Towards a Knowledge-Aware AI
A. Dimou et al. (Eds.)

doi:10.3233/SSW220018

Learning Ontology Classes from Text by Clustering Lexical Substitutes Derived from Language Models [1]

Artem REVENKO [a] Victor MIRELES [a] Anna BREIT [a] Peter BOURGONJE [b]
Julian MORENO-SCHNEIDER [c] Maria KHVALCHIK [a] Georg REHM [c]

[a] *Semantic Web Company GmbH, Austria.*
{firstname}.{second.name}@semantic-web.com
[b] *Morningsun Technology GmbH, Germany.*
peter.bourgonje@morningsun-technology.com
[c] *DFKI GmbH, Germany. {firstname}.{second.name}@dfki.de*

Abstract.
Many tools for knowledge management and the Semantic Web presuppose the existence of an arrangement of instances into classes, i. e. an ontology. Creating such an ontology, however, is a labor-intensive task. We present an unsupervised method to learn an ontology from text. We rely on pre-trained language models to generate lexical substitutes of given entities and then use matrix factorization to induce new classes and their entities. Our method differs from previous approaches in that (1) it captures the polysemy of entities; (2) it produces interpretable labels of the induced classes; (3) it does not require any particular structure of the text; (4) no re-training is required. We evaluate our method on German and English WikiNER corpora and demonstrate the improvements over state of the art approaches.

Keywords. Ontology Learning, Knowledge Discovery, Language Models

1. Introduction

The assignment of entities into a hierarchy of semantically coherent classes is the basis of knowledge organization systems, and is useful for many information and text processing tasks. Such classifications are usually created manually (a labour intensive task), but can also be identified in semi-automatic ways from a corpus. Specifically, the ontology learning task dealt with in this paper, seeks to create the class hierarchy *de novo* from a corpus, along with an assignment of entities into the induced classes. This approach constitutes a translation of the distributional semantics captured in corpus-wide statistics, into the explicit semantics described in the class hierarchy of an ontology.

Making corpus-based ontology learning effective on small domain-specific corpora, enables small organizations to tackle specific problems in reduced times. The resulting

[1] The work presented in this article has received funding from the German Federal Ministry for Economic Affairs and Climate Action (BMWK) through the project SPEAKER (no. 01MK19011), and the Austrian Research Promotion Agency (FFG) through the Project OBARIS (Grant Agreement No 877389)

ontologies can be useful for creating data models or powering search applications, among myriad other applications. In particular, the use of domain-specific ontologies in enabling knowledge-based transfer learning in information extraction systems (e.g., [14,27]) is a promising method for industrial applications. For these and other applications, ontology learning has been approached from different angles, as reviewed in Section 2.

The assignment of entities into semantically coherent classes relies on some method for recognizing these in text, such as Named Entity Recognition (NER) or Entity Linking (EL) tools, which have a variety of associated costs. Therefore, an effective corpus-based class induction method should be *able to work with the output of any annotation tool*. In turn, since many of these methods are not able to *handle polysemous words*[2], ontology learning also must include some means of disambiguation. Another important consideration is that any subdivision of a corpus into documents or similar structures might not necessarily follow the semantic categorization of entities. That is, the assumption –which e.g., topic modelling approaches build on– that semantically similar concepts are often co-occurring across documents, induces already a notion of semantic similarity which might not correspond to the task at hand or the different uses of entities by the authors of the corpus. For this reason, ontology learning methods that *don't make use of any notion of document* have a larger range of applications. Finally, since ontologies are meant to be consumed not only by machines, but rather help inform human-centered knowledge managing, it is desirable that any automatically identified classes be *interpretable by humans*, for example, by being accompanied by a natural language description.

In this work, we present a method for learning ontology classes that leverages large pre-trained language models, in order to reduce the amount of training data required and make it applicable to corpora of different sizes. We use lexical substitutes derived from the language models to capture representations of annotated entities in context. By analyzing the substitutes of different entities in different contexts we identify and cluster the different contextualized usages of entities, and propose a class hierarchy for them. By clustering contextualized usages, as opposed to entities themselves, the system also disambiguates between different senses of an entity, in particular between general and domain-specific ones. The details of the method are presented in Section 3. Finally, the learned classes are assigned descriptors, which can aid a human in distinguishing them and, eventually, assigning a label to each. Our proposal is compared to state of the art approaches to the same task in Section 4.3, according to evaluation criteria presented in Section 4.

2. Related Work

Ontology Learning Ontology learning is the process of deriving an ontology from natural language or structured data [16,4,8]. In general ontology learning includes many tasks such as identifying terms, grouping them into classes, extracting hierarchical (taxonomical) and non-hierarchical relations between classes, and discovering more complex axioms. In this work we aim at grouping the entities (terms) into a hierarchy of classes, and moreover we assume that entities are already identified and annotated in the corpus. Our tasks correspond to the third and fourth layer of the Ontology Learning Layer Cake

[2]For example, "Apple" as fruit or as a brand name.

[8]. These tasks are especially important as the class hierarchy defines the backbone of an ontology [1]. Common approaches combine linguistic features with statistics and machine learning [11,13,29,23]. These methods often have low recall and are affected by noise [7]. Also such methods often assume significant human intervention and are language-specific.

Modern end-to-end deep learning models have a chance to overcome these limitations. First, it is not necessary to provide explicit features to such systems. Second, intrinsic language understanding might overcome noise in the data. In [9,2] authors exploit static (not contextualized) word embeddings to extract ontologies. However, to the best of our knowledge (and see also [1,16]) none of the existing ontology learning approaches exploits lexical substitutes produced by pre-trained, deep learning-based, Language Models (LM) to identify classes of entities.

