/A |
| | BERT-base | foods | snacks | food | items | products | N/A |

**Table 5.9.** extended vs. base vocabulary, "[MASK] such as mozzarella sticks"

## 5.6. Results

Table 5.10 showcases our preliminary results of MLM taxonomy relation accuracy evaluation, calculated by the number of positive relations over all unique parent-child pairs in a taxonomy. We also considered word inflections and certain special cases to improve matching between taxonomy terms and machine predictions, e.g. "veggies", "vegetable" and "vegetables"; "dessert" and "desert". The entity-masking models 1a and 1b predicted the most positive relationships in each candidate taxonomy while the pre-trained models predicted the fewest. It is also surprising that BERT-base outperforms BERT-large when it comes to matching more positive concept pairs. However, all models produce similar score distributions, with the HiExpan taxonomy receiving the highest scores and the TaxoGen taxonomies receiving the lowest. **This is consistent with our manual judgements in that the HiExpan concept pairs were derived from an accurate relation dataset (Probase), whereas TaxoGen1 and TaxoGen2 contain mostly noise.** We also compute the majority voting scores for each evaluation target using the six fine-tuned models (model 0, 1, 2): a concept pair of a taxonomy is positive if and only if three or more models have successfully predicted the parent word. To demonstrate the quality of the taxonomies and the MLM predictions, in table 5.12 we sampled the Model2a evaluation matrix for several parent-child pairs in each taxonomy, where each column corresponds to one of the prompt types shown in table 5.5, a "1" indicates that the parent term was retrieved among the top ten predictions for the prompt, and a "0" indicates that it was not found. In row 1, "artichoke is a type of [MASK]" and "[MASK] such as artichokes" both returned the taxonomy parent "veggies" (vegetables), but none of the tests returned the word for "seafood" in row 2. The terms are singularized before being inserted the prompts.

| | Model1a | Model1b | Model2a | Model2b | Model0a | Model0b | BERT-large | BERT-base | Majority Voting |
|---|---|---|---|---|---|---|---|---|---|
| CoRel1 | 0.727 | 0.718 | 0.424 | 0.445 | 0.463 | 0.436 | 0.204 | 0.274 | 0.443 |
| CoRel2 | 0.782 | 0.750 | 0.544 | 0.537 | **0.572** | 0.512 | 0.259 | 0.362 | 0.572 |
| CoRel3 | 0.602 | 0.667 | 0.541 | 0.549 | **0.572** | 0.501 | 0.360 | 0.400 | 0.535 |
| CoRel4 | 0.682 | 0.646 | 0.450 | 0.390 | 0.365 | 0.381 | **0.410** | 0.418 | 0.347 |
| HiExpan1 | **0.845** | **0.847** | **0.595** | **0.567** | 0.569 | **0.643** | 0.349 | **0.420** | **0.590** |
| TaxoGen1 | 0.135 | 0.147 | 0.055 | 0.061 | 0.012 | 0.025 | 0.031 | 0.037 | 0.012 |
| TaxoGen2 | 0.000 | 0.000 | 0.000 | 0.045 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

**Table 5.10.** Relation accuracy scores evaluated by language models, calculated by the number of positive relations, or parent terms in the model predictions, divided by the number of unique parent-child pairs in each taxonomy.

Using the scores from table 5.10, we rank the taxonomies for evaluation in table 5.11. As we previously agreed that HiExpan1 and TaxoGen 1, 2 are the best and the worst taxonomies in the bunch (ranking 1,6,7), our final step is to produce a manual ranking for the four CoRel candidates whose ranks differ in the machine results. Due to the large size of each taxonomy, we gave impressions and randomly sampled 20 concept pairs for validation. We find that all CoRel candidates have comparable levels of good parent-child pairs as well as unrelated topics that frequently co-occur in the same contexts (e.g. "steak-tomato"). We also examined the machine predictions as shown in table 5.12, and assigned a ranking of 3, 2, 4, 5 to CoRel 1-4. We find that MLM majority voting strongly correlates with our manual judgements, with minor disagreements for CoRel 1 & 3.

|  | Model1a | Model1b | Model2a | Model2b | Model0a | Model0b | BERT-large | BERT-base | **Majority Voting** | **Manual Ranking** |
|---|---|---|---|---|---|---|---|---|---|---|
| CoRel1 | 3 | 3 | 5 | 4 | 4 | 4 | 5 | 5 | **4** | **3** |
| CoRel2 | 2 | 2 | 2 | 3 | 1 | 2 | 4 | 4 | **2** | **2** |
| CoRel3 | 5 | 4 | 3 | 2 | 1 | 3 | 2 | 3 | **3** | **4** |
| CoRel4 | 4 | 5 | 4 | 5 | 5 | 5 | 1 | 2 | **5** | **5** |
| HiExpan1 | 1 | 1 | 1 | 1 | 3 | 1 | 3 | 1 | **1** | **1** |
| TaxoGen1 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | **6** | **6** |
| TaxoGen2 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | **7** | **7** |

**Table 5.11.** Taxonomy rankings of automatic relation accuracy evaluation

|  |  |  | NP |  | is-a |  | "kind-of" |  |  | such-as |  |  | fav-is |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Taxo | Parent | Child | p1a | p1b | p2a | p2b | p3a | p3b | p3c | p4a | p4b | p4c | p5a | Sum |
| CoRel1 | veggies | artichokes | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 3 |
|  | seafood | grilled pork | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | steak | rib eye steak | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 5 |
|  | bacon | gruyere | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CoRel2 | soup | bisque | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 6 |
|  | salad | soy sauce | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | shrimp | wonton | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
|  | burger | tortilla chips | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CoRel3 | dessert | cheesecake | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 7 |
|  | seafood | prosciutto | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | dessert | cake | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 7 |
|  | chinese | miso | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CoRel4 | location | chinatown | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 4 |
|  | lunch | spring rolls | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | dinner | juicy | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | service | extremely rude | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 4 |
| HiExpan1 | salad | caesar | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 5 |
|  | pizza | ny style | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 7 |
|  | pasta | bolognese | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 7 |
|  | cheese | cheddar | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 6 |
| TaxoGen1 | wine bar | sushi joint | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | carne asada | pollo | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
|  | chow mein | egg roll | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | ny | wisconsin | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| TaxoGen2 | exposed brick | music videos | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | short ribs | sashimi | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | amaretto | bourbon | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | travis | friendly | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

