Pruning and repopulating a lexical taxonomy: experiments in Spanish, English and French

Rogelio Nazar, Antonio Balvet, Gabriela Ferraro, Rafael Marín, Irene Renau

Research Article

Rogelio Nazar*, Antonio Balvet, Gabriela Ferraro, Rafael Marín, and Irene Renau

Pruning and repopulating a lexical taxonomy: experiments in Spanish, English and French

https://doi.org/10.1515/jisys-2020-0044
Received Apr 29, 2020; accepted Aug 18, 2020

Abstract: In this paper we present the problem of a noisy lexical taxonomy and suggest two tasks as potential remedies. The first task is to identify and eliminate incorrect hypernymy links, and the second is to repopulate the taxonomy with new relations. The first task consists of revising the entire taxonomy and returning a Boolean for each assertion of hypernymy between two nouns (e.g. *brie is a kind of cheese*). The second task consists of recursively producing a chain of hypernyms for a given noun, until the most general node in the taxonomy is reached (e.g. $brie \rightarrow cheese \rightarrow food \rightarrow$ etc.). In order to achieve these goals, we implemented a hybrid hypernym-detection algorithm that incorporates various intuitions, such as syntagmatic, paradigmatic and morphological association measures as well as lexical patterns. We evaluate these algorithms individually and collectively and report findings in Spanish, English and French.

Keywords: hypernymy detection; language independent methods; taxonomy induction; unsupervised methods

2020 Mathematics Subject Classification: 68W06

1 Introduction

Different lexical taxonomies have been proposed in the past for many languages, whether manually crafted, obtained by automatic processes or combinations of both [31, 34, 35, 41]. However, such repositories typically present limitations both in precision (some of their hypernymy links are incorrect) and recall (they have an insufficient quantity of vocabulary). In this paper, we identify the preceding scenario as our research problem and show how a given noisy or partially constructed lexical taxonomy can be improved in quality and quantity by identifying and discarding incorrect data and, subsequently, repopulate such repository with fresh hypernymy links.

We claim it is possible to achieve both goals with an unsupervised algorithm for taxonomy induction. In the case of the first task, cleansing a taxonomy would entail analyzing each hypernym link of the resource in question and returning a Boolean, i.e. to produce *true* or *false* given an assertion such as *a desk is a kind of furniture* or *#a robin is a kind of furniture* and so on. The second task lies within the area of taxonomy induction, and is a longstanding and well-understood problem in Natural Language Processing [4, 26]. Taxonomy induction includes the sub-task of hypernymy discovery, which means producing a hypernymy of a given input noun, but goes further by recursively producing also the hypernym of the hypernym until a root node – represented by the most general term– is met. In order to accomplish both tasks, we propose a hybrid method that combines several intuitions, each one incorporated as an individual module. It combines language inde-

***Corresponding Author: Rogelio Nazar:** Pontificia Universidad Católica de Valparaíso, Chile; Email: rogelio.nazar@pucv.cl
Antonio Balvet: Université de Lille, France; Email: antonio.balvet@univ-lille.fr
Gabriela Ferraro: DATA61 & Australian National University, Australia; Email: gabriela.ferraro@data61.csiro.au
Rafael Marın: Université de Lille, France; Email: rafael.marin@univ-lille.fr
Irene Renau: Pontificia Universidad Católica de Valparaíso, Chile; Email: irene.renau@pucv.cl

pendent distributional properties of words, more precisely, paradigmatic and syntagmatic association measures; word embeddings; measures of morphological similarity between co-hyponyms and lexico-syntactic patterns *à la* Hearst [22]. Each module produces a result that contributes to a final decision made by a ranking procedure.

With the exception of two modules (one containing morphological rules and the other lexical patterns), our method is largely language-independent. The majority of the modules are based on the computation of distributional and morphological properties of words without the need for explicit linguistic information. And the fact that it is an unsupervised algorithm means that there is no need for expensive training phases and there are no risks of bias in the process.

More critically, our proposal differs from other unsupervised methods in that we produce a full taxonomy chain for each input noun using a combination of top-down and bottom-up approaches. That is, given an input noun, we start with a top-down approach by asking the most general questions: Is it an entity, a property or an event? Then, if it is an entity, we ask if it is an abstract or concrete entity, and then if it is animate or inanimate and so on, until we reach a medium point of abstraction such as, say, we find out that the noun refers to a type of weapon, or a type of fish. The latter are nouns that are general enough to be included in a "Core Ontology", a selection of ca. 300 of the most general nouns in a language. It is at this point that we apply the inverse procedure, i.e. the bottom-up approach, searching for the most specific or immediate hypernyms. With all data gathered, the algorithm chooses the best way to connect the two segments of the taxonomy chain.

We report on experiments in Spanish, English and French. With our datasets, precision tends to average at 0.8 across languages and domains. We observed, also, that the contribution of each module to the overall result is insignificant, in the sense that we can arbitrarily switch off one or two of them without affecting results. Interestingly, the strength of the method relies on all of its parts working collectively.

The method achieves competitive results in comparison to previous work on taxonomy induction and hypernymy discovery, according to tests we undertake using gold standard evaluation material provided by [4] and [7]. We have to say, however, that a non-negligible proportion of errors exists in these datasets, and this is a problem when trying to produce meaningful comparisons. This problem illustrates the need for a pruning algorithm such as the one we now present.

We propose a new, manually curated evaluation dataset, that exhibits the following three properties: 1) language closeness; 2) same semantic categories of general vocabulary across languages; and 3) inclusion of (annotated) false hypernymy links consisting of pairs of words that are semantically related but not in a hypernymy relation, e.g. *# tuna is a type of cheese*. To complement this with specialized terminology, we also compiled a large list of psychopharmacological drugs. We hope other researchers in the field will find it useful and build upon this material.

The paper is organized as follows: the next section offers a brief summary of precedents in hypernym detection. Section 3 describes our methods and materials; Section 4 presents the results and Section 5 offers some conclusions and possibilities for future work.

2 Background

2.1 Hypernymy and lexical taxonomies

The concept of hypernymy is fundamental for the semantic description of the vocabulary of a language [13], as it organizes the vocabulary into semantic categories. Entities can be hierarchically organized in categories and connected in a tree-shaped configuration, such that each node in the tree inherits the properties of its parent, e.g. a horse is a type of animal and therefore all properties that are common to animals will also apply to horses. The interest in hypernymy predates modern linguistics, and can be traced to the work of Aristotle, who outlined its basic properties in the *Categories*.

Hypernymy relations are the basic links organizing lexical taxonomies (where links are drawn between lexical units) as well as ontologies (where links are drawn between concepts). One would use taxonomies or ontologies depending on whether one has a semasiological or onomasiological perspective [45], the first being the traditional choice in linguistics and the second in artificial intelligence. Whereas an ontology requires a tree-like constraint, a lexical taxonomy does not, because nouns sometimes have more than one hypernym due to polysemy or differences in point of view (e.g., we can see a chicken or lettuce as a living organism or as a type of food).

Hypernymy relations, however, are common to both types of structures. Other properties that are common to both are:

1. Their hierarchic structure.
2. The shape of a direct acyclic graph (i.e. not containing cycles because words cannot be used in their own definition).
3. The asymmetry of the relation (1), e.g. if a robin (r) is a kind of bird (b), then a bird is not a kind of robin.

$$\forall r, b \in T, r \rightarrow b \Rightarrow \neg b \rightarrow r \tag{1}$$

4. The transitivity (2), e.g. if a robin is a bird and a bird is a kind of animal (a), then a robin is a kind of animal.

$$\forall r, b, a \in T : r \rightarrow b \wedge b \rightarrow a \Rightarrow r \rightarrow a \tag{2}$$

2.2 Taxonomy induction in NLP

Attempts at automatic extraction of hypernymy links were first made on machine readable dictionaries [8, 19]. Later, hypernymy links were induced from corpora using lexical patterns [10, 22, 37, 44, 54]. These patterns were found to have considerable precision but low recall, since both the hyponym and its hypernym must co-occur in the sentence or text span.