Induction of Topic Taxonomies A closely related field is induction of topic taxonomies [35]. Topic taxonomies can be defined as lightweight ontologies that only include class hierarchies and assertions of instances to classes. Hence, this task exactly matches the task we solve in this work. The creation of topic taxonomies has often been approached by means of clustering methods, interpreting the resulting clusters as topics. In order to use common clustering methods, a mapping from terms to vectors is required, for example by the use of word embeddings pre-trained on large corpora such as word2vec [22,33]. These approaches, however, fail to capture the idiosyncratic use of terms in a given domain[3], and so context-specific embeddings have been proposed. In this respect, contextualised embeddings, such as those underlying recent LMs, have been exploited to produce vectors that are to be fed into statistical classifiers to detect is-a relations [10,18], as well as more general relation extraction [31]. Unfortunately, the use of the LMs in existing work does not properly capture the polysemy of terms and studies have shown that the only is-a relations found are those which the model acquired during training [20]. A more careful use of contextualised embeddings can be found in TaxoGen [35] and its derivatives [30], however, its usability is limited by the need of retraining embeddings on specific sub-corpora, and sense disambiguation is not explicitly handled.

To the best of our knowledge, none of the existing methods tackles the polysemy of entities. The polysemy of terms is especially important in domain-specific corpora, as it is common for words to be adopted and given new senses in each domain. However, retention of the original sense is not uncommon, and constitutes a challenge which might degrade the quality of the resulting ontology. Often these models rely on the notion of document as a single coherent piece of text, which itself induces a notion of semantic similarity which might be more related to the process of producing and editing the corpus, than to the meanings of entities themselves. The method we introduce in this paper does not require partitioning of the corpus into documents.

Topic Modeling Topic modeling is a frequently used text-mining tool for discovery of hidden semantic structures in a text body. The goal of topic modeling is to cluster a corpus of documents into thematically coherent groups of documents and keywords [32]. These clusters of keywords could be used to produce new classes of an ontology. Therefore, this method is compatible with our task and we will use the results for comparison.

[3]For example, the action of the verb "to host" is applied on software or services in the field of Computer Science, but in the field of product reviews, it is usually applied on people.

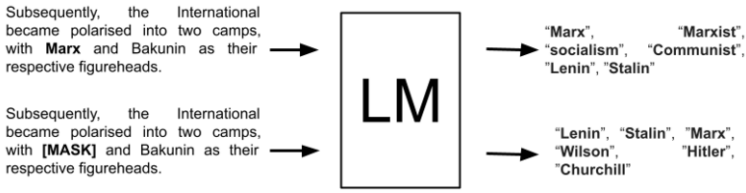


Figure 1. Top $m = 6$ substitutes predicted by a language model for an unmodified (top) and masked (bottom) context of "**Marx**".

However, the initial goal is to discover latent topics in the corpus, therefore the grouping of the terms relies more on the distribution of words across documents rather than specific patterns of the usage of words.

Language Modeling In our experiments, we exploit modern language modeling techniques. Recently, neural network architectures, in particular the Transformer architecture, have progressed the state-of-the-art in many benchmark Natural Language Processing (NLP) tasks [25,26,12]. The pre-training usually happens on Wikipedia or on news articles (due to their wide availability), though domain-specific incarnations also exist, e. g., BioBERT [17], SciBERT [6] and ClinicalBERT [15]. One essential feature of language models used in this work, is their ability to predict a word or sequence which has been masked in a text. For this particular task, the contextualized representation inherent in modern language models yields better results[25,26] than earlier vector representation-based language models [21], and bidirectionality can further improve these results [12]. These predictions, can be interpreted as lexical substitutes of the masked section.

3. Method

We start from a corpus which has been annotated with entities, for example using an NER tool, an entity linking tool or a gazetteer. Based on these entity annotations, we set out to induce a classification of the entities and produce a set of interpretable descriptors for each class. This is done in a three-step process: first we generate lexical substitutes for the entity in context, second, we induce sense representations of entities, and finally, we group these senses into classes. See Figure 2 for a graphical summary of the procedure.

Create Substitutes We consider a set of entity annotations, each with a context c consisting of a window of w words before and after the entity mention. For each of these annotations, we generate two inputs that are fed into a language model: the original unmodified context c, and the context c_{masked} in which the entity mention has been masked (see Figure 1 for an example). For each of these inputs, we obtain the top m substitutes, where m allows us to balance between a low number of high quality substitutes and high number of potentially lower quality substitutes.

Extract Senses Next, we generate a set of binary matrices $\{M^e\}$, one for every entity. Each of the matrices M^e has one row per context in which the entity e is mentioned, and one column per substitute suggested by the language model. Thus, the entry $M_{i,j}$ is 1 if and only if the language model predicts substitute j as one of the m best ranked substitutes for any of the contexts c_i, c_{i_masked}. In total, we obtain as many binary matrices as unique entities can be found in the corpus.

It is possible that a given entity is used in more than one sense throughout the corpus. In order to identify those senses, we factorize each of the binary matrices M^e using Algorithm 2 from [5], in a fashion that has also been used for word sense induction (e.g., [3]). This algorithm outputs a set $S^e = \{s_1, s_2, \ldots\}$ of factors, which we call *senses*. Each sense s consists of a set of contexts and a set of substitutes D_s, such that for each context, the annotated entity can be substituted by any of the substitutes from D_s. We call D_s the *sense descriptors* of s. We consider only at most k descriptors for each sense. Similar to m, higher values of k produce larger sense descriptions. Finally, heuristically we identified that if an entity appears less than five times it is unlikely that more than a single sense would be induced. Therefore, for such infrequent entities, a single representative cluster is produced by taking the most common substitutes for the entity.