[NP] child parent (p1a), parent child (p1b)
[is-a] child is-a parent (p2a), child is-an parent (p2b)
["kind-of"] child is-a-kind-of parent (p3a), child is-a-type-of parent (p3b), child is-an-example-of parent (p3c)
[such-as] parent such-as child (p4a), a parent such-as child (p4b), an parent such-as child (p4c)
[fav-is] my favourite parent is child (p5a)

**Table 5.12.** Sample relation accuracy evaluation results (Model2a)

## 5.7. Conclusion

According to our experiment results, hypernymy prediction through the use of masked language models can be a reliable proxy for reproducible taxonomy evaluation and ranking. We demonstrate that such a method can rival manual rankings and produce a rough estimate of taxonomy quality and relation accuracy at least for our tested dataset. By masking random or entity tokens in a domain corpus, MLMs may learn to associate domain-specific hypernymy using contexts and generate more relevant domain entities than pre-trained models. Other benefits of such an approach include MLM's ability to automatically learn Hearst-like patterns without specification, and possibly predicting the correct super-types via association for taxonomy terms not found in corpus (e.g. concepts from a manually created taxonomy). In addition, we proposed and implemented several model tuning and prompting strategies to improve hypernymy recall.

However, there are several obstacles to our approach, in that: (1) The WordPiece tokenizer can break up non-vocabulary words and prevent the model from recalling rare domain entities. Adding custom vocabulary into the model may introduce bias and slow down the training process. For our experiment, we only added the parent terms that did not already exist in the base tokenizer and resized the model's token embedding matrix to match the size of the new tokenizer. Because the new tokens are initialized with random weights, more training samples are required. (2) Rarer entities such as domain keywords may not be recalled as frequently as more common words such as stop words and collocations. To improve this, we can change BERT's output layer to only generate scores for selected taxonomy terms. (3) Prompts must be manually created for each evaluation dataset, and the best prompts for each evaluation domain and relation types are left to be discovered.

In future works, we hope to quantify the hypernym prediction results using external knowledge bases and labelled data, as well as use prompting on ontological relationships such as "has-a". It is also possible to directly generate semantically coherent taxonomies using prompting without relying on iterative clustering, which can omit important lexical semantic relations. In the end, we advocate for using real-world knowledge from trusted external sources such as WordNet [**24**] and Cloze tests to assess the knowledge learned by language models.

# Chapter 6

# Conclusion

The ability to automatically evaluate taxonomies and ontologies opens up a plethora of new possibilities, from systematically assessing the quality of knowledge representations to quantitatively reproducing and comparing evaluation results. We find that the major hindrance to automatic taxonomy scoring and ranking is the absence of ground-truth and external knowledge for verifying subsumption or "is-a" relationships that are omnipresent in a taxonomy, which also dictates human input and slows down the process of taxonomy creation and assessments. As a result, we proposed automatically learning essential topic relations required for domain taxonomy evaluation from: (1) manually annotated topic labels and, in the absence of such topic labels, (2) domain corpora using language modelling. We developed two novel taxonomy evaluation procedures with the aforementioned resources: the first of which used classifiers and data labels to generate quantitative scores based on the quality and coverage of classification features converted from taxonomy topics, and the second employed auto-evaluators for estimating the hypernymy relation accuracy of taxonomy parent-child pairs in an unsupervised fashion.

While both methods have been shown to produce consistent evaluation results, the ML classification proxy extends the application-based ontology evaluation approach and is quick in determining if a taxonomy matches the latent topic distribution provided in data labels, whereas the language modelling approach traces back to the fundamental issue of lacking domain knowledge and provides an universal solution for all taxonomy evaluation scenarios. However, both of our proposed methods satisfied the four propositions for machine taxonomy evaluation, proposed at the outset of this work, in that: (1) better taxonomies should result in higher scores on our evaluation tasks; (2) machine evaluation should correlate with human judgements; (3) the evaluation procedures should be data and model-agnostic, with no reliance on a particular dataset or taxonomy/ontology extractor; (4) the evaluation results should be reproducible. Despite this, our methods are not without flaws, such as the fact that topic classification alone cannot account for topic significance, which determines a taxonomy's usefulness at the lexical level. In addition, adapting language models to technical domains through fine-tuning is time-consuming and resource-intensive, and predicting rare domain entities is difficult. Despite this, our machine evaluation results strongly agree with human judgments, indicating that machine learning and language modelling possess the potential to become viable options for reproducible, automatic taxonomy evaluation.

In future work, improvements in feature extraction are required for the classification proxy to better represent the taxonomy structure and inter-topical relationships and reduce sparsity in the feature vectors. We also propose several new ideas that broaden our existing application-based evaluation approach, through (1) augmenting data with ATC taxonomies and measuring performance improvement; (2) using taxonomic features in ensemble with regular text classifiers; (3) incorporating taxonomic knowledge as a semantic loss function for DL and ML classifiers. For our MLM procedure, we aim to discover systematical ways to generate prompts for various data and taxonomy types, as well as to experiment with prompting on other types of semantic relationships such as those in a knowledge graph. Because the existing ATC paradigm that performs iterative clustering over word embeddings is inherently oblivious to nuanced taxonomic relations and granularity, we should theoretically be able to generate far more accurate topic taxonomies using language models, edging closer to the more authentic taxonomy creation process carried out domain experts: "is *lightning strike* a type of *weather event*?" Finally, we hope to use our knowledge and experience with word relation assessment towards systematically evaluating real-world knowledge learned by language models. ("Are *mussels* a kind of *fish*?")