This limitation inspired quantitative methods, which were applied as syntagmatic and paradigmatic association measures. Syntagmatic association measures are well-known in linguistics. Mutual Information (MI) [9], for instance, has been extensively used in lexicography to detect collocations, multiword expressions and other types of syntagmatic bonds. The basic idea is to oppose the frequency of co-occurrence of two units against their independent frequency, as done by other measures such as t-test, chi-square and cosine, among others. The limitation for these types of measures is mainly that they are symmetric. Paradigmatic association measures, on the other hand, are derived from the concept of distributional similarity [2, 6, 14, 18, 27, 30, 46, 51].

Unsupervised methods based upon distributional similarity properties dominated the hypernym detection scene for a long period of time [11, 18, 25–27, 47, 48, 51, 56]. In such methods, the task is to assign a score to a pair of words (x, y) using a measure that exploits the distributional similarity hypothesis from [21], with high scores indicating strong possibility of an hypernymy relation. Recently, values obtained with distributional similarity methods have been incorporated into supervised methods as features, outperforming unsupervised methods, as discussed in [53]. However, it is clear that supervised methods are restricted by the availability of training data, which is costly and may be even unavailable in some cases.

Several measures have been proposed by other researchers, taking into account different distributional aspects of hypernym relations. According to [53], the classification measures can be similarity, inclusion, and informativeness. Some measures are framed around the concept of similarity since similar words appear in similar contexts, even though hypernymy relations are asymmetric. Some examples of similarity measures are: cosine similarity [47]; Lin similarity [27], which calculates the ratio of shared contexts of each word; and APSyn [48], that computes the intersection of contexts of a word pair weighted according to the rank of the shared contexts.

Other measures are inspired by the concept of inclusion, in which the contexts of a hyponym are expected to be included in those of its hypernym: $X_c \subset Y_c$. This idea is exploited by [56], who quantify the inclusion of

x's contexts by the contexts of y. Later, [26] combined cosine and Weeds precision. ClarkeDE [11] measured inclusion by quantifying the weighted coverage of the hyponym's context by the contexts of the hypernym. Also, [25] designed a metric called balAPinc, that combines Lin similarity with APinc, an average precision measure taken from information retrieval. In [26], there is invCL, as a measure that takes into account the distributional inclusion of x in y and the distributional non-inclusion of y in x. Similarly, other measures are based upon the concept of reversed inclusion, in which the relevant contexts of a hypernym are expected to be included in those of its hyponyms [53].

Finally, some researchers have found inspiration in the concept of informativeness. In this context, hypernyms tend to be less informative than hyponyms, as they are likely to occur in more general contexts than their hyponyms. Cf.: SLQS [49], which measures the informativeness of a word x evaluated as the median entropy of its top N contexts; and RCTC [43], which measures the ratio of change in topic coherence.

According to the comparative study carried-out by [53], it is not possible to single out a preferred measure or parameter, even though the SLQS measure is slightly better than others when tested with English word pairs. However, as already mentioned, it is possible to incorporate scores of distributional similarity-based methods into supervised models as features. For instance, [16] presented a supervised model based on sense embeddings trained on available lexical resources such as Wikipedia, and [50], also using a supervised model, proposed an algorithm that combines different methods, such as word embeddings via word2vec, term substring and Jaccard similarity measures.

As we include in our work on vector-based representations of words, we consider it relevant, as well, to briefly review the three main approaches representing words: (i) Distributional representation; (ii) Cluster-based representation; and (iii) Distributed representations.

Distributional representation methods map each word w to a context word vector $\mathbf{C}_w$, which is constructed directly from co-occurrence counts between w and its context words. The learning methods either store the co-occurrence counts between two words w and i directly in C_{wi} [23, 55] or project the co-ccurrence counts of words into a lower dimensional space [29, 42], using dimensionality reduction techniques such as SVD [15] or LDA [3].

Cluster-based representation methods build clusters of words by applying either soft or hard clustering algorithms [28]. Some of them also rely on a co-occurrence matrix of words [38]. The Brown clustering algorithm [5] is the best-known method in this category.

Distributed representation methods usually map words into dense, low dimensional, continuous-valued vectors, with $x \in R^d$, where d is referred to as the word dimension. In this category we find word embeddings, which take a text corpus as input and produce word vectors as output [12, 30]. The hypothesis is that similar words would have similar word embedding vectors. The method we propose in this paper is hybrid but mainly belongs to the first category.

3 Methodology

In this section, we describe the two main components of the proposed method, i.e. the pruning and the repopulation algorithms. The first one takes a taxonomy and identifies and deletes errors. As a result, the original taxonomy has fewer errors but it is also reduced in size. This result is then used by the second algorithm, which extracts different kinds of features to use them for classifying new terms obtained from corpora in order to replenish the taxonomy.

The repopulation algorithm is therefore a semantic classifier, which takes a given noun as input and outputs its full taxonomy chain. This classifier consists of a number of independent modules, which are divided in top-down and bottom-up procedures, as will be explained shortly. A final module integrates the output of all the others by a decision-making process, and produces a categorization of said noun with the help of a Core Taxonomy, which contains the most general semantic categories of a language.

3.1 Materials

3.1.1 The Legacy Taxonomy

The whole approach rests on the assumption that a Legacy Taxonomy is already available with at least some proportion of true hypernymy links. As already mentioned, such material is currently available for free in different languages. The multiple versions of WordNet, for instance, are suitable candidates to be analyzed with the proposed method.

In our case, however, we used material developed in our previous studies [34], consisting of lists of ca. 50,000 pairs of single noun hyponym-hypernym pairs of general vocabulary in Spanish, English and French with an average error rate close to 30%. The idea is thus to improve this raw material by reducing the number of errors (the pruning phase) and increasing the number of hypernymy pairs (the repopulation phase).

3.1.2 The *Core Ontology*

Another resource that was used to conduct these experiments is what we call the Core Ontology (or CoreOnt, for short), a tree-shaped arrangement of ca. 300 of the most general concepts, starting with those such as *entity, property, event, group* and so on (Figure 1). This structure is used by the top-down modules.

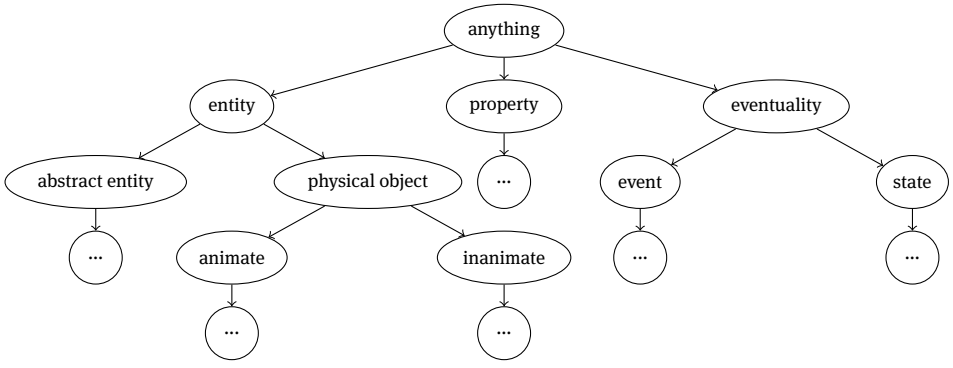

Figure 1: A fragment of the CoreOnt

The CoreOnt is loosely based on the CPA Ontology[1], developed by Patrick Hanks for the Pattern Dictionary of English Verbs [20]. We freely adapted it to our purposes, adding or eliminating terms according to their frequency in corpus, and translated it to Spanish and French.

3.1.3 The corpus

As most of the modules are corpus-based, we need a large textual corpus to operate. The corpora from which we extracted the material were the French, English and Spanish versions of the TenTen corpus [24], with approximately 10^{10} tokens per language, the largest generic corpora currently available for these and other languages. It is composed of web pages randomly crawled from web top-level domains, which have been later converted to plain text, tokenized, lemmatized and POS-tagged.

1 http://pdev.org.uk

3.2 The pruning algorithm

The input of this algorithm is a list of suspected hypernymy pairs ($a \rightarrow h$) and the output is a true or false value for each pair. As explained earlier, the objective for this pruning step is to obtain a reduced number of assertions with a high probability of being correct, from which to extract features that in turn are used to classify new nouns during the repopulation phase. The assertions that are likely to be true are denoted as matrix $T_{a,h}$, where a is a member element (a hyponym) and h is the semantic category (a hypernym).

