# Redefining Hearst Patterns by using Dependency Relations

Ahmad Issa Alaa Aldine[1,3] Mounira Harzallah[2] Berio Giuseppe[1], Nicolas Béchet[1] and Ahmad Faour[3]
[1]*IRISA, University Bretagne Sud, France*
[2]*LINA, University of Nantes, France*
[3]*Lebanese University, Lebanon*

Keywords: Ontology Learning, Hypernym Extraction, Dependency Parser, Hearst Patterns.

Abstract: Hypernym relation extraction is considered the backbone of building ontologies. Hearst patterns are the most popular patterns used to extract hypernym relation. They include POS tags and lexical information, and they are applied on a shallow parsed corpora. In this paper, we propose a new formalization of Hearst patterns using dependency parser, called Dependency Hearst patterns. This formalization allows them to match better complex or ambiguous sentences. To evaluate our proposal, we have compared the performance of Dependency Hearst patterns to that of the lexico-syntactic Hearst patterns, applied on a music corpus. Dependency Hearst patterns yield a better result than lexico-syntactic patterns for extracting hypernym relations from the corpus.

## 1 INTRODUCTION

The huge availability of textual resources, especially on the Web, challenges the knowledge acquisition task. Indeed, there is the need to automate as much as possible knowledge extraction. Ontology learning techniques have been proposed to address this automation. The idea is to extract or facilitate the task of extracting structured knowledge such as concepts, relations (hypernym and ad-hoc relations), and axioms from texts. In our work, we focus on extracting hypernym relations from a text. Hypernym relations found in texts are indeed quite useful for suggesting taxonomies (or is-a relations) for building one ontology.

Broadly speaking, hypernym relation is a semantic relationship between two concepts: $C_1$ is a hypernym of $C_2$ means that $C_1$ categorizes $C_2$ (e.g. "instrument" is a hypernym of "Piano"). In this paper, we focus on *terminological hypernym relation*, i.e. hypernym relation as stated between two terms found in a text. In the last decades, extracting such hypernym relations gains a large interest because of their importance for understanding content as required in several applications such as question answering, machine translation, information retrieval, and so on.

In the past, methods to extract hypernym relations were based on lexico-syntactic patterns. Lexico-syntactic patterns rely on shallow linguistic techniques such as tokenization, lemmatization, and part-of-speech tagging to include information about the words of the sentence. The earliest and most popular lexico-syntactic patterns have been introduced by Hearst (1992) and they are known as Hearst patterns. For instance, the pattern: "NP such as (NP ,* or | and NP)" means that a noun phrase (NP) must be followed by term "such", term "as", and then by a NP or a list of NPs. Hearst patterns suffer from low recall because they are few in number. Moreover, they are prone to make errors due to their limitation in dealing with sentence ambiguity or complexity.

In this paper, we reformulate Hearst patterns as dependency patterns. Dependency patterns comprise information about the *dependency relations* between the words in a sentence. To appreciate the differences between lexico-syntactic patterns and dependency patterns, let consider the following sentence: "I like musical instruments invented in Spain, such as guitar". Table 1 represents the lexico-syntactic information extracted from the sentence by applying shallow linguistics techniques and the dependency information extracted from the sentence by using the enhanced representation of Stanford dependency Parser[1]. In this example, lexico-syntactic information is not sufficient to identify the correct hyper-

[1]https://nlp.stanford.edu/software/lex-parser.html



Aldine, A., Harzallah, M., Giuseppe, B., Béchet, N. and Faour, A.
Redefining Hearst Patterns by using Dependency Relations.
DOI: 10.5220/0006962201480155
In *Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2018) - Volume 2: KEOD*, pages 148-155
ISBN: 978-989-758-330-8

nym relation between the hypernym term "instrument" and the hyponym term "guitar"; worse, applying the pattern identifies a wrong hypernym relation between "Spain" and "guitar". On the contrary, dependency information seems to be sufficient to identify the correct hypernym relation between "instrument" and "guitar" by using the dependency relation "nmod:such_as(instrument, guitar)".