# References

[1] Muhammad Amith, Zhe He, Jiang Bian, Juan Antonio Lossio-Ventura et Cui Tao : Assessing the practice of biomedical ontology evaluation: Gaps and opportunities. *Journal of Biomedical Informatics*, 80:1–13, 2018.

[2] Paolo Avesani, Fausto Giunchiglia et Mikalai Yatskevich : A large scale taxonomy mapping evaluation. *In International Semantic Web Conference*, pages 67–81. Springer, 2005.

[3] Arindam Banerjee, Inderjit S Dhillon, Joydeep Ghosh, Suvrit Sra et Greg Ridgeway : Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6(9), 2005.

[4] Yoshua Bengio, Réjean Ducharme et Pascal Vincent : A neural probabilistic language model. *Advances in neural information processing systems*, 13, 2000.

[5] Matthew Berland et Eugene Charniak : Finding parts in very large corpora. *In Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, pages 57–64, 1999.

[6] Gabriel Bernier-Colborne et Caroline Barriere : Crim at semeval-2018 task 9: A hybrid approach to hypernym discovery. *In Proceedings of the 12th international workshop on semantic evaluation*, pages 725–731, 2018.

[7] Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma et Prateek Jain : Sparse local embeddings for extreme multi-label classification. *Advances in neural information processing systems*, 28, 2015.

[8] Philip Bille : A survey on tree edit distance and related problems. *Theoretical computer science*, 337(1-3):217–239, 2005.

[9] Georgeta Bordea, Paul Buitelaar, Stefano Faralli et Roberto Navigli : SemEval-2015 task 17: Taxonomy extraction evaluation (TExEval). *In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 902–910, Denver, Colorado, juin 2015. Association for Computational Linguistics.

[10] Georgeta Bordea, Els Lefever et Paul Buitelaar : Semeval-2016 task 13: Taxonomy extraction evaluation (texeval-2). *In Proceedings of the 10th international workshop on semantic evaluation (semeval-2016)*, pages 1081–1091, 2016.

[11] Janez Brank, Marko Grobelnik et Dunja Mladenić : A survey of ontology evaluation techniques. *In Proc. of 8th Int. multi-conf. Information Society*, pages 166–169, 2005.

[12] Janez Brank, Marko Grobelnik et Dunja Mladenić : Automatic evaluation of ontologies. *In Natural language processing and text Mining*, pages 193–219. Springer, 2007.

[13] C. Brewster, H. Alani, S. Dasmahapatra et Y. Wilks : Data Driven Ontology Evaluation. *In Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*, Lisbon, Portugal, 2004.

[14] Tadeusz Caliński et Jerzy Harabasz : A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

[15] Philipp Cimiano, Andreas Hotho et Steffen Staab : Comparing conceptual, divisive and agglomerative clustering for learning taxonomies from text. *In Proceedings of the 16th European Conference on Artificial Intelligence*, ECAI'04, page 435–439, NLD, 2004. IOS Press.

[16] Hongliang Dai, Yangqiu Song et Haixun Wang : Ultra-fine entity typing with weak supervision from a masked language model. *arXiv preprint arXiv:2106.04098*, 2021.

[17] Sarthak Dash, Md Faisal Mahbub Chowdhury, Alfio Gliozzo, Nandana Mihindukulasooriya et Nicolas Rodolfo Fauceglia : Hypernym detection using strict partial order networks. *In Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7626–7633, 2020.

[18] Yann Dauxais, Urchade Zaratiana, Matthieu Laneuville, Simon David Hernandez, Pierre Holat et Charlie Grosman : Towards automation of topic taxonomy construction. *In International Symposium on Intelligent Data Analysis*, pages 26–38. Springer, 2022.

[19] David L Davies et Donald W Bouldin : A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.

[20] Jeroen De Knijff, Flavius Frasincar et Frederik Hogenboom : Domain taxonomy learning from text: The subsumption method versus hierarchical clustering. *Data & Knowledge Engineering*, 83:54–69, 2013.

[21] Jacob Devlin, Ming-Wei Chang, Kenton Lee et Kristina Toutanova : Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[22] Fernando Diaz, Bhaskar Mitra et Nick Craswell : Query expansion with locally-trained word embeddings. *arXiv preprint arXiv:1605.07891*, 2016.

[23] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld et Alexander Yates : Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165(1):91–134, 2005.

[24] Fellbaum : *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, mai 1998.

[25] Geoffrey Finch : *Linguistic terms and concepts*. Macmillan International Higher Education, 2016.

[26] Thomas François et Eleni Miltsakaki : Do nlp and machine learning improve traditional readability formulas? *In Proceedings of the First Workshop on Predicting and Improving Text Readability for target reader populations*, pages 49–57, 2012.

[27] Brendan J Frey et Delbert Dueck : Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007.

[28] Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang et Ting Liu : Learning semantic hierarchies via word embeddings. *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1199–1209, 2014.

[29] Jianfeng Gao, Michel Galley, Lihong Li *et al.* : Neural approaches to conversational ai. *Foundations and trends® in information retrieval*, 13(2-3):127–298, 2019.

[30] Asunción Gómez-Pérez : Evaluation of taxonomic knowledge in ontologies and knowledge bases. 1999.

[31] Martin Graham et Jessie Kennedy : A survey of multiple tree visualisation. *Information Visualization*, 9(4):235–252, 2010.

[32] Stanford NLP Group : Large and difficult category taxonomies.

[33] Thomas R Gruber : A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220, 1993.

[34] Thomas R Gruber : Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43(5-6):907–928, 1995.

[35] Nicola Guarino : Some ontological principles for designing upper level lexical resources. *arXiv preprint cmp-lg/9809002*, 1998.