The pruning is conducted according to a measure of syntagmatic association. Thus, the output of this first module for $T_{a,h}$ is obtained from estimator $M_{a,h}$ (3), which measures the frequency of co-occurrence in the corpus of the alleged hypernymy pair $a \rightarrow h$. It can be said that $M_{a,h}$ is a measure of the relative importance of h to a, and is therefore appropriate for asymmetric relations like hypernymy.

$$M_{a,h} = \frac{\sum_{i=1}^{|C|} \begin{cases} 1 & h \in C_i \\ 0 & \text{otherwise} \end{cases}}{|C| + 1} \tag{3}$$

$M_{a,h}$ is computed by the analysis of a random sample of contexts of occurrence of each hyponym a in the corpus, denoted as set $C = \{c_1, \ldots c_n\}$ (with $n <= 5000$ to allow for lower processing time). The context window is ten words at each side of the target and each context C_i is represented as a bag of words, with no order and no repetitions. M can be used to compute a ranking of the most reliable assertions or just to return a Boolean if $M_{a,h} > p$, with p as an arbitrary parameter (experimentally set to 0.3). Additionally, a true value for a pair $M_{a,h}$ is only issued if it passes the asymmetry test shown in (4).

$$S_{a,h} = \begin{cases} \text{true} & M_{a,h} > M_{h,a} \\ \text{false} & \text{otherwise} \end{cases} \tag{4}$$

3.3 The repopulation algorithm

The repopulation algorithm accepts one or more words as input and tries to classify them by adding them to the now cleaner version of the Legacy Taxonomy. It is composed of various modules, each of which is identified with a code name (e.g. *asymmetric*, *paradigmatic*, *morfeo*) and described in detail in the following subsections.

As already anticipated, much of the classification is done by extracting features from the cleaned up Legacy Taxonomy. Different modules use different types of features (e.g. distributional, morphological) but all of them are subjected to a process of weighting in order to select the most discriminant ones. We score the features according to a given semantic category, as shown in Equation 5.

$$pond(F_i, T_j) = \frac{f(F_i|T_j)}{\sqrt{|T_j|}} \cdot \frac{1}{\sqrt{D(F_i)}} \tag{5}$$

Let F_i be some feature (e.g. a morphological one, like the final letter-sequence *-itis*); T_j a given semantic category (e.g. *disease*); $f(F_i|T_j)$ the frequency of feature F_i in the category T_j (the number of hyponyms bearing this trait within the category) and $D(F_i)$ the dispersion of said feature F_i, i.e. the number of semantic categories in which it is present. This score thus promotes features that are concentrated in few categories and that, at the same time, are productive in each category. To continue with the same example, this would be the case of finding a great proportion of words ending with the segment *-itis* among the population of words in the category of *diseases*.

$$F_i \in FM_{m,h,f} \iff pond(F_i, T_j) > u \tag{6}$$

Best features are kept in a matrix $FM_{m,h,f}$ as shown in 6, where m stands for the name of the module, h for the hypernym (the semantic category), f for some feature and u is an empirically defined parameter, set to 0.001 in our experiments.

3.3.1 Asymmetric: a syntagmatic association module

This module follows an intuition similar to the one used by the pruning algorithm, i.e. that, in general, nouns have a tendency to co-occur with their hypernyms in a non-reciprocal manner [33]. For instance, if we examine the co-occurrence vector of a word such as *motorcycle* (showing those words that frequently co-occur in the vicinity of the target word), probably the second most frequent co-occurring noun after *motor* will be *vehicle*. In contrast, *motorcycle* will not be among the most frequent co-occurring words in the vector of *vehicle*. Likewise, words co-occur frequently with their co-hyponyms, and then in the co-occurrence vector of *motorcycle* we see frequent items such as *automobile, bicycle, car*, and so on.

Having observed this behavior, we derived a simple strategy based on the asymmetric association of hyponym-hypernym pairs. Given an input noun a, we first extract its concordances from the corpus and build a co-occurrence vector $\vec{a}$ (7).

$$\vec{a} = \begin{bmatrix} a & b & c & d & \dots \end{bmatrix} \tag{7}$$

The components of $\vec{a}$ are words (adjectives, verbs and nouns) having a significant frequency of co-occurrence with a in the context window. The vector includes a itself as it tends to be the most frequent word in its own contexts of occurrence. Section 3.2 already explained how the contexts of occurrence are extracted. In this case, we restricted the selection of components to single word units, but nothing would change if word n-grams were also to be included. There is certainly future work to be carried out in this direction, specifically with the aim of finding a balance between precision and costs in terms of processing time and energy consumption. Also, for simplicity, the dimensionality $|\vec{a}|$ was limited to 100, a threshold defined by testing.

This is thus how the first order co-occurrence vector is built with components that are words syntagmatically related to a. However, from this we also derive a second order co-occurrence vector, with the words that co-occur with those that co-occur with a. Correct hypernyms for a can usually be found among the most frequent words of the first and second order co-occurring vectors of a. We can represent this reasoning in a matrix (8).

$$\begin{bmatrix} a & b & c & \mathbf{d} & \dots \\ b & \mathbf{d} & a & e & \dots \\ c & f & g & h & \dots \\ d & i & \mathbf{d} & j & \dots \\ \dots & \dots & \dots & \dots & \end{bmatrix} \tag{8}$$

$$a \rightarrow d \iff d \in T \tag{9}$$

The first row in (8) represents the first order co-occurrence vector, corresponding to input word a. The first column displays the same words, such that each row represents the first order vector of the word in the first column. In this case, given that d is the most frequent word in the matrix for a, and if d happens to be a known category in T, then this algorithm will promote the candidacy of d as a hypernym of a (9).

3.3.2 Paradigmatic: a method based on distributional similarity

The next module is based on distributional similarity, inspired by [21]. It is based on the idea that semantically similar words will share some vocabulary in the contexts in which they appear. As already mentioned, various authors have exploited distributional similarity measures for the generation of semantic resources, such as [18] or [27].

We already explained how first-order co-occurrence vectors are built. We also discussed how features are selected according to a score (shown in Equation 5). In the context of this module, we represent matrix $FM_{m,h,f}$ as the features that are highly associated with h, a given hypernymy, i.e. with words showing significant frequency of co-occurrence with words that are hyponyms of h.

Given an input noun a, this module decides if h is a good hypernym candidate for a ($a \in h$) by calculating the overlap between $\vec{a}$ and $\vec{h}$, the vector from $FM_{m,h}$, as shown in Equation 10.

$$\forall a \notin h \wedge \forall h \in T : paradigm(\vec{a}, \vec{h}) = \frac{|\vec{a} \cap \vec{h}|}{|\vec{a}| + 1} \tag{10}$$

3.3.3 Embed: a word-embeddings module

In the same family of distributional approaches, we incorporated a module based on word-embeddings vectors using word2vec [30], which has become very popular in NLP and is used in a host of different applications, from polarity detection to text classification. It is considered a distributional semantics method because it is based on the intuition that words with similar distributional profiles tend to exhibit meaningful semantic relationships.

We used an instance of word2vec trained on the TenTen corpora. The training was performed by using Facebook's open-source word2vec implementation, fastText[2] [17]. We used both the *skipgram* algorithm, the most widely adopted algorithm for word vectors, as well as *cbow*, an implementation of a bag-of-words algorithm. The *skipgram* algorithm is reportedly better at detecting words similar in form to the query, while *cbow* is designed to take each word's context into account more accurately.

We expected a module based on word embeddings to contribute to a taxonomy population algorithm because word vectors are well suited for this kind of task: given an input word, word2vec yields a list of the most similar words, i.e. its nearest neighbors (*NN*). For instance, given the input term *ciclosporin* (an anti-rejection drug), word2vec bag-of-words model trained on FrTenTen yields the following list: *cyclosporine, céphalosporine, xytétracycline, chlorpromazine, paromomycine, céphalosporines, antibioprophylaxie, prochlorperazine, tétracycline* and so on. In this case, they all belong to the same semantic field as *ciclosporin*: *cyclosporine* is a spelling variant, and all the other outputs are names of drugs related to the medical domain. In another example, this time from the domain of food, given the input word *cassoulet*, its *NN* are the following: *cassoulet, saucisse, folpoulet, saucissonné, rhassoul, cassonade, bourguignon, goulet, bourguignon*. As it can be seen, sometimes the output is not semantically but merely orthographically related with the input word, or even to a word semantically related with the input, as in the case of *folpoulet*, some unknown form orthographically related with *poulet*, French for *chicken*.