Induce Classes Once we have produced a set of senses, we proceed to cluster these in order to induce classes. For this, we generate a second binary matrix $\mathscr{M}$ whose rows correspond to all senses of all entities, and whose columns correspond to all descriptors of all senses, and factorize it using the same method from [5]. The result of this factorization is a set of tuples of the form $C = (E, D)$ where E is a set of entity senses and D a set of entity descriptors. Each of these tuples represents a class, where E is a set of entity senses belonging to it, and the descriptors in D provide and interpretable representation of the class. To enforce longer class descriptions, we introduce a new parameter th and filter out clusters with less than th descriptors. Examples of the resulting classes can be seen in Examples 2 (English) and 3, 4 (German).

Note that the maximum possible size of the matrix $\mathscr{M}$ is $N_s \times (k \times N_s)$, with N_s being the total number of senses for all entities. In practice we observe smaller values for the second dimension, as descriptors for different senses overlap; which also indicates that we can expect better results for the entire procedure as many different senses share substitutes and could be efficiently grouped.

We perform the matrix factorisation twice: First for each entity separately and then for the obtained senses and their descriptors. One could skip the first factorisation and just collect all *occurrences* of entities and their respective substitutes into a single binary matrix. However, the distribution of entities is far from normal, so the over-represented ones would dominate such a matrix. In preliminary experiments we have observed that this leads to the discovery of various sub-senses of the popular entities rather than a meaningful grouping of different entities. To favor the semantic grouping of various senses we deem double factorisation necessary.

3.1. Hierarchies of Classes

It is possible to generalize the introduced method to induce hierarchies of classes. Namely, we see two possibilities to reveal hierarchies, whose evaluation is left outside the scope of this paper:

Iterative application of the method. Given a class of interest we apply the method to entities of this class. The induced classes are sub-classes of the initially given class.

Control the granularity of the class with th. Larger numbers of th produce more specific classes, smaller values of th produce more general classes with more entities. Comparing the classification with different th allows to extract hierarchies between those classes. For an example see EN-11-th3 and EN-9-th6 in Example 1 and 2.

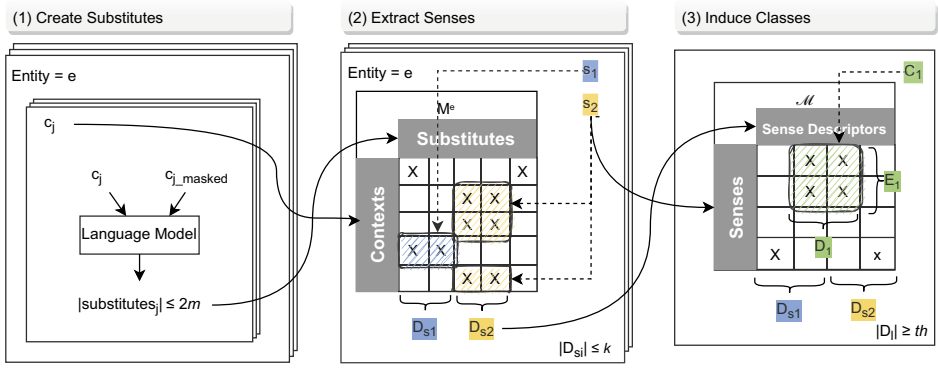


Figure 2. Class induction diagram. c_j represents the jth context. s_i is the ith induced sense; D_{s_i} are the descriptors of s_i. C_l is one of the induced classes; D_l are the descriptors of C_l, and E_l are the entity senses belonging to C_l. k, m and th are hyperparameters.

4. Experiments

4.1. Experimental Setup

We showcase the method presented here by applying it to two corpora in which named entities have been annotated. We use the first 120k tokens (including approx. 10k entities) from both the English and the German sections of the WikiNER data set [24]. While both of these corpora have been annotated for entities and their type, we ignore the specific type of each annotation/entity (except for the evaluation). In total, the German corpus contains 15,207 occurrences of 10,478 unique entities, and the English section contains 10,273 occurrences of 4,485 unique entities.

We choose the WikiNER dataset because class induction on it can be evaluated using the first method described above, using the original NER types as categories. To make evaluation by the second method possible, we link the annotated entities to Wikidata. Entity Linking (EL) was performed before executing class induction, in order to normalise the different surface forms that a putative entity can have. The linking was done using Entity Fishing[4], and resulted in a modest amount of normalisation, as shown in Table 1. One side effect of Entity Linking with modern tools is that senses are potentially disambiguated, but we observed this happening in only a small number of instances (see Table 1).

In the reported experiments, we use DistilBERT [28] as Language Model, using the HuggingFace implementation [34]. Our implementation is available online[5].

4.2. Evaluation Setup

Given a set of entities, finding the best assignment of them into classes is not a well defined problem. In general, several of the criteria to consider when the assignment is made are independent of any corpus and are more related to the final task the categorization is

[4] https://github.com/kermitt2/entity-fishing/ visited on April 18, 2021.
[5] https://github.com/semantic-web-company/ptlm_wsid

to help solve. Since the candidate classes produced by the method presented here are not specific to any downstream task, we consider three different methods to evaluate their quality, all of which are also task-agnostic.

The first evaluation method is based on the manual NE annotations originating from the used corpora. The associated entity types are relatively coarse-grained, covering *Persons, Locations, Organizations*, and *Other*.

The second evaluation method compares the candidate classes derived from the corpus by out class induction approach with other large, task-agnostic and crowd-sourced ontologies, which are manually curated but not derived from any particular corpus. This comparison is made on two assumptions: (i) said pre-existing ontology represents the collective understanding of the entities' *meaning* which is consistent with that contained in almost any corpus, and (ii) the corpus which the method was executed on is a representative sample of a putative universal corpus that informs the creation and maintenance of the preexisting ontology. Such an ontology is the Wikidata class structure, as represented by predicates `P:31` and `P:279` (*instance of*, and *subclass of*, respectively).