[36] Amit Gupta, Rémi Lebret, Hamza Harkous et Karl Aberer : Taxonomy induction using hypernym subsequences. *In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1329–1338, 2017.

[37] Rajaa El Hamdani, Majd Mustapha, David Restrepo Amariles, Aurore Troussel, Sébastien Meeùs et Katsiaryna Krasnashchok : A combined rule-based and machine learning approach for automated gdpr compliance checking. *In Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, pages 40–49, 2021.

[38] Michael Hanna et David Mareček : Analyzing bert's knowledge of hypernymy via prompting. *In Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 275–282, 2021.

[39] Marti A Hearst : Automatic acquisition of hyponyms from large text corpora. *In COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*, 1992.

[40] Sepp Hochreiter et Jürgen Schmidhuber : Long short-term memory. *Neural Comput.*, 9(8): 1735–1780, nov 1997.

[41] Jiaxin Huang, Yiqing Xie, Yu Meng, Yunyi Zhang et Jiawei Han : Corel: Seed-guided topical taxonomy construction by concept learning and relation transferring. *In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1928–1936, 2020.

[42] Meng Jiang, Jingbo Shang, Taylor Cassidy, Xiang Ren, Lance M Kaplan, Timothy P Hanratty et Jiawei Han : Metapad: Meta pattern discovery from massive text corpora. *In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 877–886, 2017.

[43] David Jurgens et Mohammad Taher Pilehvar : Semeval-2016 task 14: Semantic taxonomy enrichment. *In Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 1092–1102, 2016.

[44] Kornraphop Kawintiranon et Lisa Singh : Knowledge enhanced masked language model for stance detection. *In Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: human language technologies*, 2021.

[45] Martijn G Kersloot, Florentien JP van Putten, Ameen Abu-Hanna, Ronald Cornet et Derk L Arts : Natural language processing algorithms for mapping clinical text fragments onto ontology concepts: a systematic review and recommendations for future studies. *Journal of biomedical semantics*, 11(1):1–21, 2020.

[46] Yuval Kluger, Ronen Basri, Joseph T Chang et Mark Gerstein : Spectral biclustering of microarray data: coclustering genes and conditions. *Genome research*, 13(4):703–716, 2003.

[47] Zornitsa Kozareva et Eduard 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*, pages 1110–1118, 2010.

[48] Zornitsa Kozareva, Ellen Riloff et Eduard Hovy : Semantic class learning from the web with hyponym pattern linkage graphs. *In Proceedings of ACL-08: HLT*, pages 1048–1056, 2008.

[49] Martin Krallinger, Florian Leitner, Miguel Vazquez, David Salgado, Christophe Marcelle, Mike Tyers, Alfonso Valencia et Andrew Chatr-aryamontri : How to link ontologies and protein–protein interactions to literature: text-mining approaches and the biocreative experience. *Database*, 2012, 2012.

[50] Koren Lazar, Benny Saret, Asaf Yehudai, Wayne Horowitz, Nathan Wasserman et Gabriel Stanovsky : Filling the gaps in ancient akkadian texts: A masked language modelling approach. *arXiv preprint arXiv:2109.04513*, 2021.

[51] Claudia Leacock et Martin Chodorow : Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283, 1998.

[52] Dongha Lee, Jiaming Shen, SeongKu Kang, Susik Yoon, Jiawei Han et Hwanjo Yu : Taxocom: Topic taxonomy completion with hierarchical discovery of novel topic clusters. *In Proceedings of the ACM Web Conference 2022*, pages 2819–2829, 2022.

[53] Vladimir I Levenshtein *et al.* : Binary codes capable of correcting deletions, insertions, and reversals. *In Soviet physics doklady*, volume 10, pages 707–710. Soviet Union, 1966.

[54] David A Liem, Sanjana Murali, Dibakar Sigdel, Yu Shi, Xuan Wang, Jiaming Shen, Howard Choi, John H Caufield, Wei Wang, Peipei Ping *et al.* : Phrase mining of textual data to analyze extracellular matrix protein patterns across cardiovascular disease. *American Journal of Physiology-Heart and Circulatory Physiology*, 315(4):H910–H924, 2018.

[55] Dekang Lin *et al.* : An information-theoretic definition of similarity. *In Icml*, volume 98, pages 296–304, 1998.

[56] Xueqing Liu, Yangqiu Song, Shixia Liu et Haixun Wang : Automatic taxonomy construction from keywords. *In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1433–1441, 2012.

[57] Adolfo Lozano-Tello et Asunción Gómez-Pérez : Ontometric: A method to choose the appropriate ontology. *Journal of Database Management (JDM)*, 15(2):1–18, 2004.

[58] Anh Tuan Luu, Jung-jae Kim et See Kiong Ng : Taxonomy construction using syntactic contextual evidence. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 810–819, 2014.

[59] Jianxin Ma, Peng Cui, Xiao Wang et Wenwu Zhu : Hierarchical taxonomy aware network embedding. *In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1920–1929, 2018.

[60] Alexander Maedche et Steffen Staab : Measuring similarity between ontologies. *In International Conference on Knowledge Engineering and Knowledge Management*, pages 251–263. Springer, 2002.

[61] Sourabh Mehta : A tutorial on various clustering evaluation metrics, Mar 2022.

[62] Tomas Mikolov, Kai Chen, Greg Corrado et Jeffrey Dean : Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[63] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado et Jeff Dean : Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

[64] Ndapandula Nakashole, Gerhard Weikum et Fabian 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*, pages 1135–1145, 2012.

[65] Roberto NAVIGLI et Paola VELARDI : Learning word-class lattices for definition and hypernym extraction. *In Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 1318–1327, 2010.