The way in which we apply this algorithm is exactly as in the case of the previous model: features (the *NN*) are first scored according to Equation 5, and those who survive are admitted to $T_{m,h,f}$, to be used later in the comparison between features of an input word vector ($\vec{a}$) and the sum of the weighted features of all the hyponyms of a hypernym candidate ($\vec{h}$).

3.3.4 Morfeo: morphological similarity between co-hyponyms

The strategy of this module is different from those described earlier in that it does not involve the use of corpora or the extraction of distributional features. Instead, this step consists of analyzing the overlap between co-hyponyms in terms of final character n-grams (sequences of 3 to 5 letters at the end of each word). These features are extracted from the Legacy Taxonomy and weighted exactly as with the other modules, i.e. with Equation 5. This means that we have already a matrix $FM_{m,h,f}$ with the best scored features.

Thus, given an input word a (e.g. *poliomyelitis*) and a hypernym candidate h (e.g. *disease*), we decide if $a \rightarrow h$ according to Equation 11. Using the same notation as earlier, the symbol $\vec{h}$ denotes a vector of character

2 https://fasttext.cc/

n-grams associated with hypernym h, and $M_(a, f)$ to denote a particular feature f (e.g. *-itis*) in input word a.

$$MS(a \rightarrow h) = \begin{cases} true & M_{(a,f)} \in \vec{h} \\ false & \text{otherwise} \end{cases} \tag{11}$$

The method can be considered language-agnostic since it does not rely on any external knowledge source (e.g. lexicons, rules or morphological knowledge). Of course, the selection of character sequences depends on the phylogenetic type of the language being processed (word-initial vs. word-final affixes, Semitic languages, etc.). However, the selected pattern is common to a large subset of languages (English, Spanish and French, among others), and the procedure can be adapted to accommodate other morphological structures. We opted for this pseudo-suffix based approach because it allows us to capture repeated patterns of different sizes without the need for knowledge-intensive word morphological analysis.

3.3.5 Morfrules: a set of morpho-semantic rules

Using the output of *morfeo* and in combination with our own observation, we manually crafted a set of morphological rules with the purpose of hard-coding the association between certain morphological features and certain semantic classes. Of course, we had to limit ourselves to the most general categories (those in the first three levels of the CoreOnt), as it would be excessively laborious to proceed with each category in this way. The following are some examples of such rules:

```
'property' => 'ment|ness|ity|sion|[ea]nce|ncy',
'science' => 'ics|logy|omy|graphy|[es]try|mics',
'treatment' => 'therapy|scopy',
'surgery' => 'surgery|ctomy',
'instrument' => 'graph|scope|phone|meter',
'disease' => '[ao]sis|itis|pathy|emia|oma',
...
```

These rules mean that if an input word ends with any of the right hand-side patterns (e.g., *cholecystectomy*), then the element on the left-hand side will be promoted as the hypernym. This strategy will only affect a very limited number of terms, but at least it will be effective on such subset.

We have between 20 and 30 rules of this type per language, but we also make them extensive to the hyponyms of the elements on the left. For instance, in the case of hypernym *disease*, hyponyms of said element according to the CoreOnt will also be accepted as hypernym candidates (e.g. *disorder, pathology, affliction, inflammation, infection*).

3.3.6 Head: analysis of the syntactic head of multi-word expressions

If the input term happens to be a multi-word expression, it will be a noun phrase instead of a single noun. In such a case, this module will attempt to identify the syntactic head of the expression.

The procedure is rudimentary but effective on the vast majority of the cases: the last component is the head in the case of English and the first in the case of French and Spanish. The only added rule is, in the case of English, to start the process by eliminating all elements after the first occurrence of the preposition *of*. For instance, in the case of a term such as *acute infectious disease*, the head will be *disease*, but in the case of *disease of lung*, the head that is returned will again be *disease* and not *lung*.

Arguably, to say that the head of the noun phrase of the input is the hypernym is seldom useful. However, it can be useful when the input term cannot be found in corpus or there is insufficient information. In such cases, this module comes into play to obtain the head and then start a new instance of the hypernymy

discovery process, this time to obtain the hypernym of the head. It is likely that a valid hypernym of the head will also be a valid hypernym of the whole term.

3.3.7 Palex: a quantitative implementation of lexical patterns

The use of lexical patterns as a means to extract hypernymy relations from corpus dates back to the work of [22], but is rooted in work previously undertaken on dictionary definitions [8]. A lexical pattern is a sequence such as *is a type of*. In her original paper, Hearst [22] suggested searching for lexical patterns such as these in order to extract hypernymy relations between the nouns or noun phrases found at each side of the pattern, and many researchers used this approach. However, lexical patterns applied in this way tend to be error prone. It is in this context that efforts were made to try to identify which patterns are more reliable [39, 52].

We believe that instead of focusing energy on trying to select a number of patterns, better results could be achieved by a simple change in tactics. Instead of starting with the pattern, i.e. instead of looking up a given pattern in corpus, we propose starting by looking up a given input noun in corpus. Once a large number of contexts of occurrence of a given noun have been extracted, then we suggest searching for the occurrence of lexical patterns within these contexts. These patterns will yield a number of hypernym candidates, and the key of this procedure is precisely the frequency of occurrence of such candidates. There will be one hypernym candidate with the highest frequency, and if the frequency is not high enough (three times at least) then the candidate is rejected and the module does not return any result. This means that results will be produced only on some occasions, but when they are, they will have a high probability of being correct.

The following are some examples of the patterns we compiled for English, and similar patterns were used for Spanish and French.

```
HYPO is a HYPER
HYPO is a type of HYPER
HYPO, a kind of HYPER
HYPO and other HYPER
HYPER such as HYPO
...
```

For economy, many of these patterns are coalesced using regular expression syntax, such that one line of code can include many patterns, like this:

```
is a (type|form|kind) of
```

3.4 The central processor

In the repopulation process, a final decision is needed for the construction of a taxonomy chain for any given noun. This is where the central processor comes into play. It controls the calls to each of the modules explained earlier. The general structure of this module is represented as a flow diagram in Figure 2.

According to the flow chart, given one or more input terms, it starts by taking and analyzing them one-by-one. The symbol a represents thus the input term. At this stage, one could argue that it is necessary to check, in advance, whether the input term a is not already in the Taxonomy T, but that would prevent us from discovering new meanings of an already known term. We will assume, however, that $a \notin T$.

Following the chart, we see it takes the syntactic head c of term a and if $c \in T$, then a result is found and the system moves to the next input term. Otherwise, then the $td()$ function is called, short for top-down. This function, as well as its counterpart $bu()$ (bottom-up), are explained in detail in what follows, but they are nothing more than a particular arrangement of the already presented modules. The symbol h represents a set of hypernym candidates that are obtained as the result of these operations, conducted upon the input

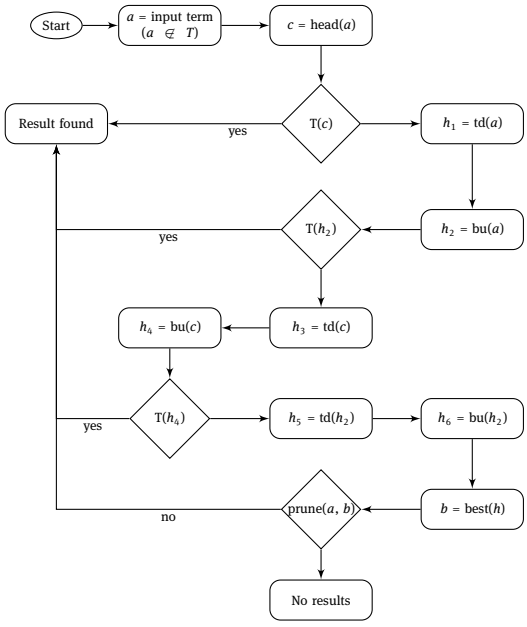

Figure 2: A flow-diagram of the central processor

term a as well as its head c and also to the hypernyms h_i that have already started to appear. At each time of the process, we check if any of the obtained hypernyms are known entities (i.e., $h_i \in T$). The final decision is taken by a ranking method function $best(h)$, which selects the best hypernym-candidate chain according to how many modules voted for them. We compute those values in a matrix R to manage the contribution of each module, and obtain the final score of pairs a and h ($Z(a, h)$) as a weighted sum of each contribution (Equation 12).

$$Z(x, y) = \sum_{i=1}^{|R|} R_i \tag{12}$$

Finally, and before deciding to promote the candidacy of a given hypernym, the system calls function $prune(a, b)$, which takes as arguments the input term and the best hypernym candidate. This function returns a Boolean by calling the pruning algorithm already explained in Section 3.2.