The third evaluation method is purely a qualitative one, in which the candidate classes resulting from the method presented here are inspected and commented upon. This method, while not capable of giving any numerical measure, does take into account several sources of knowledge (as summarized in the background knowledge of the human commentator) and gives a more complete interpretation of the results.

The first and second methods both allow for a numerical value to be assigned to any set of candidate classes produced by the method presented here, or by any other producing groupings of entities. In order to aid the explanation of the computation of this value in these cases, in the following we refer to both Wikidata classes, and to NER types, as categories, and we assume that the entities present in the corpus can be linked to members of said categories (in the experiments performed, this assumption holds most of the time, as detailed in Table 1). The quality of the match is quantified using a well-known enrichment analysis method [19]. For every candidate class C and every possibly matching category K, *enrichment* is the probability of a randomly chosen candidate class of the same size as C to contain as many entities of K as C does. If we define $N(K,C)$ as the number of entities in candidate class C that also belong to category K, enrichment is computed using a binomial test according to:

$$P(K,C) = \sum_{k=N(K,C)}^{|C|} \binom{|C|}{k} P(K)^k (1-P(K))^{|C|-k} \tag{1}$$

where $P(K) = \frac{|K|}{|E|}$, and E is the set of all entities in the corpus. Since we compute such probability for several categories K, we account for multiple testing by using Bonferroni correction (i.e. dividing by the number of different categories K that contain at least one entity in common with C).

Using the resulting p-value, we can compute the percentage of candidate classes which are significantly enriched for a category of each of the knowledge sources. In brief, this number tells us how many of the classes suggested by our method are linked to entities contained in one of the categories of the knowledge source, compensating for the overall distribution of the entities in the corpus across the categories. We now present a parameter exploration evaluated using the first and second methods, as well as a quantitative analysis of the results with several combinations of parameters.

4.3. Quantitative Analysis

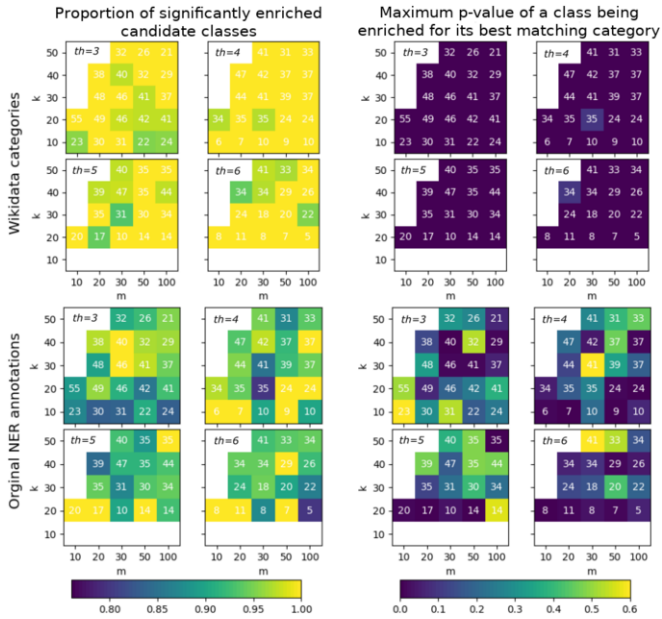


Figure 3. Quantitative evaluation of English candidate classes. Left shows the proportion of candidate classes which are significantly enriched (p-value < 0.05 according to Eq. 1 after Bonferroni correction) for at least one Wikidata category (top) or of the original NER types the dataset was annotated with (bottom); higher is better. Right shows the maximum p-value of this enrichment, lower is better. Shown in white is the number of candidate classes produced with each combination of parameters.

Using the first two evaluation methods presented earlier, we evaluate the behaviour of our method using different combinations of hyperparameters m, k and th. The number of senses produced in the second step of our method (factorisation of the first binary matrices) is dependent on m, but does not show much variance: for English we produce at most 45 polysemous entities with $m = 20$ and at least 33 with $m = 10$; for German – at most 30 polysemous entities with $m = 30$ and at least 23 with $m = 20$. Many polysemous entities only appear a few times and it is expected that our method will not induce different senses for such infrequent entities.

The results of the parameter exploration are shown in Figure 3 and Figure 4. For both the German and English corpus, most combinations of parameters lead to a good amount of candidate classes which are significantly enriched in Wikidata categories. For the English corpus, the candidate classes are not always fully contained within the original NER types, although at least 80% are, for all but two parameter combinations, see

Normalisation. Number of URIs with N entities							Disambiguation. Number of entities with N URIs				
N	**1**	**2**	**3**	**4**	**≥5**		**1**	**2**	**3**	**4**	**≥5**
English	3,972	320	40	10	13		4,099	306	37	6	1
German	9,566	397	55	23	12		10,111	286	7	0	1

Table 1. Normalisation and Disambiguation by linking to Wikidata.