[66] Roberto NAVIGLI, Paola VELARDI et Stefano FARALLI : A graph-based algorithm for inducing lexical taxonomies from scratch. *In Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[67] David NEWMAN, Jey Han LAU, Karl GRIESER et Timothy BALDWIN : Automatic evaluation of topic coherence. *In Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*, pages 100–108, 2010.

[68] Kim Anh NGUYEN, Maximilian KÖPER, Sabine Schulte im WALDE et Ngoc Thang VU : Hierarchical embeddings for hypernymy detection and directionality. *arXiv preprint arXiv:1707.07273*, 2017.

[69] Natalya F NOY, Deborah L MCGUINNESS *et al.* : Ontology development 101: A guide to creating your first ontology, 2001.

[70] Alexander PANCHENKO, Stefano FARALLI, Eugen RUPPERT, Steffen REMUS, Hubert NAETS, Cédrick FAIRON, Simone Paolo PONZETTO et 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-2016)*, pages 1320–1327, 2016.

[71] Kishore PAPINENI, Salim ROUKOS, Todd WARD et Wei-Jing ZHU : Bleu: a method for automatic evaluation of machine translation. *In Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[72] Chintan PATEL, Kaustubh SUPEKAR, Yugyung LEE et EK PARK : Ontokhoj: a semantic web portal for ontology searching, ranking and classification. *In Proceedings of the 5th ACM international workshop on Web information and data management*, pages 58–61, 2003.

[73] Ted PEDERSEN, Siddharth PATWARDHAN, Jason MICHELIZZI *et al.* : Wordnet:: Similarity-measuring the relatedness of concepts. *In AAAI*, volume 4, pages 25–29, 2004.

[74] Bo PENG, Emmanuele CHERSONI, Yu-Yin HSU et Chu-Ren HUANG : Discovering financial hypernyms by prompting masked language models. *In Proceedings of the 4th Financial Narrative Processing Workshop@ LREC2022*, pages 10–16, 2022.

[75] Matthew E. PETERS, Mark NEUMANN, Mohit IYYER, Matt GARDNER, Christopher CLARK, Kenton LEE et Luke ZETTLEMOYER : Deep contextualized word representations, 2018.

[76] Simone Paolo PONZETTO et Michael STRUBE : Taxonomy induction based on a collaboratively built knowledge repository. *Artif. Intell.*, 175(9-10):1737–1756, 2011.

[77] Simone Paolo PONZETTO, Michael STRUBE *et al.* : Deriving a large scale taxonomy from wikipedia. *In AAAI*, volume 7, pages 1440–1445, 2007.

[78] Robert PORZEL et Rainer MALAKA : A task-based approach for ontology evaluation. *In ECAI Workshop on Ontology Learning and Population, Valencia, Spain*, pages 1–6. Citeseer, 2004.

[79] Alec RADFORD, Jeffrey WU, Rewon CHILD, David LUAN, Dario AMODEI, Ilya SUTSKEVER *et al.* : Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[80] Philip RESNIK : Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*, 1995.

[81] Alan RITTER, Stephen SODERLAND et Oren ETZIONI : What is this, anyway: Automatic hypernym discovery. *In AAAI Spring Symposium: Learning by Reading and Learning to Read*, pages 88–93, 2009.

[82] Stephen ROLLER, Douwe KIELA et Maximilian NICKEL : Hearst patterns revisited: Automatic hypernym detection from large text corpora. *arXiv preprint arXiv:1806.03191*, 2018.

[83] Stuart ROSE, Dave ENGEL, Nick CRAMER et Wendy COWLEY : Automatic keyword extraction from individual documents. *Text mining: applications and theory*, 1(1-20):10–1002, 2010.

[84] Peter J ROUSSEEUW : Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[85] Lilliana SANCHO-CHAVARRIA, Fabian BECK et Erick MATA-MONTERO : An expert study on hierarchy comparison methods applied to biological taxonomies curation. *PeerJ Computer Science*, 6:e277, 2020.

[86] Jingbo SHANG, Jialu LIU, Meng JIANG, Xiang REN, Clare R VOSS et Jiawei HAN : Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge and Data Engineering*, 30(10):1825–1837, 2018.

[87] Jingbo SHANG, Xinyang ZHANG, Liyuan LIU, Sha LI et Jiawei HAN : Nettaxo: Automated topic taxonomy construction from text-rich network. *In Proceedings of The Web Conference 2020*, pages 1908–1919, 2020.

[88] Jiaming SHEN : Jmshen1994/awesome-taxonomy: A curated resource for taxonomy research.

[89] Jiaming SHEN, Zhihong SHEN, Chenyan XIONG, Chi WANG, Kuansan WANG et Jiawei HAN : Taxoexpan: Self-supervised taxonomy expansion with position-enhanced graph neural network. *In Proceedings of The Web Conference 2020*, pages 486–497, 2020.

[90] Jiaming SHEN, Zeqiu WU, Dongming LEI, Chao ZHANG, Xiang REN, Michelle T. VANNI, Brian M. SADLER et Jiawei HAN : Hiexpan: Task-guided taxonomy construction by hierarchical tree expansion. *CoRR*, abs/1910.08194, 2019.

[91] Yu SHI, Jiaming SHEN, Yuchen LI, Naijing ZHANG, Xinwei HE, Zhengzhi LOU, Qi ZHU, Matthew WALKER, Myunghwan KIM et Jiawei HAN : Discovering hypernymy in text-rich heterogeneous information network by exploiting context granularity. *In Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 599–608, 2019.

[92] Vered SHWARTZ, Yoav GOLDBERG et Ido DAGAN : Improving hypernymy detection with an integrated path-based and distributional method. *arXiv preprint arXiv:1603.06076*, 2016.

[93] Koustuv SINHA, Robin JIA, Dieuwke HUPKES, Joelle PINEAU, Adina WILLIAMS et Douwe KIELA : Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. *arXiv preprint arXiv:2104.06644*, 2021.

[94] Rion SNOW, Dan JURAFSKY et Andrew Y NG : Semantic taxonomy induction from heterogenous evidence. *In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 801–808, 2006.

