The Knowledge Engineering Review, Vol. 36, e13, 1 of 29.
doi:10.1017/S0269888921000126

A 3-phase approach based on sequential mining and dependency parsing for enhancing hypernym patterns performance

AHMAD ISSA ALAA ALDINE[1,3], MOUNIRA HARZALLAH[2], GIUSEPPE BERIO[1], NICOLAS BÉCHET[1], and AHMAD FAOUR[3]

[1]*University Bretagne Sud, IRISA Lab, France – Vannes*
Email: ahmad.issa-alaa-eddine@univ-ubs.fr, giuseppe.berio@univ-ubs.fr, nicolas.bechet@irisa.fr
[2]*LINA - University of Nantes, France*
E-mail: mounira.harzallah@univ-nantes.fr
[3]*Lebanese University, Lebanon*
Email: ahmad.faour@ul.edu.lb

Abstract

Patterns have been extensively used to extract hypernym relations from texts. The most popular patterns are Hearst's patterns, formulated as regular expressions mainly based on lexical information. Experiences have reported good precision and low recall for such patterns. Thus, several approaches have been developed for improving recall. While these approaches perform better in terms of recall, it remains quite difficult to further increase recall without degrading precision. In this paper, we propose a novel 3-phase approach based on sequential pattern mining to improve pattern-based approaches in terms of both precision and recall by (i) using a rich pattern representation based on grammatical dependencies (ii) discovering new hypernym patterns, and (iii) extending hypernym patterns with anti-hypernym patterns to prune wrong extracted hypernym relations. The results obtained by performing experiments on three corpora confirm that using our approach, we are able to learn sequential patterns and combine them to outperform existing hypernym patterns in terms of precision and recall. The comparison to unsupervised distributional baselines for hypernym detection shows that, as expected, our approach yields much better performance. When compared to supervised distributional baselines for hypernym detection, our approach can be shown to be complementary and much less loosely coupled with training datasets and corpora.

1 Introduction

Due to the permanent growth of digital textual resources, especially on the web, automatic knowledge discovery or extraction from texts become necessary to address many open research problems and applications such as information retrieval (Chandramouli *et al*. 2008), question answering (Cui *et al*. 2007; Zheng *et al*. 2019), and ontology learning (Gomez-PÉrez & Manzano-Mancho 2004; Camacho-Collados *et al*. 2018). In this paper, we focus our interest on the automatic extraction of hypernym relations from texts. Broadly speaking, hypernym relation is a semantic relationship between two terms: y is a hypernym of x (x is a hyponym of y) means that x is a subordinate term of y—e.g. 'musical instrument' is a hypernym of 'piano'. Throughout the paper, we refer to the extraction of a hypernym relation between a hyponym and hypernym as the extraction of one Hyponym Hypernym pair (HH-pair).

For finding such relationships expressed in texts, two distinct kinds of approaches have been proposed and used: distributional and pattern-based.

Earlier distributional approaches are unsupervised and based on the *'distributional inclusion hypothesis'* (Weeds & Weir 2003; Kotlerman *et al*. 2010). Recent distributional approaches are supervised relying on word embedding (Mikolov *et al*. 2013; Pennington *et al*. 2014; Devlin *et al*. 2019; Yang *et al*. 2020) to represent feature vectors for pairs of terms potentially related by hypernym relation. In general, supervised distributional approaches perform better than the unsupervised ones. However, Levy *et al*. (2015) show that the good performance of supervised approaches is due to 'lexical memorization'. Specifically, they show that supervised approaches memorize whether term *y* is a prototypical hypernym, regardless of term *x*, rather than learning the hypernym relation between *x* and *y*.

Pattern-based approaches are based on sets of patterns, where patterns are used to match sentences and extract HH-pairs. Each pattern comprises specific placeholders for hypernym and hyponym, needed for extracting HH-pair(s) found in any sentence matching with that pattern. For instance, sentence 'I like musical instruments such as piano' matches the pattern *'y such as x'* where placeholders x and y are bound to noun phrases occurring in the sentence, thus suggesting (piano, musical instrument) as one HH-pair. Patterns can be either designed manually (Hearst 1992) or extracted automatically (Snow *et al*. 2005). Most of the existing patterns show quite high precision but recall remains very low because of the large variability in natural language for expressing a given meaning. On the one hand, the manual extension of existing patterns for improving recall while keeping a good precision is quite hard—once again because of the large variability in natural language for expressing a given meaning. On the other hand, to the best of our knowledge, all approaches trying to automatically extract hypernym patterns for getting both good precision and recall, propose to use dependency paths (Snow *et al*. 2005; Sheena *et al*. 2016) as features to train classifiers. However, using dependency paths (i.e paths of grammatical dependencies found in sentences) as features leads to a huge and sparse feature space, which decreases precision without significant gain in recall. Additionally, even if patterns are often considered valid across domains, for keeping good recall, it is necessary to continuously discover new patterns. Furthermore, patterns are also closely related to a natural language so each language needs specific patterns (you need English, French, Italian,. . . patterns). As a consequence, to overcome major limitations of patterns mentioned above, effective mechanisms for continuous discovering of patterns should be developed.

According to (Mirkin *et al*. 2006; Shwartz *et al*. 2016; Yu *et al*. 2020), pattern-based and distributional approaches can be considered complementary, and combining them enables to extract more knowledge from the same texts. This is natural because patterns extract the underlying meaning of sentences; distributional approaches focus on frequent occurrences of terms in the same contexts (or some mathematical combinations of term contexts), which are partially disconnected from the sentence meaning. Thus, one pair extracted by pattern may not be extracted by using any distributional approach and vice versa. However, even if patterns and distributional approaches can be considered as complementary, various works have compared them. Roller *et al*. (2018) propose 3 statistical models to detect hypernymy based on Hearst's patterns. Their results show that their pattern-based models outperform unsupervised distributional approaches, and confirm that Hearst's patterns provide good precision mainly on large corpora. Finally, patterns are recognized as easily understandable and explainable while distributional approaches are largely black-box. The 3 aspects shortly introduced above, i.e. Complementarity, understandability, explainability, constitute the raison for the work presented in this paper, focused on patterns.

In this work, we propose a new 3-phase (supervised) approach for automatically extracting patterns, from any corpus and any dataset of known HH-pairs. The main objective of the approach is to provide a mechanism for systematically (and continuously) refining and adding patterns increasing both recall and precision of some seed patterns given as input. The approach is based on sequential pattern mining (using CloSPEC (BÉchet *et al*. 2015)), coupled with both a rich syntactical sentence representation based on grammatical dependencies and a pattern selection strategy for automatically removing the huge amount of raw patterns extracted by the mining algorithm.

The 3 phases are distinguished in term of target precision and recall to be achieved by the extracted patterns, as described below:

(i) Phase 1 mines sequential patterns for improving precision of some seed patterns; recall of extracted patterns remains comparable (or the same) as the recall of seed patterns;

(ii) Phase 2 improves recall of patterns resulting from phase 1 by discovering new sequential patterns other than sequential patterns discovered in phase 1; new patterns extract HH-pairs that cannot be extracted neither by seed patters nor by extracted patterns resulting from phase 1; additionally, new patterns do not decrease significantly the precision of seed patterns;

(iii) Phase 3 found anti-hypernym patterns, for removing wrong pairs (i.e. pairs not in the dataset of known HH-pairs) extracted by seed patterns and patterns resulting from phases 1 and 2; an anti-hypernym pattern is defined as a pattern representing any relation other than hypernym-hyponym relation; precision of the whole set of patterns+anti-patterns is then improved and recall remains constant.

The 3 phases are required because precision and recall move in opposite directions: as a consequence, each phase improves precision (or recall) till recall (or precision) is not degraded, and each phase extracts patterns from distinct sentences in the corpus. The third phase, targeting anti-hypernym patterns, is needed because we noted that if only first and second phases are performed, precision improvement remains limited, despite any modification of parameters and partitions of training and testing sentences. By the way, the usage of anti-hypernym patterns is not new as reported in the related work section below.

We validate the proposed approach by performing two sets of experiences with three distinct corpora (one domain-specific corpus on music, and two generic English corpora) and a specific set of seed patterns. Seed patterns used for experiences are Dependency Hearst's Patterns (noted **DHPs**), manually specified in a previous work (Aldine *et al*. 2018), and showing a precision comparable to precision of Hearst's patterns and a much better recall. The first phase results in Sequential Hearst's Patterns (**SHPs**) as we named them. The second phase then takes as input SHPs and results in new additional Sequential Hypernym Patterns (**SHyPs**). Finally, the third phase takes as input SHPs and SHyPs and generates anti-hypernym patterns, respectively noted as (**SHPs**$^-$ and (**SHyPs**$^-$. The first set of experiences enables to quantitatively show (i) the better performance of extracted patterns over collections of existing patterns (Hearst 1992; Seitner *et al*. 2016) (ii) the key contribution of anti-hypernym patterns for improving precision. We also show genericity of the extracted patterns i.e. to what extent one pattern extracted by using a corpus can be reused to extract HH-pairs from another corpus. Genericity of patterns can be quantitatively observed by performing some experiments. However, it can also be qualitatively assessed by directly inspecting the patterns, being patterns both explainable and understandable. The second set of experiences compares performances of extracted patterns to performances of known supervised and unsupervised distributional baselines using the three corpora mentioned above and three popular datasets (BLESS, EVALution, and Weeds) comprising known HH-pairs. The obtained results show that extracted patterns offer much better performance than unsupervised baselines. We show that even if the performance of the supervised distributional baselines is often better than our extracted patterns, patterns and supervised distributional baselines are complementary with a significant percentage of HH-pairs extracted by only one of them. Genericity of patterns is once again assessed, showing the advantage of patterns over supervised distributional baselines.

The rest of the paper is organized as follows. In Section 2, we present an overview of related work. In Section 3, we present and describe the details of the proposed 3-phase approach. In Section 4, we describe the context for conducting experiences: for that, we will provide details on how the 3-phase approach has been implemented and executed starting from DHPs patterns as seed patterns. In the same section, we present the results of the first set of experiences for validating the approach. Section 5 is devoted to compare extracted patterns with unsupervised and supervised distributional approaches. Paper ends with conclusions and perspectives.

2 Related work

The first pattern-based approach was proposed by Hearst (1992, 1998), who manually defined a set of lexico-syntactic patterns (i.e. regular expressions comprising lexical information such as some words and syntactical information such as POS tags like noun, verb, subject) suggesting HH-pairs in a sentence. Although Hearst's Patterns (**HPs**) yield good precision, they suffer from low recall (Buitelaar *et al*. 2005).

Indeed, there are several ways to express the same hypernym relation in a text while Hearst's patterns are few (6 patterns). Various approaches have been proposed to increase the recall of Hearst's patterns. These approaches extend Hearst's patterns either manually defining new patterns (Jacques & Aussenac-Gilles 2006; Orna-Montesinos 2011; Seitner *et al*. 2016) or automatically extracting new patterns (Snow *et al*. 2005). For instance, new patterns have been proposed by manually replacing words by other similar words (e.g. the Hearst's pattern '*y* such as *x*' becomes '*y* like *x*' when replacing 'such as' by 'like') or adding new words (e.g. adding 'any' to '*x* and other *y*' leads to '*x* and any other *y*'). Orna-Montesinos (2011) manually specifies some variations of Hearst's patterns and new patterns, starting from a dataset comprising known HH-pairs where the noun 'building' is the hypernym. As usual, specified patterns are those frequently matching with sentences expressing hypernym hyponym relations and comprising any HH-pair found in the dataset. However, some patterns require grammatical annotations of sentences to be matched: e.g. pattern '*x* superlative *y*' matches 'St. Peter's Cathedral in Rome, the most important building of the period' and extracts the HH-pair (St. Peter's Cathedral, building) but superlative annotation needs to be added when processing the sentence. Recently, Seitner *et al*. (2016) propose a quite wide set of patterns (59 patterns, referred in the remainder as **ExtHPs**) collected from the literature for building a large database of HH-pairs extracted from the web.