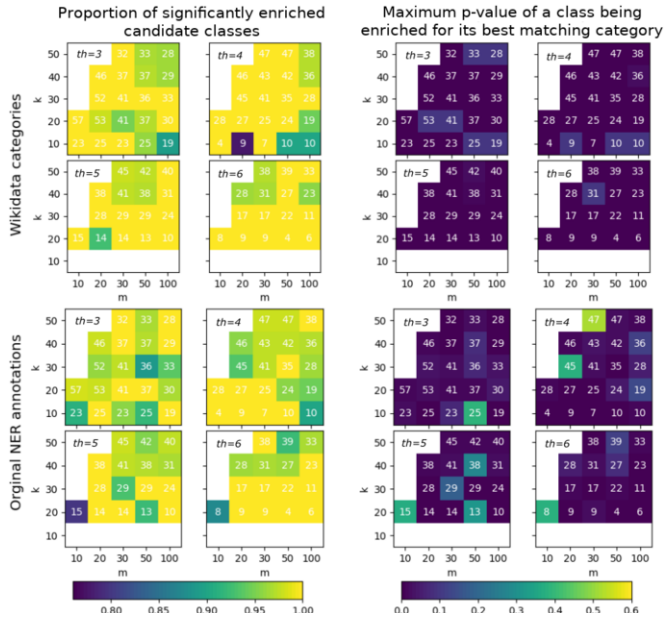
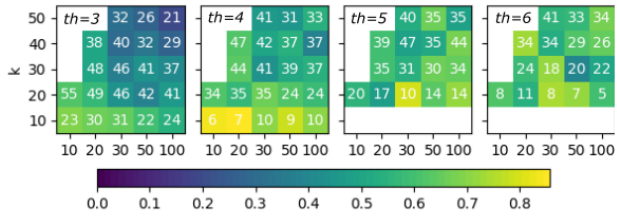


Figure 4. Quantitative evaluation of German candidate classes. Left shows the proportion of candidate classes which are significantly enriched for at least one Wikidata category (top) or of the original NER types the dataset was annotated with (bottom); higher is better. Right shows the maximum p-value of this enrichment, lower is better. Shown in white is the number of candidate classes produced with each combination of parameters.

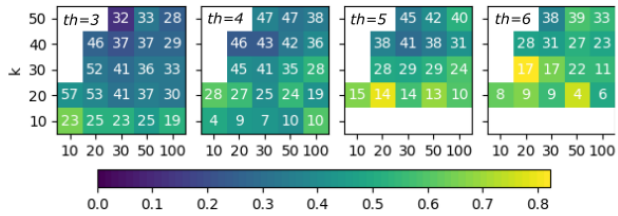
Figure 3 lower left. It is worth noting that for most parameter combinations, even those candidate classes which are not significantly enriched for Wikidata categories are still close to statistically significant (p-value less than 0.1), see Figure 3 upper right. For the German corpus, both the Wikidata categories and the original NER types are significantly enriched in the candidate classes, see Figure 4 left.

The parameters m and k regulate the number of substitutes and sense descriptors, respectively. Therefore, by increasing these parameter values, we might expect better quality of sense and class descriptors. We do observe these effects in the quantitative analysis. However, with too high parameter values this effect gets smaller as the language model produces less relevant substitutes. Moreover, the computational time increases as the binary tables get larger. The other parameter th introduces a threshold on the minimum number of class descriptors. Thus, a larger th yields finer-grained classes.

The granularity of the classes can be assessed by inspecting the Wikidata categories for which they are enriched. When the threshold th is small, the most enriched-for categories are very wide. A close analysis of this can be seen in Figure 5. As the value of th increases, more of the candidate classes found are enriched on Wikidata categories which are smaller than the mean category size (for those categories whose entities are present in the corpus). These smaller categories are more specific, so that instead of a candidate class being enriched with, for example, $Human_{Q5}$, a class could be enriched with $Heads\ of\ state_{Q48352}$. Obtaining candidate classes of different levels of specificity is one of the strong points of the method presented here.



(a) English



(b) German

Figure 5. Proportion of candidate classes which are enriched with small Wikidata categories. We consider a Wikidata category to be small if the size of its intersection with the terms in the corpus is below the mean. For every combination of parameters k and m, the ratio of candidate classes which are enriched for such small categories is shown. The number of candidate classes is given in white.

4.4. Qualitative Analysis of Induced Classes

In order to gain a better understanding of the quality of the proposed approach, we manually checked the classes that resulted from the presented experiments when $m = 30$ and $k = 40$, as the quantitative results for these values showed stable high-quality outcomes over different settings. We investigated the results for $th = 3$ (see Examples 1 and 3) and $th = 6$ (see Examples 2 and 4) both in the English and German dataset.

EXAMPLE 1 (some English Candidate Classes *for m=30 k=40 th=3*)

EN-6-th3 ***Descriptors: Greek, Greece, Athens***
Entities: sparta, bozcaada, cyprus, spitamenes, olympias, Aegean Sea, thessaloniki, athenian empire, . . .

EN-8-th3 Descriptors: Hercules, Jupiter, Prometheus, Zeus, Apollo
Entities: zeus, saturn, Athena, heracles, ajax, mt. olympus, 28 bellon . . .

EN-11-th3 Descriptors: film, Oscar, Film
Entities: william c. demille, 80th Academy Awards, the lady vanishes, sight & sound, douglas fairbanks, golden lion, Best Original Song, . . .

EN-18-th3 Descriptors: Macintosh, Home, iPhone, Apple
Entities: Macworld, apple lisa, apple usb modem, wireless keyboard, game boy color, apple, iphone 3g, magic mouse, . . .

EN-22-th3 Descriptors: Nadal, Masters, Davis, Wimbledon, Serena, Murray, set, Federer
Entities: michael stich, david wheaton, robby ginepri, patrick rafter, Federer, john mcenroe, arnaud clément, . . .

EN-37-th3 Descriptors: Mercury, Gemini, space, NASA
Entities: saturn v, vostok 6, apollo 13, Mercury, command/service module, sts-95, ufo, . . .

EXAMPLE 2 (some English Candidate Classes for *m=30 k=40 th=6*)

EN-9-th6 Descriptors: director, Newman, Oscar, film, Hollywood, Film
william c. demille, alfred hitchcock presents, jean vigo, robert bresson, laurence olivier, henry fonda, grace kelly, john ford, hitchcock, . . .

EN-11-th6 Descriptors: Switzerland, Germany, Europe, Poland, Austria, England
Entities: iceland, Germany, great britain, Holland, norway, estonia, sweden, Israel, . . .