If the input consists of a number of terms, including multi-word expressions, and there are reasons to believe that these terms belong the same specialized domain, there is then room for the application of some forms of pre-processing instead of working with the input terms one-by-one. For instance, the repetition, among this set, of certain head units would be a way to obtain relevant domain-specific semantic categories (e.g. *disorder* in a medical domain). This possibility is left, however, for future work.

3.4.1 The top-down procedure: *td*()

This procedure, done by the central processor, consists of navigating the CoreOnt from its most general nodes to the most specific. That is to say, its output will be a category of the CoreOnt which is as specific as possible, but it never reaches a great level of specificity because the CoreOnt is general by definition.

For instance, if the input term is *fluoxetine*, a successful result for this procedure would be *drug* or even *antidepressant*. It will start by asking, as we said, the most general questions: Is this an entity, a property, or an event? And if it is an entity, is it abstract or concrete?, and so on. This means that, as one travels downwards through the taxonomy, one has to decide which path to traverse next. With the exception of the last one, all

other modules are called for this task. At each step they are called with the input term and the options for categories available at each moment. Each module only votes for one of the categories it is presented with.

3.4.2 The bottom-up procedure: *bu*()

This procedure of the central processor has the purpose of finding the most immediate or domain specific hypernym, and this is the task of the last module, *Palex*. As shown in the flow-chart, the bottom-up procedure will be robust enough to produce results even if the top-down procedure fails, because from an immediate hypernym we can continue to build up a taxonomy chain by recursively submitting the obtained hypernym to a new instance of the top-down procedure. To continue with the same example, if the input term is *fluoxetine*, the result of this procedure should be *selective serotonin reuptake inhibitor*, *SSRI* or, here too, *antidepressant*. And from there, we can start the process again in order to link-up such result with a more general term in the taxonomy.

4 Results

Results are presented in Subsection 4.2 according to language and type of operation (pruning or repopulation). Subsection 4.3 presents the evaluation of the different modules in order to assess their individual and collective contribution. Finally, in Subsection 4.4, results are presented in different datasets than have been used in the literature. We start by describing all the datasets used for the evaluation.

4.1 Datasets for evaluation

We used three different datasets in our experiments: two developed by us and the other by other researchers. We will refer to these datasets by number: 1, 2 and 3.

As already discussed, different datasets have been proposed recently and used in the Semeval-2016 and SemEval-2018 tasks [4, 7]. However, we opted for the generation of a new evaluation dataset because we found a number of inconsistencies in the gold standards that are currently in use. Some of these seem to occur randomly, such as *animal* as a kind of *airplane* or *lexicology* as a *sport*. Others, in contrast, seem to be more systematic, such as the confusion between hypernymy and instance-of relations, (e.g. *Nina Simone* as a hyponym of *person* or *Green Day* as hyponym of *rock band*). These errors were more widespread during the construction of the original WordNet until, later, its developers acknowledged the problem and the need to distinguish between the two types of relations [32]. By then, however, the resource was already populated with assertions such as *Aristotle is a hyponym of philosopher*.

Thus, in order to obtain a high-quality dataset, we embarked on the task of manually curating a new, parallel trilingual dataset (Spanish, English and French), with 925 terms evenly distributed among the semantic categories of *fish*, *machine*, *cheese* and *weapon*. This is Dataset 1. In this dataset, each hyponym-hypernym pair is annotated with a Boolean value indicating its real status. This is because the sample also contains a number of false assertions with elements that are semantically related to the hypernym, but not in a hypernymy relation, like *pizza* and *cheese* or *hatchery* and *fish*. We added this extra difficulty to assess the performance of modules based on distributional properties. We also artificially inflated the number of false assertions by mixing hyponyms and hypernyms randomly and assigning zero values to their matches. This way, from the original set of assertions we obtain a new set of thousands where the false ones outnumber the correct ones by 3 to 1. Of course, this also makes it a harder task, as many of the categories are also semantically related: *cheese* and *fish* are types of *food* and *weapons* and *machines* are types of *artifacts*.

As units of the general vocabulary, this first dataset is restricted to single words only. We did this for simplicity and also to control the effect of hypernyms being the syntactic head of the hyponyms in the case

of multiword expressions, as in *applied linguistics* and *linguistics*. The proposed algorithm does have, however, the capability of dealing with multiword expressions; therefore we also crafted a second dataset of 1767 terms from a specialized domain (psychiatric drugs). All this data was obtained from various sources such as dictionaries and encyclopedias, and each link was revised by pairs. We will refer to this as Dataset 2.

Apart from these two datasets, we compiled a third one from the gold standards shared in the literature. This third dataset is needed in order to compare our results with those of other researchers. It consists of materials taken from multiple sources, including WordNet and the Eurovoc database[3]. We will refer to this as Dataset 3.

4.2 Pruning and repopulation on general vocabulary

Figures of evaluation are estimated on the basis of the results of the automatic contrast against the reference value of Dataset 1, which indicates the real true/false value of each assertion. This allows us to represent the result of each trial in the standard terms of precision, recall and F1. The script that produces the comparison with the gold standard includes what we call a "flexibility table" (flextab, for short), to account for synonyms or otherwise acceptable answers from the algorithm. For instance, in Spanish there are two words for *fish*: *pescado* and *pez*. The first is used to to refer to fish that has been caught and is intended for eating, while the other one is used the denote the animal in general. In the case of edible fish, sometimes the algorithm proposes the hypernym *pescado*, and consequently the term is classified as a type of food. This is a problem of regular polysemy [1] but, with the flextab, the evaluation script will accept both forms as correct, even when it is only the form *pez* that is attested in the gold standard.

Table 1 shows the performance of the pruning and repopulation methods in the case of Dataset 1, the trilingual general vocabulary set. Overall, the algorithm achieved competitive F1 scores across languages and the four semantic classes with an average F1 of 0.82 and 0.83, respectively (the lowest F1 being 0.76 and the highest 0.90). Cross-language performance seems stable, although in French and Spanish the algorithm tends to perform slightly better than in English.

Table 1: Evaluation results of the pruning and repopulation phases of the algorithm using Dataset 1

	Pruning			Repopulation		
Eval Set	**P**	**R**	**F1**	**P**	**R**	**F1**
English	.84	.77	.80	.77	.76	.76
Spanish	.86	.83	.84	.82	.82	.82
French	.81	.85	.82	.93	.89	.90
Average	.84	.82	.82	.84	.82	.83

Figure 3 shows a plot comparing the precision and recall of the pruning algorithm for each language. There, it can be seen that an error within the first positions of the ranking is severely penalized, as is usual in this type of graph. This is the case, in English, of *hatchery*, incorrectly classified as a kind of *fish*. After that, however, precision seems to recover in the three languages and then gracefully decay, in this case after the .40 recall mark. From that point onward, it deteriorates quicker in the case of French, but by a relatively short margin (differences are between 0.9 and 0.8). To facilitate comparison, Figure 4 shows the result of the repopulation algorithm with a bar diagram.