Generally speaking, manual approaches enable to increase recall but at the same time precision decreases. For instance, the pattern '*x* sort of *y*' has only 14% of precision because several counterexamples can be found (Seitner et al. 2016). Moreover, manual approaches tend to use simple regular expressions for representing patterns, constituted by sequences of POS tags, natural words, and placeholders, easily understandable (but also efficient for matching sentences and patterns). Unfortunately, this under-specification (related to regular expressions) often leads to erroneous interpretations. Recently, in Aldine *et al*. (2018), we try to understand the benefits of using grammatical dependencies for specifying patterns. For that purpose, we manually reformulate the original Hearst's patterns as dependency patterns (**DHPs**) using the Stanford dependency parser (Klein & Manning 2003). For instance, Hearst's pattern 'y such as x' can be transformed to DHP 'case(yHead, such) & nmod:such as(xHead, yHead)', where x and y are noun phrases and xHead and yHead' are the corresponnding x and y 'headwords' respectively (see section 3.5 for definition of headword and examples). DHPs have a better (but still low) recall than HPs and ExtHPs (20% more). Indeed, DHPs generalize HPs and some of ExtHPs. However, they show slightly worse precision because they are not constrained by term positions within a sentence, thus leading to errors, especially for long sentences. This means that once again despite the richer pattern representation (grammatical dependencies), manually specifying patterns is likely to lead to decrease precision.

Automated pattern extraction approaches (Snow *et al*. 2005; Sheena *et al*. 2016) have been developed for finding more and better patterns, when compared to manual approaches. Most of them consider richer pattern representations based on grammatical dependencies (additional to lexical information)[1]. More specifically, the shortest dependency paths linking nouns found in any training sentence and in a dataset of HH-pairs correspond to features. These features are then used for building classifiers, and dependency paths with the highest weights in the classifier are considered as hypernym patterns. For instance, three paths can be found in sentence 'I like musical instruments such as piano' for HH-pair (instrument, piano): '*x*: nmod:such_as: *y*'; '*musical*, *JJ* : *amod* : *x*, *x* : *nmod* : *such_as* : *y*′; *and*'*x* : *nmod* : *such_as* : *y*, *y* : *case* : *JJ*, *such*′. The shortest path is considered as the one better expressing the relation between the two nouns (however, this path is considered as a feature if it links at least five training HH-pairs with less than four words in between).

By using automated pattern extraction approaches, some of HPs and new patterns with respect to HPs have been respectively rediscovered and discovered, naturally increasing recall. However, the usage as a feature of a single path per HH-pair, leads to a sparse feature space, negatively affecting the performance of the trained classifier, thus negatively impacting precision. In the context of automated pattern extraction, (Sang & Hofmann 2009) have tried to assess the impact of pattern representations using

[1] In Snow *et al*. (2005), paths are derived from Minipar parser (Lin 2003) which is a shallow parser; while in Sheena *et al*. (2016), paths are derived from Stanford parser (Klein & Manning 2003) which is a dependency parser.

grammatical dependencies over simpler pattern representations using POS tags and lexical information only. They follow the supervised approach proposed by Snow *et al*. (2005) for building two distinct classifiers: a lexical-based classifier (where lexical paths are features) and a dependency-based classifier (where shortest dependency paths are features). The main point is that dependency patterns have a slightly better recall with a slight reduction in precision. This is exactly the same result that we achieved by manually redefining HPs as DHPs, as reported above. This means that dependencies can be interesting (because recall is anyway improved) but dependencies should be used in other ways for getting the advantage of the more precise syntactical information they enable to represent.

Meronymy patterns (i.e. patterns for 'part-of' relation extraction) have been used by Ponzetto & Strube (2011) for increasing precision. The idea is to remove extracted pairs identified by using patterns corresponding to a relation which does not represent hypernymy. Meronymy patterns are examples of anti-hypernym patterns, explicitly representing specific counterexamples. The usage of any mechanism (such as meronymy patterns) for carefully filtering the outcomes (the extracted pairs) of a given set of patterns is an interesting idea for increasing precision. Recently, Roller *et al*. (2018) take an extended set of Hearst's patterns and propose an approach based on the frequency of extracted pairs. They propose three different models to compute one hypernym score for any extracted pair. The first model is quite simple: the score exactly corresponds to the extraction frequency. The second model computes the score by using Positive Point Mutual Information (PPMI). The third model corresponds to *Singular Value Decomposition* (SVD) applied to the whole PPMI matrix (comprising all pairs), resulting in a truncated matrix on which the score is calculated. The results confirm that these models outperform unsupervised distributional approaches. However, we consider that an approach using only statistical information (such as a score) is less controllable and is constraining (because a pair to be recognized as HH-pair needs to frequently occur in sentences matching with the given set of patterns) that an approach based on anti-patterns, which underlying idea is quite simple and natural. For this reason, the proposed 3-phase approach adopts anti-patterns as the main mechanism for filtering extracted pairs, without preventing the usage of other mechanisms.

Taking into account the difficulties shortly highlighted in this section for increasing precision and recall of patterns, we have considered a distinct technique for learning patterns. We have naturally focused our attention on algorithms for sequential pattern mining (SPM). SPM is a task of mining sequential patterns from domains where ordering is important as it seems needed for better using grammatical dependencies. The first algorithm (AprioriAll) was first introduced by Agrawal & Srikant (1995). Several SPM algorithms have then been proposed such as GSP (Srikant & Agrawal 1996), PrefixSpan (Pei *et al*. 2001), CloSpan (Yan *et al*. 2003), and Bide (Wang & Han 2004). CloSpan and Bide were proposed for mining only closed sequential patterns (a sequential pattern is closed if it is not strictly included in another pattern having the same support). More recently, BÉchet *et al*. (2015) described a new algorithm (CloSPEC) for mining closed sequential patterns under multiple constraints such as gap and number of itemsets. The usage of SPM for extracting ad-hoc relations (i.e. not hypernym relations) from text is promising. Nguyen *et al*. (2007) propose a framework to extract ad-hoc relation patterns from Wikipedia using PrefixSpan. First, they use dependency parser to extract shortest dependency paths between given pairs of terms. Then, they represent each dependency path as a sequence to build a sequential database, and PrefixSpan is applied to the sequential database. Bechet *et al*. (2012) extend previous work (Cellier *et al*. 2010) by representing each word in a sentence as an itemset comprising lexical information and POS tags and then applying CloSPEC. Despite some difficulties related to sequential pattern mining such as the huge number of extracted patterns (that, however, in our 3-phase approach are automatically pruned by using some criteria and thresholds), a major benefit of using SPM is that it corresponds to a conceptually simple task of accounting exemplary sentences. Therefore, the proposed approach adopts SPM (specifically, CloSPEC) for extracting patterns. To maximize benefits of adopting SPM, we introduce a new pattern representation based on lexical information and grammatical dependencies. Indeed, we have shown in this state of the art review that using dependencies is interesting (because recall increases) but their free usage (e.g. unordered dependencies or using only some dependencies targeting shortest paths) may have a negative impact on precision.

3 The 3-phase approach

In this section, we describe in detail the 3-phase (supervised) approach that aims to systematically increase the precision and recall of some seed patterns, using any corpus and any dataset of known HH-pairs. Each phase is performed by following the same workflow steps, so that the steps are presented in a dedicated Section (3.4). The new pattern representation based on lexical information and grammatical dependencies is also presented and compared to pattern representations found in the relevant literature. It should be noted that the approach is generic and can be used with any set of seed patterns, independently of their representation, any corpus, and any dataset of expected HH-pairs. It should also be noted that the workflow is generic too, so that it is suitable to develop an adaptable and modular software conforms to the suggested design and configurations.

3.1 Phase 1: Mining sequential patterns associated to seed patterns

Phase 1 objective is to extract sequential patterns (*SPs*) corresponding to a given set of seed patterns such that extracted patterns improve precision of the seed patterns while keeping a comparable (or the same) recall. This is done, first, by applying the mining process for extracting *SPs* from a set of training sentences expressing hypernym relations, being these sentences matching any seed pattern and extracting any HH-pair; second, by automatically selecting the *SPs* showing better precision than precision of seed patterns and a comparable (or the same) recall.

3.2 Phase 2: Discovering new sequential patterns

Phase 2 objective is to discover new *SPs* (additional to patterns extracted in phase 1) for the purpose of increasing recall under the constraint of keeping precision stable (i.e. close to the precision of *SPs* of phase1). These new patterns are discovered by applying the mining process for extract *SPs* from sentences expressing hypernym relations but not matching with patterns extracted in phase 1. Discovered patterns are then further selected for keeping precision at least equal to precision of patterns extracted in phase 1.

3.3 Phase 3: Anti-hypernym sequential patterns

Phase 3 objective is to extract patterns enabling to identify pairs wrongly extracted as HH-pairs by *SPs* resulting from phase 1 or phase 2. Patterns extracted in phase 3 are often qualified as anti-patterns, thus referred here as anti-hypernym sequential patterns. Using both patterns and anti-patterns in combination contributes to increase precision while keeping recall unvaried. Anti-patterns are extracted by applying the mining process for extracting *SPs* from sentences not expressing hypermym relations and matching with patterns found in phase 1 and phase 2. Discovered patterns are then further selected among the ones showing lowest precision (i.e. selected patterns are likely to extract non-hypernym pairs).

3.4 Sequential hypernym patterns learning workflow

The steps required to extract sequential patterns are common to the 3 phases. This is an important aspect of the proposed approach. Figure 1 shows the generic 5 steps workflow: (i) Corpus Labeling, (ii) Sequences Preparation, (iii) SPM, (iv) Relevant Sequential Pattern Selection and (v) Evaluation. Some steps are executed only once (the ones above the dotted line in Figure 1) and other generic steps (the ones below the dotted line in Figure 1) repeated for each phase and requiring to be configured in terms of inputs and constraint parameter values, according to the objective of each specific phase.

3.4.1 Corpus labeling step and evaluation Step

Usually, hypernym patterns performance is evaluated in terms of precision and recall based on extracted pairs and expected HH-pairs found in a dataset. However, for patterns, we consider that this evaluation is partially inaccurate because (1) on the one hand, (right) patterns can match sentences indicating

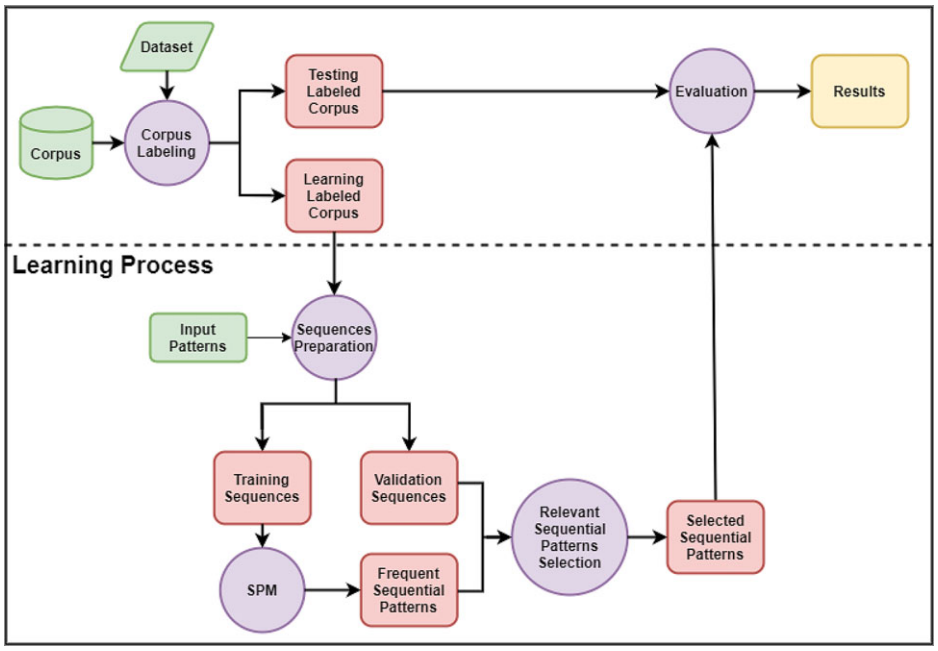

Figure 1. The workflow of sequential patterns learning.

hypernym relations and extract HH-pairs which are not among expected HH-pairs, (2) on the other hand, (wrong/imprecise) patterns can match sentences that do not express hypernym relations but extract expected HH-pairs. The first point can be traced back to misalignment between the corpus and the dataset, which can quite often be observed. As a consequence and for limiting as much as possible any human intervention (because of the huge number of sentences to be checked out), we suggest to proceed in two ways: first, virtually completing the dataset by using HH-pairs suggested by WordNet (Fellbaum 1998) (but other resources can be used); second, manually classifying the remaining extracted pairs covered neither by the dataset nor WordNet. With these ways to proceed, the evaluation does not require heavy human interventions.