EN-27-th6 Descriptors: Pluto, Mars, Orion, Galileo, Uranus, Titan, Jupiter, Apollo, Mercury
Entities: Mars, 2 Pallas, janus, 944 Hidalgo, 28 bellona, saturn, 37 fides, 52 europa, Phobos, near shoemaker. . .

EXAMPLE 3 (some German Candidate Classes for *m=30 k=40 th=3*)

DE-10-th3 Descriptors: Verein, Club, Fußball
Entities: fc bayern münchen, vfr aalen, tsv crailsheim, fc valencia, fußball-bundesliga der frauen, sc heerenveen, first vienna fc, asiens fußballer des jahres, saison 1977/78. . .

DE-30-th3 Descriptors: Militär, Reich, Volk
Entities: französische heer, kaiserlich russischen marine, nationalen widerstandsrates im iran, bayerische justizministerium, waffen-ss, nationalen volksarmee, heer, . . .

DE-31-th3 Descriptors: Eisenbahn, Strecke, Insel
Entities: turmbergbahn, u-bahn-linie u6, bahnstrecke braunschweig-magdeburg, brücke, pariser métro, b 173, mariazeller straße b20, themse, . . .

EXAMPLE 4 (some German Candidate Classes for *m=30 k=40 th=6*)

DE-8-th6 Descriptors: Bach, Donau, Rhein, Fluss, Saale, Elbe
Entities: mittellandkanal, seille, oder, omerbach, tauber, europäische hauptwasserscheide, . . .

DE-14-th6 Descriptors: Schule, Halle, Universitat, Akademie, Fakultät, Universität, Hochschule
Entities: technische universität chemnitz, universitätssternwarte wien, cambridge university, eth zürich, hochschule für musik detmold, . . .

DE-26-th6 Descriptors: Karlsruhe, Stuttgart, Baden, Tübingen, Heilbronn, Württemberg
Entities: stuttgart, baden-württemberg, cannstatt, heilbronn, ulm, alpirsbach, rottweil, konstanz, . . .

DE-28-th6 Descriptors: Anna, Katharina, Maria, Mathilde, Agnes, Helene
Entities: maria von beuthen, dorothea becker, zabel, beatrix von luxemburg, irina walentinowna moissejewa, anna friessnegg, . . .

DE-30-th6 Descriptors: Louis, Marie, François, Jean, Paul, Joseph
Entities: françois rude, charles-françois von velbrück, aurel emile joliat, auguste de montferrand, jean baptiste janssens, . . .

Overall, produced classes are of high quality, with only a very small number of classes where the assigned entities could not be semantically grouped in an obvious way. Furthermore, their semantic characteristics are quite diverse: One the one hand, classes covering sub-categories of locations were produced, such as countries (e.g, EN-11-th6[6]) and municipalities (e.g., DE-26-th6), as well as geographical entities such as rivers (DE-8-th6), and infrastructural elements such as railway tracks and streets (DE-31-th3). On the other hand, persons were subdivided due to their gender (e.g., DE-28-th6), profession (e.g., EN-9-th6), the origin of their name(DE-30-th6), or their membership to a specific group, such as Greek deities (EN-8-th3). Furthermore, classes of organisations (e.g., sports clubs DE-10-th3, armed forces DE-30-th3, or universities DE-14-th6) were produced. However, some classes also provided a wider view, e.g., EN-6-th3 combines Greece-related entities, both persons and locations. Also, very focused classes such as class EN-18-th3 (technology) and class EN-37-th3 (space) were identified.

The examples show that the specificity of different classes varies a lot (different entities that relate to Greece in EN-6-th3 compared to cities in a single German State in DE-26-th6). As seen in the quantitative analysis, the granularity of the classes is dependent

[6]We use the convention {LANGUAGE}-{CLASS NUMBER}-th{THRESHOLD}

on th, where higher values lead to more specific classes. While classes that already had a high degree of specificity with a lower threshold would remain quite stable with higher thresholds e.g., class EN-22-th3 about tennis stars is identical –both in terms of descriptors and entities– to EN-15-th6, increasing th in more coarse-grained classes can lead to the creation of sub-classes: $th = 3$ resulted in a class (EN-11-th3) about film awards, which includes different awards themselves (Golden Lion, Academy Award) as well as winning (or nominated) actors, directors, and movies. By increasing th to 6, EN-9-th6 is created, which only includes award-winning directors and actors. Another example in this regard is EN-8-th3, which includes Greek gods, but also celestial bodies named after them (Saturn::LOC, 28 Bellona). When increasing the threshold to 6, these two kinds of entities can be successfully separated (see EN-27-th6).

Taking a look at the produced descriptors, we see that for some classes, these descriptors come close to class names, scope of the class can be understood by reading the descriptors only, e.g., DE-14-th6 has the descriptors *School, Hall, University, Academy, Faculty* and summarises higher education facilities. However, for others, additional background knowledge is needed, in order to get an understanding of their content, e.g., to recognise the tennis stars theme in class EN-22-th3 with descriptors *Nadal, Masters, Davis, Wimbledon, Serena, Murray, set, Federer*. Even though the concrete granularity of a class can be only determined by taking a look at the entities, in almost all of the cases, the descriptors provided helpful insights into the class content.

For retrieving the additional background knowledge to understand the class content, it can be helpful to consider the best matching Wikidata category as it provides some useful insights for possible semantic connections between the entities. For example, while at first glance the descriptors *(Austria, Uzbekistan, Uganda, San Marino)* might only share the shallow relation of being countries, the common Wikidata link *Landlocked country*$_{Q123480}$ reveals a non-obvious, more specific connection. The same holds for DE-26-th6, where –for a non-expert– it might be difficult to see that all entities are towns located in the German state of Baden-Württemberg when not provided with the Wikidata link to *city district of Baden-Württemberg*$_{Q2327515}$. Still, in most cases, the best matching Wikidata category seems too broad to provide helpful insight (e.g., for almost all person-related classes, the category *Human*$_{Q5}$ or *Common Names*$_{Q502895}$ is provided), and sometimes it is even confusing (EN-11-th6 has the common broader *Legal science*$_{Q382995}$).