[95] Steffen STAAB et Rudi STUDER : *Handbook on ontologies*. Springer Science & Business Media, 2010.

[96] Fabian M SUCHANEK, Gjergji KASNECI et Gerhard WEIKUM : Yago: a core of semantic knowledge. *In Proceedings of the 16th international conference on World Wide Web*, pages 697–706, 2007.

[97] Yu SUN, Shuohuan WANG, Yukun LI, Shikun FENG, Xuyi CHEN, Han ZHANG, Xin TIAN, Danxiang ZHU, Hao TIAN et Hua WU : Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*, 2019.

[98] Kaveh TAGHIPOUR et Hwee Tou NG : A neural approach to automated essay scoring. *In Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1882–1891, 2016.

[99] Wilson L TAYLOR : “cloze procedure”: A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433, 1953.

[100] Pucktada TREERATPITUK, Madian KHABSA et C Lee GILES : Graph-based approach to automatic taxonomy generation (grabtax). *arXiv preprint arXiv:1307.1718*, 2013.

[101] Laurens Van der MAATEN et Geoffrey HINTON : Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[102] Ashish VASWANI, Noam SHAZEER, Niki PARMAR, Jakob USZKOREIT, Llion JONES, Aidan N GOMEZ, Łukasz KAISER et Illia POLOSUKHIN : Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[103] Johanna VÖLKER, Denny VRANDEČIĆ et York SURE : Automatic evaluation of ontologies (aeon). *In International Semantic Web Conference*, pages 716–731. Springer, 2005.

[104] Denny VRANDECIC : *Ontology evaluation.* Thèse de doctorat, Karlsruhe Institute of Technology, 2010.

[105] Alex WANG, Amanpreet SINGH, Julian MICHAEL, Felix HILL, Omer LEVY et Samuel R BOWMAN : Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[106] Chi WANG, Marina DANILEVSKY, Nihit DESAI, Yinan ZHANG, Phuong NGUYEN, Thrivikrama TAULA et Jiawei HAN : A phrase mining framework for recursive construction of a topical hierarchy. *In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 437–445, 2013.

[107] Jason WEI et Kai ZOU : Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*, 2019.

[108] William E WINKLER : Overview of record linkage and current research directions. *In Bureau of the Census*. Citeseer, 2006.

[109] Wentao WU, Hongsong LI, Haixun WANG et Kenny Q ZHU : Probase: A probabilistic taxonomy for text understanding. *In Proceedings of the 2012 ACM SIGMOD international conference on management of data*, pages 481–492, 2012.

[110] Zhibiao WU et Martha PALMER : Verb semantics and lexical selection. *arXiv preprint cmp-lg/9406033*, 1994.

[111] Jingyi XU, Zilu ZHANG, Tal FRIEDMAN, Yitao LIANG et Guy Van den BROECK : A semantic loss function for deep learning with symbolic knowledge. *In* Jennifer DY et Andreas KRAUSE, éditeurs : *Proceedings of the 35th International Conference on Machine Learning*, volume 80 de *Proceedings of Machine Learning Research*, pages 5502–5511. PMLR, 10–15 Jul 2018.

[112] Hui YANG et Jamie CALLAN : A metric-based framework for automatic taxonomy induction. *In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 271–279, 2009.

[113] Yi YANG, Mark Christopher Siy UY et Allen HUANG : Finbert: A pretrained language model for financial communications. *arXiv preprint arXiv:2006.08097*, 2020.

[114] Zheng YU, Haixun WANG, Xuemin LIN et Min WANG : Learning term embeddings for hypernymy identification. *In Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[115] Chao ZHANG, Fangbo TAO, Xiusi CHEN, Jiaming SHEN, Meng JIANG, Brian SADLER, Michelle VANNI et Jiawei HAN : Taxogen: Unsupervised topic taxonomy construction by adaptive term embedding and clustering. *In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2701–2709, 2018.

[116] Jieyu ZHANG, Xiangchen SONG, Ying ZENG, Jiaze CHEN, Jiaming SHEN, Yuning MAO et Lei LI : Taxonomy completion via triplet matching network. *In Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4662–4670, 2021.

[117] Kaizhong ZHANG et Dennis SHASHA : Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing*, 18(6):1245–1262, 1989.

[118] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger et Yoav Artzi : Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

[119] Yuchen Zhang, Amr Ahmed, Vanja Josifovski et Alexander Smola : Taxonomy discovery for personalized recommendation. *In Proceedings of the 7th ACM international conference on Web search and data mining*, pages 243–252, 2014.

[120] Cai-Nicolas Ziegler, Georg Lausen et Lars Schmidt-Thieme : Taxonomy-driven computation of product recommendations. *In Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 406–415, 2004.

# Appendix A

# MLM Proxy Evaluation Targets

| Topic | Subtopic Clusters |
|---|---|
| Steak | rib_steak sirloin_steak tenderloin porterhouse flank_steak sirloin rib_eye cooked_medium ribeye rib med_rare cooked_medium_rare medium_rare filet ribeye_steak striploin bone boneless filet_mignon ... |
| | tomatos cucumbers green_peppers tomatoes romaine_lettuce diced peppers jalape_os lettuce banana_peppers bell_peppers red_peppers onions bell_pepper jalapenos shredded_lettuce shredded ... |
| | ...... |
| Veggies | artichokes red_peppers olives green_peppers bell_peppers artichoke tomatoes peppers fresh_mozzarella kalamata roasted_peppers jalape_os parsley bell_pepper italian_sausage fresh_basil mozzarella ... |
| | peanut coconut_milk soy coconut cashew soy_milk milk cashews coconut_cream taro durian winter_melon lychee black_tea tapioca pandan chai sweetened green_tea cottage_cheese almond earl_grey sago ... |
| | ...... |
| Dessert | sticky_rice ginger papaya mango panang_curry coconut tom_yum_soup penang tamarind satay lemongrass panang peanut_sauce drunken_noodles cashew pad_thai lettuce_wraps larb thai_basil crunch_roll ... |
| | brownie cookie peanut_butter cookie_dough vanilla cone ice_cream sundae fudge cupcake chocolate_syrup chocolate_chip_cookie reese marshmallow hot_fudge oreo_cookie brownie_sundae cookie_monster caramel ... |
| | ...... |
| ... | ...... |