For taking into account the second point, we label each sentence in a corpus C as positive or negative. A positive sentence expresses at least one hypernym relation according to a given set of HH-pairs, a negative sentence is not a positive one. The set of positive sentences (PS) and the set of negative sentences (NS) in C are defined as follows:

$$PS = \{s \mid s \in C,\ \exists\, (x, y) \in \textit{HH−pairs s expresses hypernym relation between x and y}\} \tag{1}$$

$$NS = C - PS \tag{2}$$

where *'s expresses hypernym relation between x and y'* means that sentence s meaning expresses one hypernym relation between two terms x and y occurring in s.

Then, to accurately assess the quality of a set of patterns P, we propose to evaluate Matching Precision (M-Precision) and Matching Recall (M-Recall), more adapted than usual precision and recall, defined as:

$$M{-}Precision(P, C) = \frac{|TM|}{|TM| + |FM|} \tag{3}$$

$$M{-}Recall(P, C) = \frac{|TM|}{|TM| + |FNM|} \tag{4}$$

where **TM** (True Matched sentences), **FM** (False Matched sentences), and **FNM** (False Not Matched sentences) are computed as follows:

$$TM = \{s \mid s \in PS,\ \exists p \in P,\ p \textit{ matches } s \textit{ extracts } (x, y), (x, y) \in HH{-}pairs\} \cup \{s \mid s \in NS,\ \exists p \in P,\ p \textit{ matches } s \textit{ extracts } (x, y),\ (x, y) \textit{ is validated manually as } HH - pair,\ s \textit{ expresses hypernym relation between } x \textit{ and } y\} \quad (5)$$

$$FM = \{s \mid s \in C,\ \exists p \in P,\ p \textit{ matches } s\ s \notin TM\} \quad (6)$$

$$FNM = \{s \mid s \in PS,\ \forall p \in P, \neg p \textit{ matches } s\} \quad (7)$$

Where

- *p matches s* & *extracts* (x, y) means that the pattern p matches sentence s and, at the same time, extracts pair (x, y),
- *p matches s* is equivalent to $\exists (x, y)$ *p matches s* & *extracts* (x, y), and
- (x, y) *is validated manually as HH − pair* means that an human expert has validated pair (x, y) as one hypernym relation.

However, using M-Precision, M-Recall requires to know for every sentence in a corpus which hypernym relations are expressed in the sentence itself. This cannot be expected for any corpus; thus, we have designed a corpus labeling heuristics for approximating as better as possible the required information, as explained hereinafter.

Corpus labeling objective is to label each sentence in corpus C as positive or negative according to Definitions 1 and 2 above. Labeling is done once, in the beginning of workflow (Figure 1), and makes possible to compute TM, FM, and FNM when needed. For the purpose of labeling, we have defined some heuristics: indeed, for labeling a sentence as positive it is not enough to check if the sentence comprises at least one HH-pair. For instance, sentence 'Country music began a slow rise in American main pop charts' comprises HH-pair (country music, pop), but it does not express one hypernym relation between those terms. Therefore, after having analyzed several counterexamples in available corpora (see Section 4.1), we have defined 5 heuristics to select sentences comprising both HH-pairs and potentially expressing hypernymy relations between those pairs:

(i) at least one HH-pair (i.e. found in the given dataset) occurs in the sentence;
(ii) the hyponym and the hypernym occur in the sentence as noun phrases (NP). Generally, the relationships expressed in a sentence are between noun phrases (NP). For instance, in sentence 'Country music began a slow rise in American main pop charts', hypernym 'pop' occurs as a 'modifier' and not as a noun phrase;
(iii) the distance between the hyponym and the hypernym in a sentence is 10 words maximum (this is similar to what it is suggested in the reviewed approaches for deciding if a dependency path should be a feature). For instance, sentence 'Art rock aspires to elevate rock from teen entertainment to an artistic statement, opting for a more experimental and conceptual outlook on music' is likely to do not express hypernymy for pair (rock, music) because the distance between 'rock' and 'music' is more than 10 words;
(iv) hyponym and hypernym should not be directly related by 'and' or 'or' conjunction. For instance, sentence 'The contradictions may stem from different definitions of the terms ragtime and jazz' is likely to do not express hypernymy for pair (ragtime, jazz) and (jazz, ragtime) because terms are related by 'and';
(v) hyponym and hypernym do not occur in distinct brackets in the sentence. For instance, in sentence 'By the 7th century, the koto (a zither) and the biwa (a short-necked lute) had been introduced into Japan from China', terms belonging to HH-pair (zither, short-necked lute) occur in distinct brackets, so that the sentence is likely to do not express hypernymy between those terms.

Once labeling is performed, the set of positive sentences belonging to the labeled corpus is randomly partitioned: 60% of positive sentences constitutes the learning corpus while 40% of remaining positive

sentences constitutes the testing corpus. Learning corpus is then completed by adding a number of negative sentences equal to the number of positive sentences already contained in it. Testing corpus is also completed as done for the learning corpus. Evaluation step is also performed once for a given corpus, whenever all 3 phases have been completed. The objective is to evaluate the final achieved M-precision, M-recall, and f-score (i.e. the harmonic mean of M-precision and M-recall) for whole set of extracted patterns, possibly in combination with anti-patterns. These values are then compared to the metrics values calculated for the seed patterns (non-regression evaluation), original HPs, and ExtHPs; all metrics are evaluated by using the testing corpus as well.

3.4.2 Sequences preparation step

The main goal of this step is to prepare the relevant input data (called the sequence database) for the SPM algorithm and the relevant data for pattern selection, according to the phase objective.

First, sentences belonging to the learning corpus are matched to input patterns and classified, as needed by the phase objective, according to the TM, FM, and FNM definitions. Thus, each sentence falls in exactly one TM, FM, or FNM partition. It should be noted that TM, FM and FNM partitions are dependent on the specific phase because they are computed by using input patterns (and the learning labeled corpus). In phase 1, input patterns correspond to seed patterns; in phase 2 input patterns are the ones extracted in phase 1; and in phase 3, input patterns are the ones extracted in phase 1 and phase 2.

Second, and depending on the phase objective, classified sentences are further partitioned in a set of training sentences for mining patterns and a set of validation sentences for selecting extracted patterns according to targeted precision and recall. The general rule of 70%/30% applies. However, used sentences for computing training and validation sentences depend on the phase. For instance, in phase 1, training sentences are computed by randomly adding 70% of TM sentences and validation sentences are computed by adding both 30% of remaining TM sentences and an equal number of FM sentences. A detailed presentation can be found in Section 4.

All sentences are represented as sequences, thus a set of Training Sequences (TS) and a set of Validation Sequences (VS) are then constituted, both ones proper to the specific phase. The representation of sentences as sequences is described in Section 3.5.

3.4.3 SPM step

This step aims to mine frequent sequential patterns from the set of training sequences (TS) by applying one SPM algorithm. For experiences presented in Section 4, we use CloSPEC (BÉchet *et al*. 2015), an algorithm to mine frequent closed sequential patterns under multiple constraints. A sequence pattern is considered closed if no subsequence pattern (i.e. a more general pattern because comprising fewer elements) exists with the same support. Making possible to specify constraints is important to drive the mining algorithm to extract interesting sequential patterns and to keep under control the complexity, as better explained below. Constraints are considered as parameters, which values must be tuned for the purpose of each phase. Constraints are:

- **Support constraint:** refers to the minimum occurrence frequency to consider a sequence as frequent. In general, high support value is preferred to extract only highly frequent patterns. Unfortunately, interesting hypernym patterns may be infrequent in a corpus, thus using low support value is necessary. But using a low support value negatively impacts on the mining complexity and too many frequent sequential patterns are often extracted. For this reason, the other two constraints described below are very beneficial to reduce the number of extracted frequent sequential patterns and keep the mining complexity under control.
- **Gap constraint:** refers to the minimum and maximum number of sequence elements that can be ignored between two other sequence elements. In our approach, the minimum gap constraint is always fixed to zero because we noted that using a minimum gap greater than zero avoid extracting most of the hypernym patterns. Adopting a strictly positive maximum gap is interesting because it enables to ignore words (and dependencies) that are irrelevant for recognizing a hypernym relation. For instance, words ‘like jazz’ occurring in the following sentence between brackets are irrelevant to be part of the

hypernym pattern corresponding to 'x is a y': 'funk (like jazz) is a music that requires the musicians playing it . . .'. Limiting as much as possible the value of maximum gap is also beneficial to avoid extracting sequential patterns where their elements occur too far from each other in a sentence and likely do not express any hypernym relation.

- **Element number constraint:** refers to the minimum and maximum number of elements (itemsets) composing the sequential patterns. In general, hypernym patterns should comprise at least 2 elements (the hyponym and the hypernym); in our approach, a minimum number of itemsets fixed to 2 is always necessary to extract interesting patterns such as pattern 'x, y' (corresponding to an appositive phrase). Limiting as much as possible the maximum number of itemsets is beneficial to reduce the number of extracted irrelevant patterns while keeping under control the complexity of mining algorithm. This seems to be feasible because, for instance, we carefully analyzed ExtHPs (the 59 lexico-syntactic patterns used in (Seitner *et al*. 2016)) and found that the maximum length of patterns is 6. Thus, similar values can be used as maximum number of itemsets.

3.4.4 Sequential patterns selection step

Although the usage of constraints makes possible to limit the number of extracted patterns, plenty of them remain raw patterns and are not relevant. Earlier works (Nguyen *et al*. 2007; Bechet *et al*. 2012) address the selection of relevant patterns from raw patterns by asking experts to perform a manual selection. This is not very effective. We propose to automatically select relevant sequential patterns by using additional criteria. These criteria depend on validation sequences stated in step 'sequences preparation' and the objective to be achieved for the phase. Patterns selection is performed by following the substeps listed below:

(i) Selecting patterns (from the whole set of extracted raw patterns) comprising hypernym-hyponym paths. The existence of one dependency path between hypernym and hyponym in the pattern increases the guarantee that hypernym and hyponym are related in sentences, despite the *distance* occurring between the hypernym and hyponym (Aldine *et al*. 2018).

(ii) By using validation sequences, computing M-precision of each sequential pattern selected in the previous substep and the total M-recall (i.e. M-recall of the whole set of selected patterns). Thus, a pattern is kept if it shows M-precision greater (or lesser in the case of anti-patterns) than a given threshold while total M-recall remains greater than a threshold too. The 2 thresholds are parameters, which values are fixed according to the objective of the phase.

(iii) Removing any sequential pattern that is super-sequence of another pattern and showing the same M-precision. In other words, if two or more patterns have the same M-precision, only the one comprising fewer elements is kept (corresponding to the more general pattern).

3.5 Sequential representation of sentences

The sequence representation of a sentence is one important aspect of the proposed approach. Indeed, pattern representation may or may not enable to extract relevant relations. For instance, a simple lexico-syntactic pattern representation cannot allow representing the relationship between 'musical instrument' and 'guitar' in the sentence 'I like musical instruments invented in Spain, such as guitar' because of 'invented in Spain'. However, by using deep grammatical dependencies, it is possible to found that 'musical instruments' is related to 'guitar'. Nevertheless, as reported the related work section, grammatical dependencies can be used in several ways both for representing and discovering patterns, leading to better or worse precision and recall.

Before presenting in detail the proposed representation of patterns based on deep grammatical dependencies, we list and explain below the main differences and similarities between the proposed representation of sequences and the ones described in (Bechet *et al*. 2012; Nguyen *et al*. 2007) works:

- **Sequence coverage.** In Nguyen *et al*. (2007), the sequence only represents the shortest dependency path in a sentence connecting pairs of terms. In Bechet *et al*. (2012), all sentence words are represented

in the sequence. We suggest covering all sentence words as in Bechet *et al*. (2012) because we assume that the shortest dependency paths may not comprise all information that characterizes hypernym patterns.

- **Sequence elements.** In Nguyen *et al*. (2007), one sequence element is an item representing either a word or a dependency relation linking two words: if the word is a noun, 'NN' replaces the word itself. In Bechet *et al*. (2012), the sequence element is an itemset (a set of items) including the word itself, its lemma, and its POS tag. We also suggest that a sequence element is an itemset, precisely representing any dependency relation with one another element in the sequence (as better explained in the remainder).
- **Taking Noun Phrases into account.** Rather than referring to nouns (Nguyen *et al*. 2007; Bechet *et al*. 2012), we refer to Noun Phrases (NPs) for sentence representation as a sequence. Thus, the extracted sequential patterns enable to extract relations between NPs, which is more adapted because a sentence often conveys a hypernym relation between NPs, and not between single nouns.
- **Links between sentence elements.** In Nguyen *et al*. (2007), sentence elements are linked by the grammatical relation linking them together in the dependency tree. In Bechet *et al*. (2012), sentence elements are linked by their order of occurrence in the sentence. We suggest that sentence elements need to be linked by both their order of occurrence in the sentence and the grammatical relation linking them together in the dependency tree. The dependency relation of each element is part of the itemset (more details are provided in the remainder).