As regards to error analysis, we found that incorrect results fell into three main categories of relatively equal importance in terms of proportion. The first category is made up of cases in which the algorithm proposes as hypernyms elements that bear a semantic relation to the input term without it being a hypernymy

3 https://eur-lex.europa.eu/browse/eurovoc.html

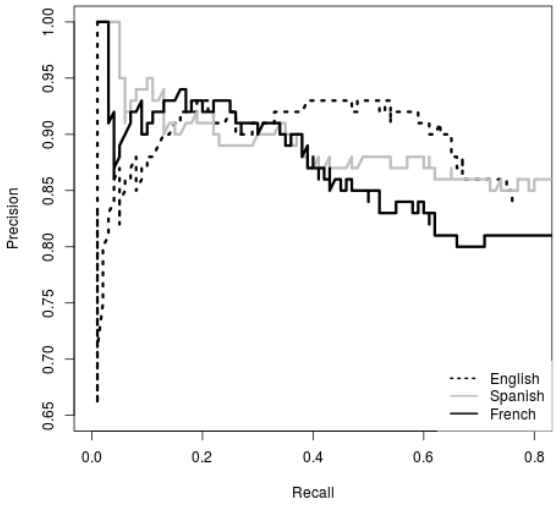

Figure 3: Precision and recall plot of the pruning algorithm across languages using Dataset 1

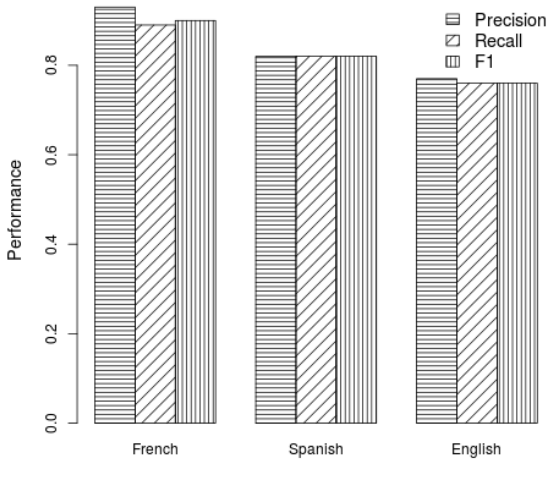

Figure 4: Evaluation of the repopulation algorithm across languages using Dataset 1

type of relation, such as *quesadilla*, a Mexican dish, incorrectly assigned to the category of cheese, or cases of meronymy, like in *scale* which is part of a fish rather than a hyponym of such category.

The second category of error consists of cases that are tagged as errors according to the gold standard and yet could be considered correct, such as *artillery* and *bazooka*, which are classified as machines in disagreement with the gold standard, where they are classified as weapons. The same occurs in French, for instance with *arquebuse* (arquebus) and *mitrailleuse* (machine-gun) and in Spanish with *ametralladora* (machine-gun) among others. One can argue, however, that these are indeed incorrect due to underspecification (the hypernym is not specific enough).

The third type of error consist of those that are attributable to polysemy, that is, when the algorithm produced acceptable results that, however, did not match those in the gold standard. Figure 5 shows a few such cases. Naturally, the algorithm is attracted to the most common meanings, as for example in the case of *horse*, which appears a kind of fish in the gold standard, same as *dory*. The word *tomahawk* appears as a weapon in the gold standard because of the missile. And we also see again cases of regular polysemy, as in the case of *fax*, which appears as a type of machine in the gold standard, but it can also be the document produced by this machine.

4.3 Individual and collective evaluation of the modules

We were interested in measuring the contribution of each individual module to the overall result; therefore, we tested the performance of the algorithm by repeating experiments and switching off one module at a time. Figure 6 shows an evaluation of each module using the leave-one-out method with Dataset 2, the psychotropics gold standard. As already discussed, specialized terminology is better for evaluating the performance of the different modules, since some of them are especially designed to work with multiword expressions. This figure shows how the team performs when one member is not working, and as it can be seen, the overall output seems to be remarkably robust in such scenario. The test shows the relative importance of module *Morfrules* and *Palex*, if any, but none appears to be indispensable.

In Figure 7, in turn, we test the performance of the whole system by switching off more than one module at a time. Here again, we see the system is stable even after the loss of four modules. Excluding more modules results in dramatic loss of recall, but precision remains unaffected.

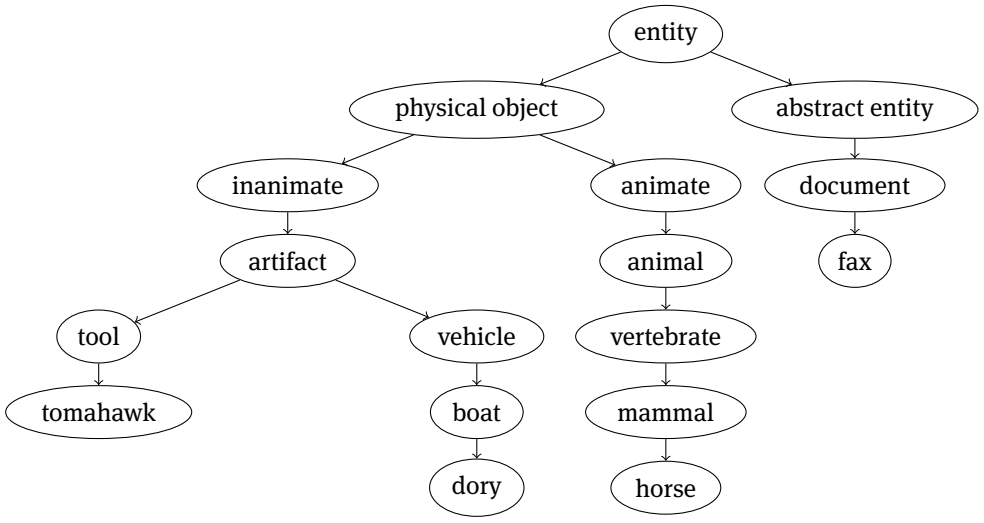

Figure 5: Some examples of results of the algorithm that are in disagreement with the gold standard due to polysemy

4.4 Evaluation with Dataset 3

As already explained, in addition to the dataset prepared by ourselves, we experimented with material compiled by other researchers, in order to obtain a comparative evaluation, and this is what we called Dataset 3. Table 2 shows the results of the repopulation algorithm with various subsets in the three languages. We include again Dataset 2, the psychopharmacology dataset used in Subsection 4.3, to facilitate comparison. Dataset 3 was originally divided into training and test data. In our case, however, as we work with an unsupervised algorithm, we renamed them as corpus 1 and corpus 2 in order to avoid confusion (Medical and Music).

A qualitative analysis of the results reveals that here too the cases of errors fall in the same categories as in Subsection 4.2. Again we encounter confusion between hypernymy and other relations like meronoymy (*airplane* → *engine*) or others (*airfield* → *air*). We also encounter multiple instance of polysemy-related

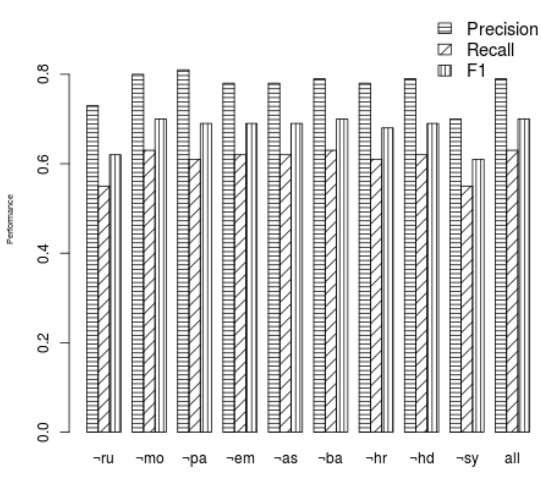

Figure 6: Evaluating the contribution of each module to the overall output by leaving out one at a time, using Dataset 2

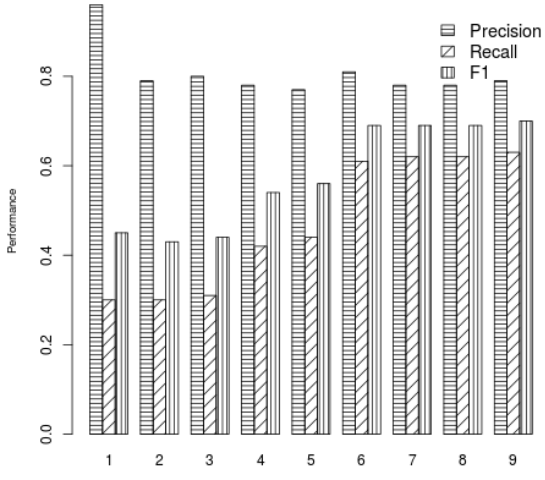

Figure 7: Evaluation of performance with different number of activate modules, using Dataset 2

Language	Eval Set	P	R	F1
Es	Psychopharmacology	.79	.63	.70
En	Medical 1	.55	.51	.52
En	Medical 2	.49	.45	.46
En	Music 1	.48	.44	.45
En	Music 2	.53	.51	.51
Fr	Food	.43	.24	.30
En	Food	.48	.43	.45
Es	General	.39	.38	.38
En	General	.40	.38	.38
En	Science	.73	.70	.71
Fr	Science	.70	.61	.65
Fr	Science WN	**.82**	**.78**	**.79**
Fr	Science Eurovoc	.78	.75	.76

Table 2: Evaluation results of the population algorithm using Dataset 3

problems, such as *classic rock* $\rightarrow$ *rock* $\rightarrow$ *mineral*, etc. or, in French, *guppy* $\rightarrow$ *transport* while the gold standard says it is a type of fish. Finally, we also find here a large number of cases in which our algorithm is correct but the result is tagged as incorrect because of the aforementioned errors in the gold standard. In our results, precision is most affected in the subsets where incorrect links are concentrated, such as General and Food.