**Table A.1.** CoRel1 sampled clusters

| Topic | Subtopic Clusters |
|---|---|
| Shrimp | satay prawns ginger dumpling broth. noodles, marinated skewers steamed lamb dumplings squid duck, noodles. noodles teriyaki tendon vegetable clams hamachi broth pea curry, miso stir_fried noodle ...<br>shredded_beef enchiladas tamale tortillas guac guacamole carne_asada salsa tortillas. salsa. taco burritos burrito beans tacos refried_beans Salsa burrito chips guacamole guac carnitas burrito ...<br>...... |
| Burger | hot_sauce grilled_onions jalapenos flour jalapeno batter chipotle tortilla wheat tortillas, tortillas pico_de_gallo cilantro, burrito. baja habanero avocados tortilla, cilantro tortilla. refried_beans ...<br>peanut coconut_milk soy coconut cashew soy_milk milk cashews coconut_cream taro durian winter_melon lychee black_tea tapioca pandan chai sweetened green_tea cottage_cheese almond earl_grey sago ...<br>...... |
| Dessert | mushrooms flatbread asparagus spinach eggplant grilled onions goat_cheese artichoke ricotta spaghetti mushroom tomato meatballs meatballs, pasta pesto rigatoni prosciutto artichokes tomatoes alfredo ...<br>passion_fruit lychee green_tea mint mango peach coconut pineapple ginger fruity lemonade lemonade tea. tequila rum vodka, strong. mojito jasmine margarita. lime tea boba martini, pomegranate mango ...<br>...... |
| ... | ...... |

**Table A.2.** CoRel2 sampled clusters

| Topic | Subtopic Clusters |
|---|---|
| Chinese | rice_noodles cabbage steamed bean_sprouts pork shredded vegetables sauteed sauce tofu chicken. peas breaded sesame spicy_sauce shrimp. pepper black_pepper white_rice broccoli cabbage, pan_fried pork diced pea stir_fried ginger sautu00e9ed pickled tofu. wasabi white_rice. sauce, carrots. broth. peanut_sauce vegetables veggies veggies. noodles. carrots rice. beef noodles, shrimp, squid miso, vegetable minced tofu, |
| Seafood | prawns king_crab crabs crab_cakes crab oysters scallops mussels lobster_tail legs prime_rib jumbo sea_bass lobster mussels. soft_shell_crab clams, chops kobe oysters, clams claws crawfish scallop ...<br>pea peel quail leaf orange, shaved leaves. root mango, zucchini, ginger, color. prosciutto, oil, lump herbs, bell_pepper fragrant caviar flakes pieces, thinly_sliced sliced sharp bits colored celery ...<br>...... |
| Dessert | white_chocolate brownie whipped_cream blueberry apples marshmallow chocolate_chip peanut_butter chocolate. mouse cookie pecan cinnamon dark_chocolate salted_caramel red_velvet peanut_butter ...<br>mochi soft_serve frozen_yogurt green_tea fruits yogurt froyo shaved_ice boba soy self-serve boba. markets snow smoothies yogurt. Boba ramen smoothie Smoothie Sprouts places, vanilla, Snow Sephora ...<br>...... |
| ... | ...... |

**Table A.3.** CoRel3 sampled clusters

| Topic | Subtopic Clusters |
|---|---|
| Location | arena venue theater balcony stadium stage theatre target dance comedy stage. show, Beatles performers stage, viewing dancers screens movie songs, acrobatics tv outdoor_area theater, acrobatic DJ dance ...<br>strip_mall strip_mall, plaza shopping_center tucked strip_mall. unassuming shopping_center. hidden shopping_center, hole-in-the-wall corner hidden_gem corner. hole located Located divey gem dive ...<br>... |
| Service | John assistant specialist detailed pressure tint technician solution thorough technique paint quote Rob tires. extensions records tires, repair Mark auto James consult repairs Alex removal repair ...<br>extremely_rude acting male rude, garbage unfriendly lousy unprofessional disrespectful rude. acted rude ignorant miserable yelled clueless snotty blatantly offended manners attitude, unprofessional ...<br>...... |
| Food | chinese noodle Chinese dim_sum asian Vietnamese Asian Korean korean Filipino vietnamese Hawaiian Mongolian Japanese ramen Chinese_food Pho Pho, Taiwanese Thai noodles, thai stir_fry Cantonese Ramen...<br>hash_browns pancakes eggs biscuits scrambled_eggs omelet french_toast eggs_benedict pancakes. corned_beef_hash hashbrowns toast hash_browns. French_toast eggs. potatoes eggs, hash scramble gravy...<br>...... |
| ... | ...... |