Proposed representation. Let $S =< e_1, e_2, \cdots, e_n >$ denote the sequence of elements representing a sentence. As also said above, an element of a sentence represents either a NP or any other word that does not belong to any NP. Let $I_{e_j} =< i_{(j,1)}, i_{(j,2)}, \cdots, i_{(j,m)} >$ a set of items (the itemset) representing all information about the element e_j.

For one element representing a NP, most information refers to the NP head word[2].The headword is the most important word in the NP, conveying the main meaning of the NP. A NP is composed of several words that usually are either nouns or modifiers, but only one of these nouns corresponds to the head word (root).

The nature of information associated to each element can be lexical, syntactic, and dependency. Information used in the proposed representation is:

- **Element label:** the label of the element. If the element is a NP, the label is the concatenation of the words (separated by '_') composing the NP.
- **NP:** in case of a NP element, 'NP' item is added.
- **Lemma:** the lemma (or the head word if the element represents a NP).
- **POS tag:** the POS tag (possibly the POS tag of the head word if the element represents a NP).
- **Relation and its direction:** the grammatical dependency in which the element is involved (possibly the grammatical dependency in which the head word is involved if the element represents a NP) concatenated with the dependency direction. A grammatical dependency is a binary relation between two words: the governor and the dependent (the element represents the dependent in the relation). The direction is right ($\rightarrow$) if the location of the governor word is after the dependent in the sentence; otherwise, it is left ($\leftarrow$).
- **Governor word:** the word (it can be a head word of a NP) to which the element is grammatically related.

For the purpose of learning sequential hypernym patterns, hyponym and hypernym should also be explicitly indicated in the sequence. Thus, two items, 'hypo' and 'hyper', are added. Additionally, for the purpose of representing the dependency path connecting the hyponym to the hypernym, the governor word item in a sequence element can be replaced by the generic 'hypo_gov' or 'hyper_gov'.

[2] For instance, in noun phrases 'musical instrument' and 'the title of book', 'instrument' and 'title' are the respective headwords.

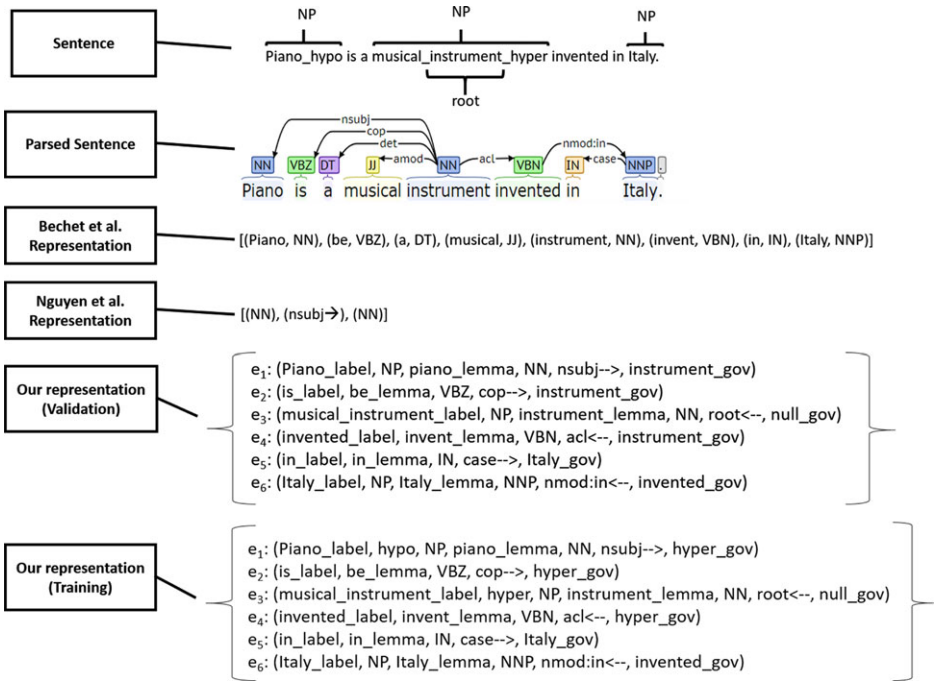

Figure 2. Examples of sequence representation of a sentence.

Let's consider the following sentence 'Piano is a musical instrument invented in Italy'. Figure 2 shows the representation of the sentence as a sequence according to the proposed representation and both (Nguyen *et al.* 2007; Bechet *et al.* 2012) representations ('_label', '_lemma', and '_gov' suffixes are used in the sequence representation for making easy to recognize items as corresponding to **Element label**, **Lemma**, and **Governor Word**). Looking to the shown representations of the sentence above, it is clear that the shortest path dependency is not enough for expressing all needed information (e.g. the tense is not necessarily part of the path linking hypernym and hyponym so that the relationship between terms is not explicit). It is also clear that a sequence based on POS-tags is not enough because hypernym and hyponym are not explicitly related by dependencies (and hypernym and hyponym may be unrelated in the sentence).

4 Experiments

This section presents the first evaluation, performed to validate the proposed approach by comparing the performances (M-precision and M-recall) of extracted patterns to the ones of three sets of patterns: DHPs, HPs, and ExtHPs. DHPs are used as seed patterns, thus the evaluation should make evident the nonregression of performances with respect to the seed patterns; ExtHPs and HPs are included for showing the capability of the proposed approach to extract patterns showing better performances than the ones of known manually designed patterns. The performed evaluation also shows the interest and need for the 3 phases by showing the contribution of each phase to the performance improvement. The section ends by an additional analysis for assessing to what extent the proposed approach tends to learn generic patterns.

4.1 Corpora and datasets

For conducting the experiments, we use three different corpora and two datasets. Two of three corpora are provided for the hypernym discovery task at SemEval2018 (Camacho-Collados *et al.* 2018): a domain-specific corpus (Music) and a general-purpose corpus (English-1). The third corpus is also a general-purpose corpus (English-2), used by Shwartz *et al.* (2017), which is a concatenation of the following two corpora: a corpus constructed by crawling the.uk domain (*ukWaC*), and a 2009 dump of the English

Table 1 Corpus labeling results expressed in number of sentences

Corpora	Initial Size	Learning Labeled Corpus Size		Testing Labeled Corpus Size	
		Positive	Negative	Positive	Negative
Music	∼3 Million	4052	4052	2702	2702
English-1	∼55 Million	2909	2909	1939	1939
English-2	∼100 Million	6828	6828	4552	4552

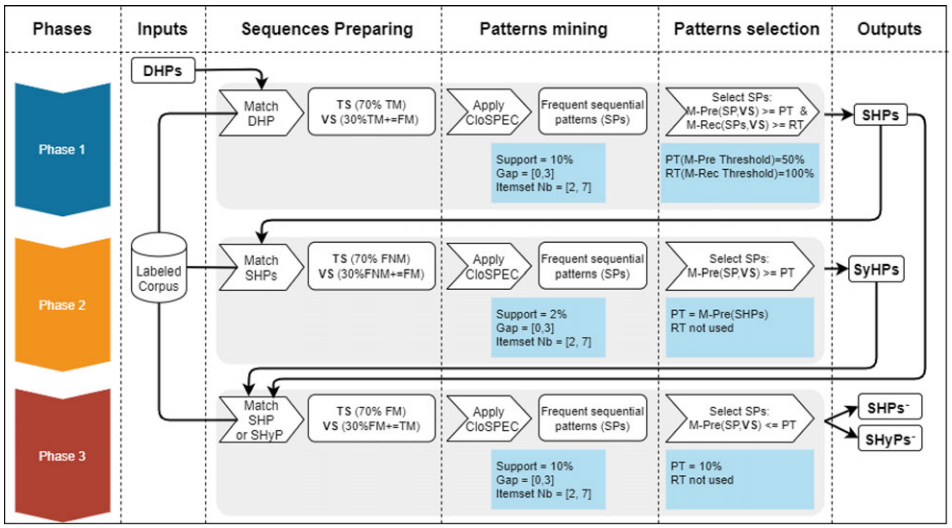

Figure 3. Sequential patterns learning using the 3-phase approach.

Wikipedia (*WaCkypedia EN*). The two datasets comprise the known HH-pairs, and are more or less specialized: Music is a dataset consisting of specific hypernym relations for the music domain (used with Music corpus), and English is a dataset consisting of general hypernym relations for the English language (used with both English-1 and English-2 corpora). Both datasets have been provided for the task of hypernym discovery at SemEval2018.

As described in Section 3.4.1, starting from a given corpus and a dataset of known HH-pairs, labeling is performed once for each corpus. For all used corpora, Table 1 shows the results of the corpus labeling step using the heuristics: numbers of sentences, corresponding numbers of positive and negative sentences, and numbers of learning and testing sentences.

4.2 Setting up workflow configuration

Each learning labeled corpus is used to learn sequential patterns by following the proposed 3-phase approach as shown in Figure 3. In previous work (Aldine *et al*. 2018), DHPs have been manually designed from HPs and it has been shown that DHPs lead to better performance than HPs and ExtHPs. For this reason, we use DHPs as seed patterns instead of HPs or ExtHPs. It should be noted that each manually designed DHP corresponds to one and only one HP: in the remainder, we therefore use the HP simple syntax to refer to the corresponding DHP because DHPs are syntactically complex and the interested reader can refer to Aldine *et al*. (2018).

Starting from DHPs, in phase 1, learning process steps (Figure 1) are repeated for each DHP, leading to one extracted set of sequential Hearst patterns for each DHP (i.e. a $SHPs_i$ for a DHP_i, $1 \leq i \leq 6$). SHPs $= \bigcup_{i=1}^{6} SHPs_i$ is then the union of all extracted sets of sequential patterns for the six DHPs. It should be

noted that with this strategy, the usage of CloSPEC can be parallelized, limiting the risk of memory fault due to the mining complexity.

In phase 2, learning process steps are performed once to extract new sequential patterns from FNM sentences (FNM sentences are found according to definition 7 taking into account patterns SHPs and the learning labeled corpus). SHyPs $= \{SHyPs_j, 1 \leq j \leq m\}$ denotes the set of extracted sequential patterns in this phase.

In phase 3, learning process steps are repeated for each $SHPs_i$ and for each $SHyPs_j$, leading to two sets of sequential anti-hypernym patterns $SHPs^-$ and $SHyPs^-$ respectively. $SHPs^- = \bigcup_{i=1}^{M} SHPs_i^-$ is the union of all extracted sets of sequential anti-hypernym patterns for each sequential pattern extracted in phase 1. $SHyPs^- = \bigcup_{i=1}^{K} SHyP_i^-$ is the union of all extracted sets of sequential anti-hypernym patterns for each sequential pattern extracted in phase 2. Even in this case, the strategy enables to parallelize the CloSPEC usage.

As described in Section 3.4, performing the generic workflow requires to set up some configurations. Specifically, the values of constraint parameters used for sequential patterns learning and thresholds for selection step should be set. The values of parameters and threshold shown in Figure 3 are the best-fit values after having performed a tuning process. Tuning values are given below. Note that some values have been fixed for all phases.