4.5. Benchmark and comparison to other methods

The work presented here produces sets of semantically coherent entities from an annotated corpus. This task is, in a wide sense, equivalent to the tasks solved by two different approaches: topic modeling (TM), and topic taxonomy induction (TTI). The former is well-known and has many different solutions, and the latter deals specifically with the construction of hierarchies and state of the art solutions make also use of embeddings. Both approaches make the additional assumption that the corpus is partitioned into documents, and derive from this partition information about the semantic relatedness of entities. In order to gauge the performance of our method, we compare to both approaches.

For TTI, the state of the art TaxoGen[35] method was used, as per the original author's implementation. For TM, a standard approach using count-based vectorisation followed by LDA, both using the Scikit-learn library, was performed, and an entity was deemed part of a topic by thresholding the resulting row-normalised term-topic matrix

Table 2. Comparison to other methods. We compare the best sets of topics produced by different methods according to three criteria. The best values are marked in **bold**. TTI stands for the TaxoGen implementation of Topic Taxonomy Induction, and TM for topic modelling.

Optimized for:	Enrichment with small Wikidata categories			Many classes enriched with Wikidata categories			Max. p-value of enrichments to Wikidata categories		
Method	our	TTI	TM	our	TTI	TM	our	TTI	TM
Number of candidate classes	7	5	36	14	17	12	26	19	10
Average candidate class size	15.7	10.0	88.0	16.4	10.0	390.5	80.8	10.0	461.2
Prop. enriched with small Wikidata categories (HIB)	0.86	**1.0**	0.75	0.71	**0.76**	0.0	0.27	**0.74**	0.0
Prop. significantly enriched (HIB)	**1.0**	**1.0**	0.83	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Max. p-value of enrichment (LIB)	**4.5e-4**	5.5e-4	1.9e-3	3.7e-4	8.7e-5	**2.9e-16**	9.9e-9	2.9e-5	**5.4e-18**

with thresholds 0.01, 0.1, 0.2, and 0.5. In both cases, each document in the corpus was represented as a list of entities (as per the annotations), and the number of topics (clusters, in TTI) was varied from 5 to 50 in increments of two. For each criteria, the best decomposition by each method was selected, and basic statistics on topic size were computed (see Table 2). Comparison was done only for the English language corpus.

The results of the comparison (Table 2) show that TM tends to generate very large topics, which makes it an unfavorable choice when trying to find collections of entities with very specific similarities. Therefore, our method outperforms TM in finding groups of entities matching small Wikidata categories, while TM yields very good p-values for enrichment. In contrast, both our method and TaxoGen lead to smaller classes (clusters), which are also very good matches to Wikidata categories. Our method achieves results comparable to the state-of-the-art, while neither requiring specific re-training of embeddings, nor assuming the corpus to be partitioned into documents.

Our method more often produces more classes that are better populated with entities than the results of TaxoGen. In the second column of Table 2, though our method outputs only 14 candidate classes as opposed to 17 by TaxoGen, the coverage of original entities is 35% higher than by TaxoGen because our candidate classes on average have 64% more entities. In other columns the difference is even larger. Therefore, the method presented here covers the given entities better and produces a more complete classification. This feature is specifically important when working with small corpora or with corpora with little annotations.

5. Conclusions

We present a method to automatically induce classes from an entity-annotated corpus. Our method exploits the ability of modern language models to predict lexical substitutes for a target in a given context, to tackle potential polysemy of the annotated entities to induce senses of the annotations on the fly. The generated entity senses are grouped into coherent classes with human-interpretable class descriptors. Importantly, our method requires no additional supervision, can work with annotations coming from different kinds of tools, and does not require the partitioning of input corpus into documents.

With different parameter combinations, this method allows for classes to represent differently grained classifications of entities. This allows, for example, recognizing par-

ticular entities as belonging to the coarse topic of *persons* and, with a different combination of parameters, as belonging to more fine-grained subclasses of people, such as *Presidents of the United States*, or *19th century painters*. Generalizations of our method are capable of extracting hierarchies of classes.

We evaluate our method on large general-purpose corpora in two languages. Quantitative and qualitative evaluations show our method's ability to induce a set of classes that is in agreement with external classification schemes in Wikidata. The quality of the results from our method is comparable to one of the state of the art methods (TaxoGen), however our method covers the original annotations better and yields more a complete classification of the original entities.

Our method is also applicable to small domain-specific corpora, since the usage of pre-trained language models on short contexts (as opposed to documents for other methods) allows for capturing the contextual semantics of previously unseen domain-specific words. Additionally a good coverage of the original entities in the output classification makes efficient use of smaller quantities of input data.