**Table A.4.** CoRel4 sampled clusters

| Topic | Subtopic Clusters |
|---|---|
| Seafood | mussel, clam, lobster, oyster, shrimp, scallop, crab, chicken, salmon, fish, pork, octopus, beef, steak, meat, tofu, duck, avocado, sausage, rice, brisket, crawfish, pork_belly, prime_rib |
| Salad | wedge_salad, cesar_salad, mixed_green, sandwich, caesar, caesar_salad, volcano, hamburger, bloody_mary, kale, pre_package, mediterranean, basic, mixed_greens, apple, greek, house, seaweed, grill_chicken, green... |
| Dessert | ice_cream, cake, pastries, sauce, gelato, latte, cheesecake, milkshake, cupcake, taste, cream, smoothie, pancake, waffle, mousse, bread_pudding, muffin, french_toast, syrup, butter, lemonade, croissant, sorbet... |
| Appetizer | fry_calamari, stuff_mushroom, fry_zucchini, fry, shaker, squid, waiter_waitress, curly_fry, pineapple, smash_potato, acoustic, apple_butter, portion_size, lettuce, halibut, carrot, jalapeno_popper, lamb_chop... |
| Food | food_quality, no_real_complaint, table_side_guac, starbucks, no_complaint, 4_._5_star, 4_star, 3_star, 5_star, four_star, five_star, three_star, 1_star, no_worry, no_big_deal, zero_complaint, no_problem ... |
| Sushi | sashimi, spicy_tuna, yellow_tail, spicy_salmon, tuna, california, rainbow, dragon, dynamite, spider, eel, tempura, philadelphia, ahi_tuna, vegas, tiger, las_vegas, cinnamon, spicy_scallop, cucumber, soft_shell_crab, hawaiian, cali |
| ... | ... |
| "Desert" | creme_brulee, cannoli, tiramisu, flan, chocolate, chocolate_cake, baklava, pastry, strawberry, custard, cassoulet, ratatouille, panna_cotta, biscotti, torte, creme_brule, key_lime, pecan_pie, flourless_chocolate_cake, meringue... |
| Music | band, background_music, r_b, k_pop, dj, electronic_music, ghetto, oldie, music_video, beatle, loud_music, rock, techno_music, hawaiian_music, tiesto, trumpet, smooth_jazz, big_screen_tv, christmas_music, indie_music... |

**Table A.5.** HiExpan1 sampled topics

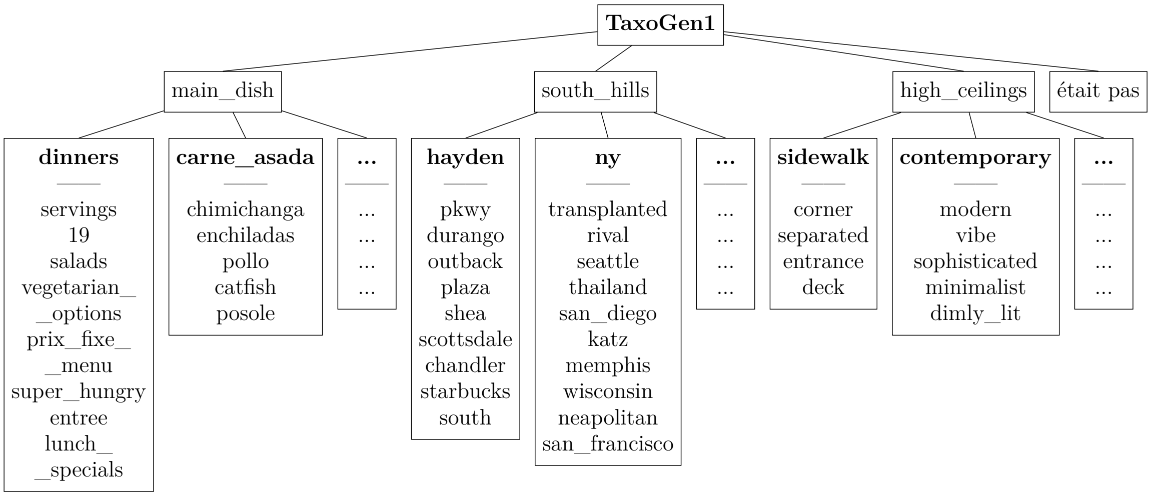


**Fig. A.1.** TaxoGen1 sampled topics

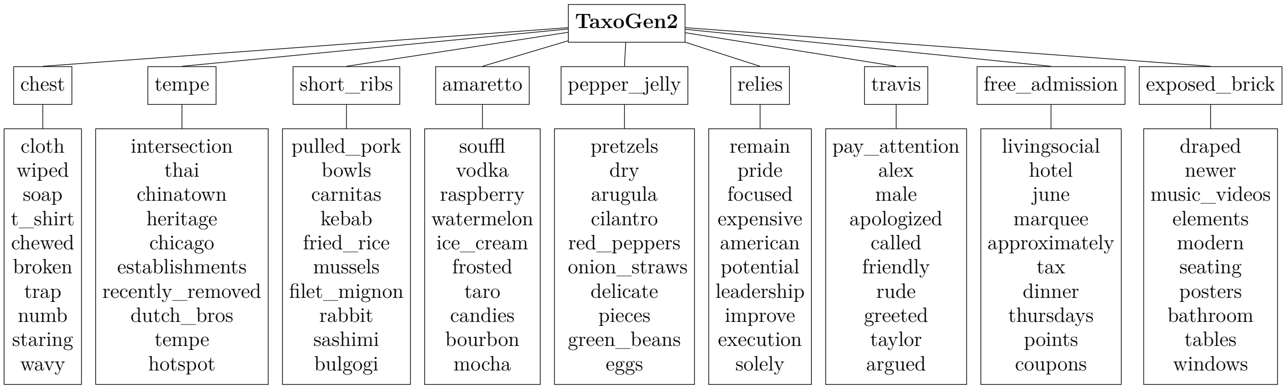


**Fig. A.2.** TaxoGen2 topics

## A.1. Evaluation Terms Added to the BERT Vocabulary

```
"sushi",
"carne asada",
"halibut",
"exposed brick",
"wine bar",
"visiting family",
"main dish",
"veggie",
"mont royal",
"high ceiling",
"pepper jelly",
"appetizer",
"short rib",
"amaretto",
"lounging",
"lasagna",
"crispy",
"italian place",
"chow mein",
"sashimi",
"warm tone",
"korma",
"panang",
"comfortable bed",
"buffalo chicken wrap",
"high expectation",
"free admission",
"south hill",
"tempe",
"babaganoush"
```