- **Support:** tuned using the following relatively low values {20%, 10%, 5%, 2%}. Indeed, some interesting patterns can be relatively infrequent.
- **Minimum gap:** fixed to 0. Positive values do not allow to find patterns.
- **Maximum gap:** tuned by using the following values {0, 3, 5, 10}. As already explained, we found that it is needed to set a maximum gap strictly positive (>0). We also found that using greater values than 3 has no advantage since no new patterns are discovered while, at the same time, falling in high mining complexity.
- **Minimum itemset number Nb:** fixed to 2.Patterns must have at least 2 elements i.e. one hypernym and one hyponym.
- **Maximum itemset number Nb:** tuned by using the following values {7, 8, 9, 10}. We found that using 7 is enough to learn hypernym patterns because with greater values extracted patterns are mostly wrong while, at the same time, falling in high mining complexity.
- **PT (M-Precision Threshold):** set according to the objective of each phase.
  - Phase 1 objective is to find sequential patterns (SHPs) more precise than DHPs. For this purpose, precision threshold is set to 0.5 corresponding to the lowest bound of M-precision of DHPs evaluated on VS1 (by definition VS1 comprises only 30% of TM and the equal number of randomly selected FM sentences; both TM and FM are computed with DHPs and the learning labeled corpus, written as VS1=30%TM+=FM). Additional details can be found in Section 4.2.1.
  - Phase 2 objective is to find patterns (SHPyPs) other than SHPs, thus improving recall, and showing the same or better precision than SHPs. For this purpose, precision threshold is set to the value of M-Precision of SHPs computed in the validation step.
  - Phase 3 objective is to find anti-patterns for patterns extracted in phases 1 and 2. For this purpose, precision threshold is set to a very low value (0.1) and patterns extracted should have even lower precision, because these patterns should be representative counterexamples of hypernym relations, as better explained in Section 4.4.
- **RT (M-Recall Threshold):** is only used in phase 1 where the learned patterns are constrained to do not reduce M-recall of DHPs. Thus, recall threshold is set to 1 because, by definition of VS1 (which does not comprise any FNM sentence with respect to DHPs), M-recall of DHPs on VS1 is necessarily equal to 1.

4.2.1 Phase 1: SHPs learning results

The sequence preparation step in phase 1 computes TM, FM and FNM by using DHPs and the learning labeled corpus (comprising both positive and negative sentences). For each DHP, a set of training

Table 2 Phase 1: Values for FCSPs, selected patterns for each selection criterion

DHPs		**DHP**$_{is\text{-}a}$	**DHP**$_{such\text{-}as}$	**DHP**$_{including}$	**DHP**$_{other}$
Music	FCSPs	73 161	208 038	32 293	74 620
	With Hyper-Hypo path	2024	5104	1329	1803
	M-Pre>= 0.5 & M-Rec= 1	15	16	16	13
	SHPs (only sub-sequnces)	11	14	14	13
	HiPre-SHPs (M-Pre>= 0.8)	2	2	13	7
English-1	FCSPs	51 870	37 251	Irrelevant TS (=11)	45 105
	With Hyper-Hypo path	1424	954		1281
	M-Pre>= 0.5 & M-Rec= 1	7	10		9
	SHPs (only sub-sequnces)	7	8		6
	HiPre-SHPs (M-Pre>= 0.8)	4	4		3
English-2	FCSPs	125 276	89 245	Irrelevant TS (=8)	24 392
	With Hyper-Hypo path	3720	2393		982
	M-Pre>= 0.5 & M-Rec= 1	9	5		7
	SHPs (only sub-sequnces)	7	4		7
	HiPre-SHPs (M-Pre>= 0.8)	4	2		4
Total distinct	SHPs	21	22	14	23
	HiPre-SHPs	10	8	13	12

sequences (TS) is constituted with 70% of TM sentences (written as TS=70%TM); a corresponding set of validation sequences (VS) is also constituted with the remaining 30% of TM sentences and the same number of FM sentences. VS1 and TS1 are the union of all computed VS and TS respectively (detailed results of this step are provided in Appendix A).

Table 2 shows the number of frequent closed raw sequential patterns extracted by applying CloSPEC on each TS (FCSPs) and, for each selection criterion, the number of selected patterns. The results show the high selectivity of the first criterion (i.e. existence of one hypernym-hyponym dependency path), specifically targeting precision. This selectivity is however partially balanced by keeping highest the resulting recall. We also report in the table the case of High Precision Sequential Hearst's patterns (HiPre-SHPs) selected by setting $PT = 0.8$ and without setting any RT. This specific experiment shows how to manage thresholds and what can be the impact—as a consequence, the number of patterns selected is dramatically reduced because PT is very high and recall may freely decrease.

Table 3 shows representative examples of SHPs learned from the Music corpus alongside additional details. For each SHP, the table points the corresponding DHP, one sample of TM sentence with the extracted HH-pair, one sample of FM sentence matching the corresponding DHP while not matching with the SHP, and the explanation/interpretation of the SHP. The M-precision evaluated on the relevant VS is also shown.

4.3 Phase 2: SHyPs learning results

The sequence preparation in phase 2 computes TM, FM and FNM partitions with SHPs, resulting from phase 1, and the learning labeled corpus. A set of training sequences (TS2) is constituted as 70% of FNM sentences (written as TS2=70%FNM); a set of validation sequences (VS2) is also constituted with 30% of the remaining FNM sentences and the same number of FM sentences (written as VS2=30%FNM+=FM). Detailed results of this step are shown in Appendix B.

Table 4 shows the size of TS2, the number of raw patterns extracted by CloSPEC on TS2 (FCSPs), and, for each selection criterion, the number of selected patterns; as for phase 1, the high selectivity of the first criterion should be noted.

Table 3 Phase 1: Representative samples of learned SHPs

DHPs	SHPs	M-Pre
DHP_{is-a}	[(NP, hypo, nsubj→ , hyper_gov), (be_lemma, cop→ , hyper_gov), (NP, hyper, band_lemma)] **TM sentence**: Strangers is a band founded in 2008 . . . ; HH-pair(Strangers, band) **FM sentence by DHP and not matched by the SHP**: Another notable percussionist was Parvinder Bharat (parv) of wolverhampton. **Explanation**: the lemma of the head of the hypernym NP is 'band'. In other words, this pattern extracts hyponyms for the specific hypernym 'band'. It increases the precision of DHP from 0.5 to 0.938. But, its recall will be almost null for a corpus not dealing with 'band'!	0.938
DHP_{is-a}	[(NP, hypo, nsubj→ , hyper_gov), (is_word, be_lemma, cop→ , hyper_gov), (NP, NN, hyper, root← , null_gov), (acl:relcl← , hyper_gov)] **TM sentence**: Jazz is a music genre that originated . . . ; HH-pair(Jazz, music genre) **FM sentence by DHP and not matched by the SHP**: The two founding members Jockel and Fritz are brothers. **Explanation**: the verb connecting the hyponym NP to hypernym NP is specified to 'is', the head of the hypernym NP should be a singular noun (NN) and the root of the dependency tree, and the hypernym NP should be followed by any word having 'acl:relcl← ' as grammatical relation.	0.784
DHP_{other}	[(NP, NN, hypo, dobj←), (and_word, and_lemma, CC, cc←), (amod→ , other_word, other_lemma, JJ, hyper_gov), (NP, hyper, conj:and← , hypo_gov)] **TM sentence**: it features piano and other keyboards . . . ; HH-pair(piano, keyboards) **FM sentence by DHP and not matched by the SHP**: you plug one end into your phone socket and the other end into your personal computer . . . **Explanation**: the head of the hyponym NP is specified to occur as object (dobj←).	0.656
$DHP_{such-as}$	[(NP, hyper, NNS), (case→ , such_lemma, JJ, hypo_gov), (as_lemma, IN, mwe← , such_gov), (NP, hypo, nmod:such_as← , hyper_gov)] **TM sentence**: Unlike genres such as jazz or opera, pop is . . . ; HH-pair(jazz, genres) and HH-pair(opera, genres) **FM sentence by DHP and not matched by the SHP**: . . . to lead to a general harmony such as bands made up of instrument known . . . **Explanation**: the head of the hypernym NP is specified to be plural noun (NNS).	0.538

Table 5 shows all the (five) new learned SHyPs from the three corpora and samples of TM sentences. To the best of our knowledge, 2 of them ($SHyP_{ranging}$ and $SHyP_{who}$, the first and third SHyP in the table) have not been identified in the literature. New sequential patterns refining the Hearst's pattern 'NP and/or other NP' have also been discovered. These new patterns do not have any counterpart in both DHPs and SHPs—indeed, dependency paths connecting hyponym to hypernym in the newly discovered patterns are

Table 4 Phase 2 : Values for TS2, FCSPs, and selected patterns for each selection criterion

Corpus	**Music**	**English-1**	**English-2**
TS	1129	800	1955
FCSPs	810 423	521 032	913 351
With Hyper-Hypo path	2362	799	1137
With M-Pre >= M-Pre of SHPs on VS1	15	17	8
SHyPs (only sub-sequences)	3	4	2

different from paths occurring in DHPs and SHPs. The limited number of new patterns can be explained by setting their M-precision to be equal or greater than M-precision of SHPs.

4.4 Phase 3: SHPs $^-$ learning results

The objective of phase 3 is to discover anti-patterns for patterns discovered in phases 1 and 2. Hereinafter, we first focus on anti-patterns for patterns discovered in phase 1 (i.e. SHPs). The sequence preparation step provides training and validation sequence sets per each set of SHPs patterns corresponding to one DHP (remeber that for one DHP several SHPs have been discovered in phase 1). Accordingly, for each set of SHPs patterns corresponding to one DHP, each TS is constituted with 70% of FM sentences (written as TS3=70%FM), being FM sentences computed with that set of SHPs and the training labeled corpus. The same is done for defining validation sets: each validation set VS is constituted with 30% of remaining FM sentences and the equal number of TM sentences (written as VS=30%FM+=TM), being TM sentences computed with that set of SHPs and the training labeled corpus. Discovering patterns showing a very low M-precision corresponds to patterns mostly matching with FM sentences (found in TS and VS). FM sentences are by definition sentences (positive or negative) matching with a pattern such that: a pair is extracted but it cannot be found in known HH-pairs or in WordNet, or human experts do not validate the sentences as expressing hypernymy stated by the pair. As a consequence, these sentences are likely to provide counterexamples of hypernymy.

Table 6 shows the number of extracted raw patterns from each TS (except the ones with few elements) and the number of selected patterns for each selection criterion.

Table 7 shows representative samples of learned SHPs$^-$. Each SHP$^-$ is provided with one sample of FM sentence alongside with the false HH-pair extracted from the sentence, and an explanation. We also show the M-precision of each SHP$^-$ calculated on VS. The very low precision of these patterns allows to filter the FM sentences by SHPs but also reduces slightly the recall by filtering some TM sentences matching SHPs. For instance, the second learned SHP$^-$ in Table 7 filters sentences like 'NP is in NP' (e.g. 'the vehicle is in motion') and 'NP is under NP' (e.g. 'the situation is under control'). However, the same pattern also filters sentences like 'NP is among NP' (e.g. 'the brain is among the organs that$\cdots$'), which both express hypernymy and pattern extracts a right pair of hypernym-hyponym.

4.5 Phase 3: SHyPs$^-$ learning results

In this section, we shortly present the anti-patterns SHyPs$^-$ for patterns discovered in phase 2 (SHyPs). In Appendix C, Table 17 shows values for TM, FM, TS, VS computed with each SHyP and on each training labeled corpus[3]. It should be noted that only $SHyP_{like}$, learned from Music corpus, has a relevant number of training sequences (54 >= 20). Therefore, it is the only pattern for which anti-patterns may be found. Starting from the associated training sequence set, 486 818 raw anti-patterns have been extracted and only patterns with M-precision lower than 0.1 are kept. Unfortunately, this criterion is highly selective and no pattern has been found (SHyPs$^-$ is void).

[3] Blank cells are those corresponding to patterns that are not learned by the corpus.