References

[1] Al-Aswadi, F.N., Yong, C.H., Gan, K.H.: Automatic ontology construction from text: a review from shallow to deep learning trend. Artif. Intell. Rev. **53**(6), 3901–3928 (2020)

[2] Albukhitan, S., Helmy, T., Alnazer, A.: Arabic ontology learning using deep learning. In: Proceedings of the International Conference on Web Intelligence. p. 1138–1142. WI '17, Association for Computing Machinery, New York, NY, USA (2017)

[3] Amrami, A., Goldberg, Y.: Word Sense Induction with Neural biLM and Symmetric Patterns. In: Proceedings of EMNLP 2018. pp. 4860–4867. Brussels (2018)

[4] Asim, M.N., Wasim, M., Khan, M.U.G., Mahmood, W., Abbasi, H.M.: A survey of ontology learning techniques and applications. Database **2018** (2018)

[5] Belohlavek, R., Vychodil, V.: Discovery of optimal factors in binary data via a novel method of matrix decomposition. Journal of Computer and System Sciences **76**(1), 3 – 20 (2010)

[6] Beltagy, I., Lo, K., Cohan, A.: Scibert: A pretrained language model for scientific text. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3606–3611 (2019)

[7] Browarnik, A., Maimon, O.: Ontology learning from text: Why the ontology learning layer cake is not viable. Int. J. Signs Semiot. Syst. **4**(2), 1–14 (2015)

[8] Buitelaar, P., Cimiano, P., Magnini, B.: Ontology learning from text: An overview (2005)

[9] Casteleiro, M.A., Prieto, M.J.F., Demetriou, G., Maroto, N., Read, W.J., Maseda-Fernandez, D., Des Diz, J.J., Nenadic, G., Keane, J.A., Stevens, R.: Ontology learning with deep learning: a case study on patient safety using pubmed. In: SWAT4LS (2016)

[10] Chen, C., Lin, K., Klein, D.: Inducing Taxonomic Knowledge from Pretrained Transformers. arXiv preprint arXiv:2010.12813 (2020)

[11] Cimiano, P., Hotho, A., Staab, S.: Learning concept hierarchies from text corpora using formal concept analysis. J. Artif. Intell. Res. **24**, 305–339 (2005)

[12] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2018)

[13] Drymonas, E., Zervanou, K., Petrakis, E.G.M.: Unsupervised ontology acquisition from plain texts: The ontogain system. In: NLDB (2010)

[14] Geng, Y., Chen, J., Zhuang, X., Chen, Z., Pan, J.Z., Li, J., Yuan, Z., Chen, H.: Benchmarking Knowledge-driven Zero-shot Learning. arXiv e-prints arXiv:2106.15047 (Jun 2021)

[15] Huang, K., Altosaar, J., Ranganath, R.: Clinicalbert: Modeling clinical notes and predicting hospital readmission. arXiv preprint arXiv:1904.05342 (2019)

[16] Khadir, A.C., Aliane, H., Guessoum, A.: Ontology learning: Grand tour and challenges. Comput. Sci. Rev. **39**, 100339 (2021)

[17] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J.: Biobert: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics **36**(4), 1234–1240 (2020)
[18] Liu, H., Perl, Y., Geller, J.: Concept placement using BERT trained by transforming and summarizing biomedical ontology structure. Journal of Biomedical Informatics **112** (Dec 2020)
[19] Mi, H., Muruganujan, A., Casagrande, J.T., Thomas, P.D.: Large-scale gene function analysis with the PANTHER classification system. Nature protocols **8**(8), 1551–1566 (2013)
[20] Michael, J., Botha, J.A., Tenney, I.: Asking without Telling: Exploring Latent Ontologies in Contextual Representations. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 6792–6812. Association for Computational Linguistics, Online (Nov 2020)
[21] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 26, pp. 3111–3119. Curran Associates, Inc. (2013)
[22] Nikishina, I., Logacheva, V., Panchenko, A., Loukachevitch, N.V.: RUSSE'2020: Findings of the First Taxonomy Enrichment Task for the Russian language. CoRR **abs/2005.11176** (2020)
[23] Nivre, J.: Incrementality in deterministic dependency parsing. In: Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together. pp. 50–57. Association for Computational Linguistics, Barcelona, Spain (Jul 2004)
[24] Nothman, J., Ringland, N., Radford, W., Murphy, T., Curran, J.R.: Learning Multilingual Named Entity Recognition from Wikipedia. Artif. Intell. **194**, 151–175 (Jan 2013)
[25] Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)
[26] Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training. URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf (2018)
[27] Revenko, A., Breit, A., Mireles, V., Moreno-Schneider, J., Sageder, C., Karampatakis, S.: Annotating entities with fine-grained types in austrian court decisions. In: Further with Knowledge Graphs, pp. 139–153. IOS Press (2021)
[28] Sanh, V., Debut, L., Chaumond, J., Wolf, T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter (2019)
[29] Shamsfard, M., Barforoush, A.A.: Learning ontologies from natural language texts. Int. J. Hum. Comput. Stud. **60**, 17–63 (2004)
[30] Shang, J., Zhang, X., Liu, L., Li, S., Han, J.: Nettaxo: Automated topic taxonomy construction from text-rich network. In: Proceedings of The Web Conference 2020. pp. 1908–1919 (2020)
[31] Sousa, D., Couto, F.M.: BiOnt: Deep Learning Using Multiple Biomedical Ontologies for Relation Extraction. In: Jose, J.M., Yilmaz, E., Magalhães, J., Castells, P., Ferro, N., Silva, M.J., Martins, F. (eds.) Advances in Information Retrieval. pp. 367–374. Springer International Publishing, Cham (2020)
[32] Vayansky, I., Kumar, S.A.: A review of topic modeling methods. Information Systems **94**, 101582 (2020)
[33] Vedula, N., Nicholson, P.K., Ajwani, D., Dutta, S., Sala, A., Parthasarathy, S.: Enriching taxonomies with functional domain knowledge. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. p. 745–754. SIGIR '18, Association for Computing Machinery, New York, NY, USA (2018)
[34] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Brew, J.: HuggingFace's Transformers: State-of-the-art Natural Language Processing. CoRR **abs/1910.03771** (2019)
[35] Zhang, C., Tao, F., Chen, X., Shen, J., Jiang, M., Sadler, B., Vanni, M., Han, J.: TaxoGen: Unsupervised Topic Taxonomy Construction by Adaptive Term Embedding and Clustering. In: Proceedings of the 24th ACM SIGKDD. p. 2701–2709. KDD '18, Association for Computing Machinery, New York, NY, USA (2018)