Despite the large number of incorrect relations, we see that results still compare well with those reported by other authors. The TAXI system [36], which ranked best in the Semeval-2016 shared task, performs with 0.26 precision and 0.34 recall (0.33 F1) in the Food dataset and with 0.37 precision and 0.38 recall (0.37F1) in the Science dataset in the case of English. The rest of the numbers fall within this range, and their average precision across domains is 0.33 precision and 0.32 recall (0.32 F1). For French, they report an average precision of 0.33 and 0.24 recall (0.28 F1).

The baseline used to compare results is the overlap between hyponym and hypernym. In the best case, this baseline achieved 0.62 precision and 0.28 recall (0.20 F1) in the Science domain in English and 0.87 precision and 0.26 recall (0.40 F1) in the Science domain of Eurovoc in French. The baseline shows a similar pattern across languages and domains: higher precision (0.57 on average in English and 0.58 in French) than recall (0.24 and 0.23 respectively).

With respect to other sets, the CRIM system [7] performed best in English in the SemEval-2018 shared Task, reporting .30 precision in the general corpus, .49 precision in the Medical 2 corpus and .48 precision in Music 2. Our results compare favourably here too, though not by a very large margin. In the case of the Spanish General set, the NLP_HZ system [40] performed best in this task reporting 0.21 precision, a result we improved by a larger margin.

5 Conclusion

This paper proposed an unsupervised algorithm to detect errors in already created taxonomies and to continue with their enlargement or population. Our method combines different modules. Some modules extract hypernyms by exploiting language independent distributional properties of words in large corpora, others measure morphological similarities between co-hyponyms and others use lexical and morphological patterns. The techniques used and described in this paper are relatively simple, most of them based on word co-occurrence. The only language-dependent information used are the lexical patterns. We used the POS-tags already present in the corpus, and there are no grammar or lexicon involved.

We conducted experiments in Spanish, English and French on different evaluation sets, some of which were developed by other researchers, and in addition we produced a large trilingual evaluation dataset. Our results seem competitive with reference to previous research, although it must be said that comparisons should be made with caution because of the previously mentioned problems with the gold standards and the differences in procedure each team uses to evaluate.

Our datasets and code are available for download on the project's website[4]. The code is a single self-contained Perl script with no dependencies, simple to execute on a Linux machine. We encourage NLP researchers to use this code to try to replicate experiments in other languages. This is one of our lines of future work, and we have already started work with Catalan and German. We are also interested to see what kinds of difficulties will arise when working with totally unrelated languages, like those of the Slavic or Semitic families.

Other lines of future work were already mentioned throughout the paper, such as the pre-processing of the input set and experimenting further with different parameter values. The most important and probably most difficult challenge will be to develop a method to deal with polysemy in lexical taxonomies. And this is, indeed, a very attractive line of research, because the fact that the algorithm reports a hypernym that is different from the one reported in a lexical database is something that can be exploited in linguistic research such as semantic neology or semantic change. We believe our algorithm could be adapted to serve as a tool for semantic neology detection.

Acknowledgement: This research has been supported by two successive grants. The first one was Ecos Sud-Conicyt Project C16H02 "Inducción automática de taxonomías del español y el francés mediante técnicas cuantitativas y estadística de corpus" (Automatic taxonomy induction from corpora for Spanish and French using quantitative corpus analysis), lead by Irene Renau (2016-2019). The second one is Project Fondecyt Regular 1191481 "Inducción automática de taxonomías de marcadores discursivos a partir de corpus multilingües" (Automatic induction of taxonomies of discourse markers from multilingual corpora), lead by Rogelio Nazar (2019-2021).

References

[1] Jurij Apresjan, Regular Polysemy, *Linguistics* (1974), 5–32.

[2] Marco Baroni and Alessandro Lenci, Distributional Memory: A General Framework for Corpus-based Semantics, *Comput. Linguist.* **36** (2010), 673–721.

[3] David M Blei, Andrew Y Ng and Michael I Jordan, Latent dirichlet allocation, *Journal of Machine Learning Research* **3** (2003), 993–1022.

[4] Georgeta Bordea, Els Lefever and Paul Buitelaar, SemEval-2016 Task 13: Taxonomy extraction evaluation (texeval-2), in: *SemEval-2016*, Association for Computational Linguistics, pp. 1081–1091, 2016.

[5] Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra and Jenifer C. Lai, Class-Based n-gram Models of Natural Language, *Computational Linguistics* **18** (1992), 467–479.

[6] John Bullinaria, Semantic Categorization Using Simple Word Co-occurrence statistics, in: *ESSLLI Workshop on Distributional Lexical Semantics*, Hamburg, Germany, 2008.

[7] José Camacho-Collados, Claudio Delli Bovi, Luis Espinosa Anke, Sergio Oramas, Tommaso Pasini, Enrico Santus, Vered Shwartz, Roberto Navigli and Horacio Saggion, SemEval-2018 Task 9: Hypernym Discovery, in: *Proceedings of The 12th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT, New Orleans, Louisiana, June 5-6, 2018*, pp. 712–724, 2018.

[8] Martin Chodorow, Roy Byrd and George Heidorn, Extracting semantic hierarchies from a large on-line dictionary, in: *Proc. of the 23rd annual meeting on ACL (Chicago, Illinois, USA)*, pp. 299–304, 1985.

[9] Kenneth Ward Church and Patrick Hanks, Word Association Norms, Mutual Information, and Lexicography, *Comput. Linguist.* **16** (1990), 22–29.