Table 5 Phase 2: All learned SHyPs from the three corpora

Corpus	SHyPs	M-Pre
Music	[(NP, hyper), (ranging_word, range_lemma, VBG, acl← , hyper_gov), (from_word, from_lemma, IN, case→ , hypo_gov), (NP, hypo, nmod:from← , range_gov)] **TM sentence**: . . . into various other genres ranging from punk rock to electronic . . . ; HH-pair(punk rock, genres) **FM sentence**: . . . many different styles at work here ranging from funk to metal . . .; due to dependency parsing error 'ranging' is related to 'work' and not 'styles' leading to identify wrong HH-pair(funk, work) **Explanation**: A hypernym noun phrase that is followed by 'ranging from' and a hyponym noun phrase, and at the same time, both hypernym and hyponym noun phrases are grammatically related to the word 'ranging'.	0.903
English-1	[(NP, hyper), (other_word, other_lemma, JJ), (than_word, than_lemma, IN, case→ , hypo_gov), (NP, hypo, nmod:than← , hyper_gov)] **TM sentence**: . . . better than any metal other than silver and copper . . .; HH-pair(silver, metal) and HH-pair(copper, metal) **Explanation**: A hypernym noun phrase that is followed by 'other than' and a hyponym noun phrase, and at the same time, hypernym and hyponym noun phrases are grammatically related by the grammatical relation 'nmod:than← '.	1.0
English-1	[(NP, hyper, NN), (who_word, who_lemma, WP, nsubj→ , hypo_gov), (NP, hypo, NN, acl:relcl← , hyper_gov)] **TM sentence**: . . . a person who has been a patient for more . . .; HH-pair(patient, person) **FM sentence**:a friend who is a teacher spoke to . . . **Explanation**: a hypernym noun phrase that is followed by 'who' and a hyponym noun phrase, and at the same time, hyponym and hypernym are grammatically related by the grammatical relation 'acl:relcl← '.	0.82
Music English-1 English-2	[(NP, hyper), (like_word, like_lemma, IN, case→ , hypo_gov), (NP, hypo, nmod:like← , hyper_gov)] **TM sentence**: What's great about living in a city like Paris is that there is literally never a dull moment; HH-pair(Paris, city) **FM sentence**: . . . he had friends like most boys . . . **Explanation**: a hypernym noun phrase that is followed by 'like' and a hyponym noun phrase, and at the same time, hyponym and hypernym are grammatically related by the grammatical relation 'nmod:like← '.	0.538
Music English-1 English-2	[(NP, hypo, NN, compound→ , hyper_gov), (CC, cc← , hypo_gov), (other_word, other_lemma, JJ), (NP, hyper)] **TM sentence**: You can check it on youtube or other websites like this; HH-pair(youtube, websites) **FM sentence**: Their style is a mixture of electro pop, punk, new wave, techno and other trends. **Explanation**: a hyponym noun phrase that is followed by conjunction word such as 'or' and 'and' and then followed by 'other' word and hypernym noun phrase, and at the same time, hyponym and hypernym are grammatically related by the grammatical relation 'compound→ '.	0.78

Table 6 Phase 3: Values for FCSPs, and selected patterns for each selection criterion

SHPs		**SHPs$_{is\text{-}a}$**	**SHPs$_{such\text{-}as}$**	**SHPs$_{including}$**	**SHPs$_{other}$**
Music	FCSPs	569 389	284 220	Irrelevant TS	Irrelevant TS
	With Hyper-Hypo path	5742	2325	(=14)	(=13)
	M-Pre<= 0.1	1	2		
	SHP$^-$ (only sub-sequences)	1	2		
English-1	FCSPs	506 451	132 571	150 258	Irrelevant TS
	With Hyper-Hypo path	5130	2471	3012	(=19)
	M-Pre<= 0.1	2	0	0	
	SHP$^-$ (only sub-sequences)	2	0	0	
English-2	FCSPs	1 424 351	201 621	123 771	176 271
	With Hyper-Hypo path	8742	1621	1224	921
	M-Pre<= 0.1	2	0	2	0
	SHP$^-$ (only sub-sequences)	2	0	2	0

Table 7 Phase 3: Representative samples of learned SHPs$^-$

SHPs	**SHPs$^-$**	**M-Pre**
SHPs$_{is-a}$	[(nmod:poss→ , hypo_gov), (NP, NN, hypo, nsubj→ , hyper_gov), (is_word, VBZ, be_lemma, cop→ , hyper_gov), (NP, hyper)] **FM sentence**: His instrument is the piano . . . ; False HH-pair(instrument, piano) **Explanation**: the verb connecting the hyponym NP to hypernym NP is specified to 'is', the lemma of the head of the hypernym NP is specified to be singular noun (NN), and the hypernym NP is preceded by a possessive pronoun (nmod:poss→).	0.08
SHPs$_{is-a}$	[(NP, hypo, nsubj→ , hyper_gov), (be_lemma, cop→ , hyper_gov), (case→), (NP, hyper)] **FM sentence**: the situation is under control . . . ; False HH-pair(situation, control) **Explanation**: Existence of a word between the verb (be_lemma) and the hypernym NP, playing a role in the grammatical relation 'case→ '.	0.09
SHPs$_{such-as}$	[(NP, hyper), (NP), (case→ , such_lemma, JJ, hypo_gov), (as_lemma, IN, mwe← , such_gov), (NP, hypo, nmod:such_as← , hyper_gov)] **FM sentence**: Many artists from different styles such as metal, pop . . . ; False HH-pair(metal, artists) and False HH-pair(pop, artist) **Explanation**: A NP occurs between the hypernym NP and the word 'such'.	0.05

4.6 Evaluation step: Quantitative results

Experiments presented in previous sections, are performed in the context the evaluation step (Figure 1) with the main objective to validate the 3-phase approach. For that purpose, precision and recall of available patterns HPs, ExtHPs, and DHPs, computed with each testing label corpus, are compared to precision and recall of discovered patterns, computed with the same corpus. Table 8 shows those M-precision, M-recall, and f-score for respectively Music, English-1, and English-2 testing corpora. The table also shows M-precision and M-recall stated on the testing corpora for the learned patterns resulting from each phase.

Table 8 Evaluation results on Music, English-1, and English-2 corpora

		Lexico-syntactic patterns		Dependency patterns	Sequential patterns			
Corpus	**Metrics**	**HPs**	**ExtHPs**	**DHPs**	**HiPre-SHPs**	**SHPs**	**SHPs SHPs$^-$**	**SHPs + SHPs$^-$ + SHyPs**
Music	M-Precision	0.556	0.576	0.464	**0.667**	0.486	0.518	0.516
	M-Recall	0.157	0.192	0.207	0.125	0.203	0.201	**0.24**
	F-score	0.245	0.288	0.286	0.210	0.287	0.289	**0.327**
English-1	M-Precision	0.451	0.474	0.428	**0.663**	0.482	0.493	0.499
	M-Recall	0.073	0.163	0.149	0.86	0.146	0.145	**0.177**
	F-score	0.125	0.242	0.221	0.152	0.224	0.224	**0.261**
English-2	M-Precision	0.41	0.41	0.371	**0.493**	0.392	0.424	0.427
	M-Recall	0.047	0.102	**0.123**	0.055	0.12	0.113	0.12
	F-score	0.084	0.163	0.185	0.099	0.184	0.179	**0.187**

This is indeed useful for understanding the contribution of each phase to the improvement of precision and recall.

The best f-score for the 3 corpora and for all patterns compared in Table 8, is achieved by the combination of all learned patterns and anti-patterns. On the one side, this confirms that selection criterion used for automatically selecting patterns are effective because performances are better than seed patterns. Indeed, for the 3 corpora, the table shows that, there is no regression of M-Precision and M-Recall (only for English 2, M-Recall is slightly under the M-Recall shown by DHPs) and detected improvements of M-Recall and M-Precision are significant. On the other side, the interest of the approach over the manual design of patterns is also confirmed. The approach can then be practically used, starting from other corpora for getting additional patterns in a systematic way. It can also be noted that if M-precision threshold used in phase 1 is set very high (HiPre-SHPs), M-recall dramatically decreases. This fact suggests that for discovering patterns, any M-precision threshold used in phase 1 should be set only slightly greater than 0.5.

Concerning the contribution of each phase, Table 8 shows that the best results are anyway got whenever the 3 phases are completed (last column of the table). Specifically, phase 2 is really needed to guarantee a much better recall and much better f-score. However, phase 3 only (column header SHPs+SHPs$^-$) is required to guarantee as much as possible a good M-Precision of discovered patterns (only slightly reduced by patterns discovered in phase 2) when compared to available patterns (HPs and ExtHPs). For better understanding of M-Precision/M-Recall variations, a qualitative analysis is presented in the next section.

4.7 Evaluation step: Qualitative analysis of learned patterns

In this section, a qualitative analysis for explaining quantitative results presented in the previous section, is developed. The first analysis is about DHPs (i.e. seed patterns) and learned patterns. We have inspected sentences both classified as FM with DHPs (i.e. sentences matching with any DHP but extracting a wrong HH-pair) and, at the same time, not matching with any SHPs. These are 75 sentences out of 653 of FM sentences with respect to DHPs. One example of these 75 sentences is: 'Some prominent Swedish bands spawned during this second wave, such as Marduk, Nifelheim and Dark Funeral'. This sentence does not match any SHPs because in all SHPs the headword of the hypernym has to be plural while 'wave' in 'second wave' is singular. However, constraining hypernym requiring headword to be plural in patterns type 'NP such as NP', is not necessarily interesting in all situations. For instance,

for sentence "So before a band such as Tower of Power would reject the style . . .', SHPs do not match (because 'band' is singular), but a HH-pair is rightly extracted by using DHPs. This situation provides a partial explanation for the slight reduction in recall when moving from DHPs to SHPs. Once again, this situation explains why completing all phases is needed for achieving better performances. The usage of SHPs$^-$ shows the important impact of these patterns on M-precision when moving from DHPs. These patterns have the greatest impact on FM sentences computed with SHPs. Out of 621 sentences found in FM with respect to SHPs, 75 sentences match to SHPs$^-$. An example of such sentences is: 'their music is rock, pop, and alternative' where the occurrence of possessive pronoun 'their' leads the sentence to match the first SHP$^-$ in Table 7. In this case, the anti-pattern is able to correct the absence of article 'a', usually part of 'is-a' patterns. However, anti-patterns cut some pairs extracted by SHPs, so that the recall naturally decreases. To understand the slight reduction in M-recall caused by SHPs$^-$, we have analyzed TM sentences computed with SHPs and also matching SHPs$^-$. We have found only 5 of such sentences out of 557 TM sentences, which explains the limited impact on M-recall. An example of such sentences is: 'Berry was among the first musicians to be inducted into the rock and roll hall of fame on its opening in 1986' where the occurrence of 'among' and the associated grammatical relation 'case→ ', leads the sentence to match the second SHP$^-$ in Table 7. Moving from DHPs and adding SHyPs, Table 8 shows a considerable M-recall improvement. To better understand this quantitative result, we have compared TM and FM sentences computed with SHyPs and TM and FM sentences computed with SHPs. TM computed with SHyPs comprises 111 additional sentences. An example of such additional sentences is: 'they fuse together sounds from different genres like metal and reggae' — HH-pair(metal, genres) and HH-pair(reggae, genres) are extracted. FM computed with SHyPs comprises 105 additional sentences. An example of such additional sentences is: 'Her voice is an instrument all it's own, slicing through the air like a weapon sometimes' — wrong HH-pair(weapon, air) is extracted. Therefore, because moving to SHyPs results in comparable TM and FM increments, M-precision remains stable. However, the additional 111 sentences in TM, are sentences removed from FNM computed with SHPs. Therefore, M-recall is dramatically improved. Phase 2 is therefore important to find several additional pairs even though several useful sentences remain unmatched.

4.8 Evaluation step: Analysis of learned patterns generality

As said in the Introduction, patterns are interesting because they can be interpreted and explained. For this raison, it can be possible to assess to what extent a pattern can be used across distinct domains. In this section, we are going to analyze 'pattern generality'. We define 'pattern generality' as the pattern ability to extract (right) HH-pairs across distinct corpora. Table 9 shows specific and common patterns (SHPs, SHyPs, and SHPs$^-$) learnt from each of the three corpora. For instance, 52 patterns have been learnt from the Music corpus and additional 5 have been learnt from both Music and English-1 corpora. The two columns headed 'Total common' and 'Total distinct' provide the total numbers of patterns found by using any of the 3 corpora and found by using only some of these corpora. It should be noted that, globally, very few patterns can be found by using all corpora and patterns are often specific to the corpus. Figure 4 shows for each corpus, the f-score across the three corpora got with the related learnt patterns. The f-score of DHPs is also shown. As it can be seen, the results confirm that all patterns learnt by using one corpus can be used for any other corpus without any significant negative impact on the f-score.

The results also highlight the advantage of the learnt sequential patterns from English-1 corpus over patterns learnt with the other two corpora. By conducting a deeper analysis, we have observed that there are 2 additional SHyPs patterns learnt from English-1 corpus compared to SHyPs patterns learnt form English-2 corpus. This fact explains why M-Recall (and f-score) is, for English-2 corpus, greater when using SHyPs patterns learnt with English-1 corpus. However, the situation is not the same for Music corpus. This is because English-1 and English 2 are general-purpose corpora while Music corpus is more specialized. So that, Music-corpus needs adapted patterns for raising M-recall (and f-score too).

Table 9 Replicability of the learned sequential patterns

		Music	English-1	English-2	Total common	Total distinct
SHPs	Music	52	5	3	3	80
	English-1	5	21	5		
	English-2	3	5	18		
SHyPs	Music	3	2	2	2	5
	English-1	2	4	2		
	English-2	2	2	2		
SHPs$^-$	Music	3	0	0	0	9
	English-1	0	2	0		
	English-2	0	0	4		

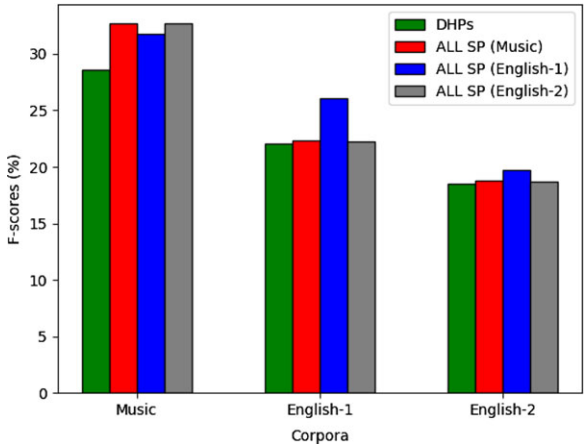

Figure 4. Performance of the learned sequential patterns on the three corpora.

5 Comparison to unsupervised and supervised hypernym detection baselines

Supervised and unsupervised distributional approaches target hypernym detection i.e. they are able, with specific mechanisms, to detect if a pair of terms given as input is or is not a hypenym relation. In this section, we compare the performance of learnt sequential patterns (**SHPs** + **SHPs**$^-$ + **SHyPs**) to the performance of various distributional baselines, specifically implemented for the purpose. The comparison is done with four datasets comprising known HH-pairs and the three corpora (Music, English-1, and English-2), being the latter already used for learning sequential patterns (see Table 1). Three datasets, BLESS, EVALution, and Weeds, are benchmark datasets convering general knowledge, commonly used to evaluate hypernymy detection approaches. The additional dataset, Music dataset, is a specialized dataset covering music domain, already used for learning patterns from Music corpus. However, Music dataset comprises only HH-pairs while here there is the need to provide pairs which do not represent hypernym relations. Thus, we add to Music dataset pairs that can be found in any other dataset (i.e. BLESS, EVALution, and Weeds) and not expressing hypernymy relations and, ii) both terms in such pairs can be found in the Music corpus. The size of each dataset is shown in Table 10.

5.1 Comparing patterns to unsupervised distributional approaches

Four measures typically used to evaluate unsupervised approaches are considered here: one similarity measure (**Cosine**), two inclusion measures (**ClarkeDE** and **invCL**), and one informativeness measure (**SLQS**). According to results presented in Shwartz *et al.* (2017), distributional semantic spaces built

Table 10 Datasets sizes

Datasets	**Music**	**BLESS**	**EVALution**	**Weeds**
Hypernym pairs	4478	1337	3415	1469
Non-hypernym pairs	14 849	25 217	10 260	1459
Total pairs	19 237	26 554	13 675	2928

Table 11 Average precision for comparing learnt patterns to both existing patterns and unsupervised methods on the base the four datasets and the three corpora

		Unsupervised Measures				**Other patterns**			**Learnt patterns SHPs+SHP$^-$**
Corpus	**Dataset**	**Cosine**	**ClarkeDE**	**invCL**	**SLQS**	**HPs**	**ExtHPs**	**DHPs**	**+SHyPs**
Music	Music	0.253	0.375	0.382	0.194	0.374	0.382	0.390	**0.42**
English-1	BLESS	0.093	0.119	0.120	0.047	0.477	0.481	0.493	**0.531**
	EVALution	0.279	0.348	0.353	0.220	0.321	0.322	0.326	**0.372**
	Weeds	0.532	0.685	**0.689**	0.392	0.556	0.556	0.557	0.602
English-2	BLESS	0.090	0.103	0.105	0.059	0.531	0.526	0.525	**0.57**
	EVALution	0.277	0.341	0.347	0.209	0.330	0.329	0.329	**0.42**
	Weeds	0.530	0.683	**0.688**	0.393	0.570	0.571	0.567	0.594

using dependency-based contexts are semantically richer than those using window-based contexts; moreover, one weighting feature over another one has no impact. Thus, we first build a distributional semantic space for each corpus using dependency-based contexts and occurrence frequency as feature weight. Then, we use the measures and distributional semantic spaces to compute a score for each pair in a dataset. Finally, pairs are ranked based on their score and the average precision is computed (Zhang & Zhang 2009).

For meaningful comparison with unsupervised distributional baselines, there is the need to compute the average precision for patterns too. For this purpose, patterns are first applied to extract HH-pairs from the three corpora. For each extracted pair (x, y), taking into account the extraction frequency of such a pair, a score is computed as follow:

$$score(x, y) = \frac{w(x, y)}{max(\{w(x_i, y_i)\}_{i=1}^{n})} \tag{8}$$

where $w(x, y)$ counts how often the pair is extracted by using all patterns (the frequency) and n is the number of distinct pairs extracted by the patterns. Then, the average precision is computed.

Table 11 shows the average precision for learnt patterns, HPs, DHPs, ExtHPs, computed with the four measures mentioned above. Globally, the results confirm that any set of patterns outperforms any unsupervised distributional baselines. However, the best results are achieved by the full set of learnt patterns (**SHPs** + **SHP**$^-$ + **SHyPs**). Nevertheless, unsupervised approaches based on inclusion measures (**ClarkeDE** and **invCL**) show better results on Weeds dataset. As stated by Shwartz *et al*. (2017), the reason is that Weeds dataset comprises numerous specific hypernyms for hyponyms, a situation specifically well handled by inclusion measures.

5.2 Comparing patterns to supervised distributional baselines

To compare patterns and supervised distributional approaches, patterns cannot be used as such because patterns cannot be directly used for hypernym detection. Therefore, we have specifically implemented one

Table 12 Performance of pattern based and embedding classifiers for hypernym detection with the four datasets and the three corpora

			Pattern-based		Embedding
Corpus	**Dataset**	**Metrics**	**Patterns as features**	**Dep paths as features**	**Concatenation**
Music	Music	Pre	**0.84**	0.77	0.82
		Rec	0.34	0.27	**0.36**
		F1	0.48	0.4	**0.50**
English-1	BLESS	Pre	0.81	0.7	**0.95**
		Rec	**0.54**	0.45	0.44
		F1	**0.65**	0.55	0.60
	EVALution	Pre	0.74	0.5	**0.89**
		Rec	0.3	0.32	**0.67**
		F1	0.43	0.39	**0.77**
	Weeds	Pre	**0.84**	0.72	0.76
		Rec	0.25	0.25	**0.79**
		F1	0.39	0.37	**0.77**
English-2	BLESS	Pre	0.79	0.73	**0.94**
		Rec	0.45	0.4	**0.50**
		F1	0.57	0.52	**0.65**
	EVALution	Pre	0.8	0.41	**0.90**
		Rec	0.32	0.52	**0.65**
		F1	0.46	0.46	**0.75**
	Weeds	Pre	**0.77**	0.71	0.71
		Rec	0.34	0.28	**0.88**
		F1	0.47	0.4	**0.78**

supervised baseline enabling to detect hypernyms, based on learnt patterns as features. Additionally, for better showing the interest of learnt patterns, we have also implemented an additional supervised baseline using the shortest dependency paths as features, as proposed in (Snow *et al*. 2005). All the classification results (precision, recall, and f-score) become therefore comparable. Details of the implemented baselines are given hereinafter.

Patterns as features. A similar method of Snow approach (Snow *et al*. 2005) has been implemented to train a hypernym classifier. But rather than using the shortest dependency paths as features, we use all patterns extracted from the 3 phases.

Dependency paths as features. We have re-implemented the approach described in (Snow *et al*. 2005) to train a classifier model. Accordingly, we first extract all shortest dependency paths connecting dataset noun pairs in a corpus; we consider only dependency paths that occur at least 5 times between noun pairs; we extend paths with satellite links to cover patterns like 'such NP as NP'. These paths are then used as features to train the classifier.

Embedding. We have implemented a distributional supervised baseline that relies on *word embedding* to represent the feature vector between a pair of nouns by *concatenating* their word embedding vector (Baroni *et al*. 2012). For this purpose, we train 3 word embedding models for each corpus using **Word2Vec**[4] with CBOW and dimension equal to 300.

All classifiers are then trained and evaluated by performing 10-fold cross-validation on each dataset using SVM with RBF kernel. Table 12 shows the average precision, recall, and f-score across all folds

[4] Word2Vec is available in Gensim python library.

Table 13 Percentages of HH-pairs detected both jointly and exclusively

Corpus	Dataset	Patters as features	Embedding	Both
Music	Music	17.1%	18.87%	17.0%
English-1	BLESS	24.7%	14.69%	29.2%
	EVALution	14.35%	51.82%	15.25%
	Weeds	7.02%	61.07%	17.95%
English-2	BLESS	24.19%	28.93%	20.7%
	EVALution	13.7%	46.46%	18.18%
	Weeds	7.2%	60.84%	26.81%

Table 14 Performance across distinct datasets of classifiers using patterns as features and embedding

	Dataset		Our method			Concatenation		
Corpus	Training	Testing	Pre	Rec	F1	Pre	Rec	F1
English-1	BLESS	EVALution	0.60	0.28	**0.38**	0.61	0.02	0.04
	EVALution	BLESS	0.89	0.22	**0.36**	0.31	0.24	0.27
Engish-2	BLESS	EVALution	0.46	0.27	**0.34**	0.75	0.01	0.02
	EVALution	BLESS	0.76	0.18	**0.30**	0.57	0.11	0.18

for the three approaches listed above, the three corpora and the four datasets. The results show that using patterns as features outperforms dependency paths as features. However, the best results are mostly achieved by the classifier based on word embedding even if precision is sometimes comparable.

As stated in the literature review, various works (Mirkin *et al*. 2006; Shwartz *et al*. 2016) have considered pattern-based and distributional approaches complementary and combined approaches have been proposed. For this purpose, we compute the percentage of hypernym pairs detected by relevant implemented baselines (features as patterns and feature as embedding vectors) and the percentage of pairs detected by exclusively one of these baselines (see Table 13). The results confirm this complementary: for instance, for English-1 corpus and BLESS dataset, 24.7% of the total hypernym pairs is detected by using patterns as features and not detected by using embedding vectors as features.

We also propose to evaluate the generality of using patterns or embedding vectors as features. For this purpose, we train two respective classifiers on the BLESS dataset, test them on the EVALution dataset, and vice versa using both English-1 and English-2 corpora. Table 14 shows the performance of each classifier. We highlight that the performance of the embedding based classifier is now much worse in comparison with its corresponding performance shown in Table 12; on the contrary, the performance of the classifier based on patterns as features also decreases but much less than the embedding based classifier. Hence, we can state that using patterns as features is expected to show a better general validity than pure embedding vectors, probably because patterns remain less dependent on the training dataset. Moreover, these results also confirm that the supervised distributional baselines are heavily affected by lexical memorization while this is not the case for patterns.

6 Conclusion and perspectives

In this paper, we have described a 3-phase approach that integrates the usage of sequential pattern mining and grammatical dependencies to systematically improve the performance of pattern-based hypernym extraction. The first phase aims to extract sequential patterns (SHPs) from a given set of seed patterns in order to improve precision of the seed patterns without degrading recall. The second phase aims to extend SHPs with distinct sequential hypernym patterns (SHyPs) for improving the recall while keeping

precision stable. The third phase aims to find anti-patterns to filter wrongly identified hypernym relations by both SHPs and SHyPs. Evaluation results show that the combination of the whole set of learnt sequential patterns from the three phases outperforms existing patterns such as lexico-syntactic Hearst's patterns, an extended set of lexico-syntactic patterns, and seed patterns (i.e. dependency Hearst's patterns). These results confirm the validity of the 3-phase approach. To complete the evaluation, we have shown that extracted patterns are loosely coupled with the corpora from which they have been extracted. This means that the 3-phase approach tends to learn generic patterns even if they have been extracted from domain-specific corpora. Last section presents an extensive comparison between learnt patterns and various representative distributional baselines (both unsupervised and supervised) for hypernymy detection. This comparison shows 2 key points. Learnt patterns, when made comparable with unsupervised distributional approaches, result in a much better performance (average precision, recall, and f-score). Learnt patterns, when made comparable with supervised distributional approaches, result in a close performance (f-score) even if recall remains worse. The comparaison is extended by focusing on two additional side aspects. First, patterns as features classifier outperforms dependency paths as features classifier, confirming that the usage of patterns is better than the usage of the shortest dependency path for hypernym detection. Second, patterns as feature classifier is less dependent on the training corpus than both dependency as feature and embedding classifiers. Thus, the 3-phase approach tends to learn generally valid patterns across distinct domains. Finally, we have analyzed the complementary between patterns and embedding. The results confirm that using both ones leads to a considerable increase in recall due to the ability of each of them to detect a large set of hypernymy pairs that cannot be detected by the other one.