4 http://www.tecling.com/kind

[10] Philipp Cimiano and Johanna Völker, Text2Onto: A Framework for Ontology Learning and Data-driven Change Discovery, in: *Proceedings of the 10th International Conference on Natural Language Processing and Information Systems*, NLDB'05, pp. 227–238, Springer-Verlag, Berlin, Heidelberg, 2005.
[11] Daoud Clarke, Context-theoretic Semantics for Natural Language: An Overview, in: *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, GEMS '09, pp. 112–119, Association for Computational Linguistics, Stroudsburg, PA, USA, 2009.
[12] Ronan Collobert and Jason Weston, A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning, in: *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pp. 160–167, ACM, New York, NY, USA, 2008.
[13] David Cruse, *Lexical Semantics*, Cambridge University Press, Cambridge, UK, 1986.
[14] Gerard de Melo, Mohit Bansal, David Burkett and Dan Klein, Structured Learning for Taxonomy Induction with Belief Propagation, in: *Proceedings of ACL*, Baltimore, Maryland, USA, June 2014.
[15] Susan T Dumais, George W Furnas, Thomas K Landauer, Scott Deerwester and Richard Harshman, Using latent semantic analysis to improve access to textual information, in: *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pp. 281–285, 1988.
[16] Luis Espinosa Anke, Jose Camacho-Collados, Claudio Delli Bovi and Horacio Saggion, Supervised Distributional Hypernym Discovery via Domain Adaptation, in: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 424–435, Association for Computational Linguistics, Austin, Texas, November 2016.
[17] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin and Tomas Mikolov, Learning Word Vectors for 157 Languages, in: *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.
[18] Gregory Grefenstette, *Explorations in Automatic Thesaurus Discovery*, Kluwer Academic Publishers, Norwell, MA, USA, 1994.
[19] Louise Guthrie, Brian Slator, Yorick Wilks and Rebecca Bruce, Is there content in empty heads?, in: *Proc. of the 13th International Conference on Computational Linguistics, COLING'90 (Helsinki, Finland)*, pp. 138–143, 1990.
[20] Patrick Hanks and James Pustejovsky, A Pattern Dictionary for Natural Language Processing, *Revue Francaise de Langue Appliquée* **10** (2005).
[21] Zellig Harris, Distributional structure, 10, pp. 146–162, 1954.
[22] Marti A. Hearst, Automatic Acquisition of Hyponyms from Large Text Corpora, in: *Proceedings of the 14th Conference on Computational Linguistics - Volume 2*, COLING '92, pp. 539–545, Association for Computational Linguistics, Stroudsburg, PA, USA, 1992.
[23] Timo Honkela, Self-organizing maps of words for natural language processing applications, in: *In Proceedings International ICSC Symposium on Soft Computing*, 1997.
[24] Miloš Jakubíček, Adam Kilgarriff, Vojtěch Kovář, Pavel Rychlý and Vít Suchomel, The TenTen Corpus Family, in: *7th International Corpus Linguistics Conference CL 2013*, pp. 125–127, Lancaster, 2013.
[25] Lili Kotlerman, Ido Dagan, Idan Szpektor and Maayan Zhitomirsky-geffet, Directional Distributional Similarity for Lexical Inference, *Nat. Lang. Eng.* **16** (2010), 359–389.
[26] Alessandro Lenci and Giulia Benotto, Identifying hypernyms in distributional semantic spaces, in: **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pp. 75–79, Association for Computational Linguistics, 2012.
[27] Dekang Lin, Automatic Retrieval and Clustering of Similar Words, in: *Proceedings of the 17th International Conference on Computational Linguistics - Volume 2*, COLING '98, pp. 768–774, Association for Computational Linguistics, Stroudsburg, PA, USA, 1998.
[28] Dekang Lin and Xiaoyun Wu, Phrase clustering for discriminative learning, 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*, pp. 1030–1038, Suntec, Singapore, 2009.
[29] Kevin Lund and Curt Burgess, Producing high-dimensional semantic spaces from lexical co-occurrence, *Behavior Research Methods, Instruments, & Computers* **28** (1996), 203–208.
[30] Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean, Efficient Estimation of Word Representations in Vector Space, *CoRR* **abs/1301.3781** (2013).
[31] George A. Miller, WordNet: A Lexical Database for English, *COMMUNICATIONS OF THE ACM* **38** (1995), 39–41.
[32] George A. Miller and Florentina Hristea, Squibs and Discussions: WordNet Nouns: Classes and Instances, *American Journal of Computational Linguistics* **32** (2006), 1–3.
[33] Rogelio Nazar, *A Quantitative Approach to Concept Analysis*, Ph.D. thesis, Universitat Pompeu Fabra, 2010.
[34] Rogelio Nazar and Irene Renau, A Taxonomy of Spanish Nouns, a Statistical Algorithm to Generate it and its Implementation in Open Source Code, in: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 1485–1492, European Language Resources Association (ELRA), Portorož, Slovenia, May 2016.
[35] Lluís Padró, Samuel Reese, Eneko Agirre and Aitor Soroa, Semantic Services in FreeLing 2.1: WordNet and UKB, in: *Principles, Construction, and Application of Multilingual Wordnets* (Pushpak Bhattacharyya, Christiane Fellbaum and Piek Vossen, eds.), Global Wordnet Conference 2010, pp. 99–105, Narosa Publishing House, Mumbai, India, February 2010.

[36] Alexander Panchenko, Stefano Faralli, Eugen Ruppert, Steffen Remus, Hubert Naets, Cédrick Fairon, Simone Paolo Ponzetto and Chris Biemann, TAXI at SemEval-2016 Task 13: a Taxonomy Induction Method based on Lexico-Syntactic Patterns, Substrings and Focused Crawling, in: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 1320–1327, Association for Computational Linguistics, San Diego, California, June 2016.

[37] Patrick Pantel and Marco Pennacchiotti, Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations, in: *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pp. 113–120, Association for Computational Linguistics, Stroudsburg, PA, USA, 2006.

[38] Fernando Pereira, Naftali Tishby and Lillian Lee, Distributional clustering of English words, in: *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pp. 183–190, Columbus, USA, 1993.

[39] Alessandra Potrich and Emanuele Pianta, L-ISA: Learning Domain Specific Isa-Relations from the Web., in: *LREC*, European Language Resources Association, 2008.

[40] Wei Qiu, Mosha Chen, Linlin Li and Luo Si, NLP_HZ at SemEval-2018 Task 9: a Nearest Neighbor Approach, in: *Proceedings of The 12th International Workshop on Semantic Evaluation*, pp. 909–913, Association for Computational Linguistics, New Orleans, Louisiana, June 2018.

[41] Thomas Rebele, Fabian M. Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey and Gerhard Weikum, YAGO: A Multilingual Knowledge Base from Wikipedia, Wordnet, and Geonames., in: *International Semantic Web Conference (2)* (Paul T. Groth, Elena Simperl, Alasdair J. G. Gray, Marta Sabou, Markus Krötzsch, Freddy Lécuée, Fabian Flöck and Yolanda Gil, eds.), Lecture Notes in Computer Science 9982, pp. 177–185, 2016.

[42] Radim Řehůřek and Petr Sojka, Software framework for topic modelling with large corpora, in: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pp. 51–55, Valetta, Malta, 2010.

[43] Laura Rimell, Distributional Lexical Entailment by Topic Coherence, in: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 511–519, Association for Computational Linguistics, 2014.

[44] Sara Rydin, Building a Hyponymy Lexicon with Hierarchical Structure, in: *Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition - Volume 9*, ULA '02, pp. 26–33, Association for Computational Linguistics, Stroudsburg, PA, USA, 2002.

[45] Juan C. Sager, *A Practical Course in Terminology Processing*, John Benjamins, Amsterdam/Philadelphia, 1990.

[46] Magnus Sahlgren, The distributional hypothesis, *Rivista di Linguistica* **20** (2008), 33–53.

[47] Gerard Salton and Michael J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, Inc., New York, NY, USA, 1986.

[48] Enrico Santus, Alessandro Lenci, Tin-Shing Chiu, Qin Lu and Chu-Ren Huang, Unsupervised Measure of Word Similarity: How to Outperform Co-Occurrence and Vector Cosine in VSMs, in: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pp. 4260–4261, 2016.

[49] Enrico Santus, Alessandro Lenci, Qin Lu and Sabine Schulte Im Walde, Chasing hypernyms in vector spaces with entropy, in: *In Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pp. 38–42, 2014.

[50] Rajdeep Sarkar, John Philip McCrae and Paul Buitelaar, A supervised approach to taxonomy extraction using word embeddings, in: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, European Language Resources Association (ELRA), Miyazaki, Japan, May 7-12, 2018 2018 (english).

[51] Hinrich Schütze and Jan O. Pedersen, A Cooccurrence-based Thesaurus and Two Applications to Information Retrieval, *Inf. Process. Manage.* **33** (1997), 307–318.

[52] Julian Seitner, Christian Bizer, Kai Eckert, Stefano Faralli, Robert Meusel, Heiko Paulheim and Simone Paolo Ponzetto, A Large DataBase of Hypernymy Relations Extracted from the Web., in: *LREC*, European Language Resources Association (ELRA), 2016.

[53] Vered Shwartz, Enrico Santus and Dominik Schlechtweg, Hypernyms under Siege: Linguistically-motivated Artillery for Hypernymy Detection, in: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pp. 65–75, Association for Computational Linguistics, 2017.

[54] Rion Snow, Daniel Jurafsky and Andrew Y. Ng, Semantic Taxonomy Induction from Heterogenous Evidence, in: *Proceedings of the 21st International Conference on Computational Linguistics, Sydney, Australia, 17-21 July 2006*, 2006.

[55] Peter D. Turney and Patrick Pantel, From frequency to meaning: Vector space models of semantics, *Journal of Artificial Intelligence Research* **37** (2010), 141–188.

[56] Julie Weeds and David Weir, A General Framework for Distributional Similarity, in: *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, 2003.