Future works targets three objectives: applying the 3-phase approach on other sets of seed patterns (e.g. some ExtHPs that do not correspond to any DHP) for getting new patterns and continuing to increase both recall and precision associated to patterns; designing and implementing an approach for hypernym detection combining patterns and distributional methods because recall of pure patterns is likely to remain limited; applying the 3-phase approach to discover other types of relation (for instance, meronymy/part-of) because the 3-phase approach is generic and only seed patterns are impacted by the type of relation to be discovered.

References

Agrawal, R. & Srikant, R. 1995. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, *ICDE 1995*, IEEE Computer Society, 3–14, http://dl.acm.org/citation.cfm?id=645480.655281

Aldine, A. I. A., Harzallah, M., Giuseppe, B., BÉchet, N. & Faour, A. 2018. Redefining hearst patterns by using dependency relations. In *Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - Volume 2*: KEOD, INSTICC, SciTePress, 148–155, doi: 10.5220/0006962201480155

Baroni, M., Bernardi, R., Do, N. Q. & Chieh Shan, C. 2012. Entailment above the word level in distributional semantics. In *EACL,* 23–32.

Bechet, N., Cellier, P., Charnois, T. & Cremilleux, B. 2012. Sequential pattern mining to discover relations between genes and rare diseases. In *IEEE Int. Symp. on Computer-Based Medical Systems (CBMS),* 1–6.

BÉchet, N., Cellier, P., Charnois, T. & CrÉmilleux, B. 2015. Sequence mining under multiple constraints. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing, SAC 2015*, ACM, 908–914, doi: 10.1145/2695664.2695889, http://doi.acm.org/10.1145/2695664.2695889.

Buitelaar, P., Cimiano, P. & Magnini, B. 2005. Ontology learning from text: An overview. In *Ontology Learning from Text: Methods, Applications and Evaluation,* 3–12.

Camacho-Collados, J., Delli Bovi, C., Espinosa-Anke, L., Oramas, S., Pasini, T., Santus, E., Shwartz, V., Navigli, R. & Saggion, H. 2018. SemEval-2018 Task 9: Hypernym discovery. In *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018)*, Association for Computational Linguistics.

Cellier, P., Charnois, T. & Plantevit, M. 2010. Sequential patterns to discover and characterise biological relations. In *Computational Linguistics and Intelligent Text Processing*, Gelbukh, A. (ed). Springer Berlin Heidelberg, 537–548.

Chandramouli, K., Kliegr, T., Nemrava, J., Svatek, V. & Izquierdo, E. 2008. Query refinement and user relevance feedback for contextualized image retrieval. In *2008 5th International Conference on Visual Information Engineering (VIE 2008)*, 453–458.

Cui, H., Kan, M. Y. & Chua, T. S. 2007. Soft pattern matching models for definitional question answering. *ACM Transactions on Information Systems* **25**, 8.

Devlin, J., Chang, M. W., Lee, K. & Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Fellbaum, C. 1998. *Wordnet: An Electronic Lexical Database*. MIT Press.

Gomez-PÉrez, A. & Manzano-Mancho, D. 2004. An overview of methods and tools for ontology learning from texts. *The Knowledge Engineering Review* **19**(3), 187–212. doi: 10.1017/S0269888905000251.

Hearst, M. A. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics,* 539–545.

Hearst, M. A. 1998. Automated Discovery of Wordnet Relations. WordNet: An Electronic Lexical Database, 131–152.

Jacques, M. P. & Aussenac-Gilles, N. 2006. VariabilitÉ des performances des outils de tal et genre textuel. *cas des patrons lexico-syntaxiques* **47**, 11–32.

Klein, D. & Manning, C. D. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, *ACL 2003*, Association for Computational Linguistics, 423–430, doi: 10.3115/1075096.1075150, https://doi.org/10.3115/1075096.1075150.

Kotlerman, L., Dagan, I., Szpektor, I. & Zhitomirsky-Geffet, M. 2010. Directional distributional similarity for lexical inference. *NLE*, 359–389.

Levy, O., Remus, S., Biemann, C. & Dagan, I. 2015. Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, 970–976. doi: 10.3115/v1/N15-1098, https://www.aclweb.org/anthology/N15-1098.

Lin, D. 2003. Dependency-based evaluation of minipar. *Treebanks - Building and Using Parsed Corpora,* 317–329.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS,* 3111–3119.

Mirkin, S., Dagan, I. & Geffet, M. 2006. Integrating pattern-based and distributional similarity methods for lexical entailment acquisition. In *COLING and ACL,* 579–586.

Nguyen, D. P. T., Matsuo, Y. & Ishizuka, M. 2007. Exploiting syntactic and semantic information for relation extraction from wikipedia. In *IJCAI07-TextLinkWS.*

Orna-Montesinos, C. 2011. Words & Patterns: Lexico-Grammatical Patterns and Semantic Relations in Domain-Specific Discourses, 24.

Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U. & Hsu, M. C. 2001. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *International Conference on Data Engineering*, 215–224.

Pennington, J., Socher, R. & Manning, C. D. 2014. Glove: Global vectors for word representation. In *EMNL*, 1532–1543.

Ponzetto, S. P. & Strube, M. 2011. Taxonomy induction based on a collaboratively built knowledge repository. *Artificial Intelligence* **175**(9), 1737–1756, https://doi.org/10.1016/j.artint.2011.01.003, http://www.sciencedirect.com/science/article/pii/S000437021100004X

Roller, S., Kiela, D. & Nickel, M. 2018. Hearst patterns revisited: Automatic hypernym detection from large text corpora. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Association for Computational Linguistics, 358–363, http://aclweb.org/anthology/P18-2057.

Sang, E. T. K. & Hofmann, K. 2009. Lexical patterns or dependency patterns: Which is better for hypernym extraction? In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, *CoNLL 2009*, Association for Computational Linguistics, 174–182.

Seitner, J., Bizer, C., Eckert, K., Faralli, S., Meusel, R., Paulheim, H. & Ponzetto, S. P. 2016 A large database of hypernymy relations extracted from the web. In *LREC.*

Sheena, N., Jasmine, S. M. & Joseph, S. 2016. Automatic extraction of hypernym and meronym relations in english sentences using dependency parser. *In Procedia Computer Science,* 539–546.

Shwartz, V., Goldberg, Y. & Dagan, I. 2016. Improving hypernymy detection with an integrated path-based and distributional method. CoRR abs/1603.06076, http://arxiv.org/abs/1603.06076,

Shwartz, V., Santus, E. & Schlechtweg, D. 2017. 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*, Association for Computational Linguistics, 65–75, https://www.aclweb.org/anthology/E17-1007

Snow, R., Jurafsky, D. & Ng, A. 2005. *Learning Syntactic Patterns for Automatic Hypernym Discovery*. MIT Press, 1297–1304.

Srikant, R. & Agrawal, R. 1996. Mining sequential patterns: Generalizations and performance improvements. In *Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology, EDBT 1996*, Springer-Verlag, 3–17, http://dl.acm.org/citation.cfm?id=645337.650382

Wang, J. & Han, J. 2004. Bide: Efficient mining of frequent closed sequences. In *Proceedings of the 20th International Conference on Data Engineering, ICDE 2004*, IEEE Computer Society, 79, http://dl.acm.org/citation.cfm?id=977401.978142

Weeds, J. & Weir, D. 2003. A general framework for distributional similarity. In *EMLP,* 81–88.

Yan, X., Han, J. & Afshar, R. 2003. Clospan: Mining closed sequential patterns in large datasets. In: *SDM*, 166–177.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. & Le, Q. V. (2020) Xlnet: Generalized autoregressive pretraining for language understanding.

Yu, C., Han, J., Wang, P., Song, Y., Zhang, H., Ng, W. & Shi, S. (2020) When hearst is not enough: Improving hypernymy detection from corpus with distributional models.

Zhang, E. & Zhang, Y. 2009. *Average Precision*, Springer US, 192–193. doi: 10.1007/978-0-387-39940-9_482, https://doi.org/10.1007/978-0-387-39940-9_482

Zheng, W., Cheng, H., Yu, J. X., Zou, L. & Zhao, K. 2019. Interactive natural language question answering over knowledge graphs. *Information Sciences* **481**, 141–159, doi: https://doi.org/10.1016/j.ins.2018.12.032, https://www.sciencedirect.com/science/article/pii/S0020025518309848

Appendix A SHPs learning: Step 1 results

Table 15 shows for each corpus, and for each DHP, the size of TM, FM, TS, and VS. Some DHPs result in few TM sentences, leading to very limited number of TS sentences too. We have considered that whenever TS is less than 20 sentences, it is unreliable for mining (any TS with less than 20 sentences is shown with a gray background in the table)[5]. As a consequence, no mining has been performed on sentences belonging to any of the 3 corpora and exclusively matching with DHPs corresponding to HPs 'NP especially NP' and 'such NP as NP'. The same for sentences belonging to corpora English-1 and English-2 and exclusively matching with DHP corresponding to HP'NP including NP'.

Table 15 Phase 1: Values for TM, FM, TS, and VS for each DHP

DHPs	**DHP**$_{is\text{-}a}$		**DHP**$_{such\text{-}as}$		**DHP**$_{including}$		**DHP**$_{other}$		**DHP**$_{especially}$		**DHP**$_{as}$	
Music	TM	FM	TM	FM	TM	FM	TM	FM	TM	FM	TM	FM
	283	762	336	114	74	81	78	26	7	1	27	11
	TS	VS	TS	VS	TS	VS	TS	VS	TS	VS	TS	VS
	198	170	235	201	52	44	55	46	5	2	19	16
English-1	TM	FM	TM	FM	TM	FM	TM	FM	TM	FM	TM	FM
	239	328	58	36	15	33	138	35	3	0	4	2
	TS	VS	TS	VS	TS	VS	TS	VS	TS	VS	TS	VS
	167	143	40	36	11	8	97	70	2	0	3	2
English-2	TM	FM	TM	FM	TM	FM	TM	FM	TM	FM	TM	FM
	592	1008	93	81	12	83	60	72	4	5	5	10
	TS	VS	TS	VS	TS	VS	TS	VS	TS	VS	TS	VS
	414	355	65	56	8	8	42	36	3	2	4	2

Appendix B SHyPs learning: Step 1 Results

Table 16 shows the size of training and validation sequence sets for each set of SHP patterns corresponding to one DHP.

[5] Number 20 has been chosen by observing unsuccessful tests to extract good patterns from TS with less than 20 sentences; additionally, few sentences can be manually analyzed if needed.

Table 16 Phase 3: Values for TM and FM sentences, TS and VS for each set of SHPs corresponding to one DHP

Corpora	**SHPs$_{is\text{-}a}$**		**SHPs$_{such\text{-}as}$**		**SHPs$_{including}$**		SHPs$_{other}$	
Music	TM	FM	TM	FM	TM	FM	TM	FM
	214	530	225	60	50	20	62	18
	TS	VS	TS	VS	TS	VS	TS	VS
	371	318	42	36	14	12	13	10
English-1	TM	FM	TM	FM	TM	FM	TM	FM
	135	223	50	32	15	33	79	27
	TS	VS	TS	VS	TS	VS	TS	VS
	156	134	22	20	23	20	19	16
English-2	TM	FM	TM	FM	TM	FM	TM	FM
	426	703	71	49	12	83	39	35
	TS	VS	TS	VS	TS	VS	TS	VS
	492	422	34	30	58	24	25	20

Appendix C SHyPs^{-} learning: Step 1 results

Table 17 Phase 3: Values for TM, FM, TS, and VS for each SHyP

Corpora	**SHyP$_{ranging}$**		**SHyP$_{other\text{-}than}$**		**SHyP$_{who}$**		**SHyP$_{other}$**		**SHyP$_{like}$**	
Music	TM	FM					TM	FM	TM	FM
	20	7					32	14	152	78
	TS	VS					TS	VS	TS	VS
	4	6					9	10	54	48
English-1			TM	FM	TM	FM	TM	FM	TM	FM
			10	2	18	9	26	20	30	13
			TS	VS	TS	VS	TS	VS	TS	VS
			1	2	6	6	14	12	9	8
English-2							TM	FM	TM	FM
							13	14	23	25
							TS	VS	TS	VS
							9	10	17	16