Table 1: Lexico-syntactic and dependency information representation.

| Lexico-syntactic | Dependency |
|---|---|
| I/PRP | nsubj(like, I) |
| like/VBP | root(ROOT, like) |
| musical/JJ | amod(instrument, musical) |
| instrument/NNS | dobj(like, instrument) |
| invented/VBN | acl(instrument, invented) |
| in/IN | case(Spain, in) |
| Spain/NNP | nmod:in(invented, Spain) |
| such/JJ | case(guitar, such) |
| as/IN | mwe(such, as) |
| guitar/NN | nmod:such_as(instrument, guitar) |

Thus, the contribution of this work is to reformulate Lexico-syntactic Hearst Patterns (LSHPs) using dependency relations, in order to improve precision and recall. We call the new definition of LSHPs, Dependency Hearst Patterns (DHPs). For evaluation purpose, we use a music corpus provided for the task of hypernym discovery in SemEval2018 (Camacho-Collados et al., 2018). The results of matching DHPs on the corpus confirm our assumption that patterns reformulated in term of dependency relations perform better.

The rest of the paper is organized as follows. Section 2 presents prominent existing methods for hypernym relation extraction based on patterns. Then, we introduce and describe DHPs in section 3. In section 4, we shortly present the corpus and describe how we clean it. In section 5, we evaluate precision and recall for DHPs and compare them to the ones found for LSHPs and ExtLSHPs (Extended set of lexico-syntactic Hearst patterns) . Finally, conclusions and perspectives of this work are presented in section 6.

## 2 RELATED WORKS

In the last decades, pattern-based approaches have been extensively used to extract hypernym relation between terms in a corpus. The patterns are either manually defined or automatically extracted. The most popular handcrafted patterns have been provided by Hearst (1992, 1998) to extract hypernym relations between noun phrases (NPs), and since then, known as Hearst patterns. Hearst patterns are categorized as lexico-syntactic patterns. They suffer from a low recall because they are few in number (see Table 2), while there are several syntactical ways to express the same relationship between terms in a sentence (Buitelaar et al., 2005). In addition, using these patterns on ambiguous and complex sentences dramatically reduces their precision.

Several approaches have been proposed to improve recall and precision of Hearst patterns. Most of them focus on extending Hearst patterns or using an extended set of Hearst patterns to increase the recall (Jacques and Aussenac-Gilles, 2006; Ponzetto and Strube, 2011; Orna-Montesinos, 2011; Klaussner and Zhekova, 2011; Seitner et al., 2016; Kamel et al., 2017). Jacques and Aussenac-Gilles (2006) propose new variant patterns for each Hearst pattern by replacing some words of the patterns by other words with similar meaning. For instance, replacing "such as" in the pattern "*NP* such as $NP_s$" by "like" to obtain a new pattern "*NP* like $NP_s$". Orna-Montesinos (2011) extends Hearst patterns by manual extraction of patterns that occur frequently between known hypernym relations. Seitner et al. (2016) use an extended set of Hearst patterns (59 patterns) collected from the past literature. Also, Kamel et al. (2017) use an extended set of Hearst patterns, starting from patterns defined by Jacques and Aussenac-Gilles (2006). Moreover, Ritter et al. (2009) proposed to improve the recall by applying inference rules to extract additional hypernym relations. On the other hand, Ponzetto and Strube (2011) propose to improve precision i) by identifying meronym (not hypernym) patterns, then ii) by removing hypernym pairs if they match meronym patterns more than hypernym patterns.

The approaches mentioned above are based on lexico-syntactic patterns. With the considerable performance achieved by dependency parsing of sentences, several approaches moved to the usage of dependency relations to improve the precision and recall of pattern-based approaches. Snow et al. (2005) proposed an approach to automatically predict hypernym relation between terms. They collect all noun pairs occurring in a corpus and label these pairs as hypernym related or not using WordNet (Fellbaum, 1998). Then, they use Minipar (Lin, 2003) dependency parser to extract *dependency paths* (an ordered set of relations without gaps) from sentences where the noun pairs occur. The occurrence frequencies of dependency paths fit as a feature vector to train a logistic regression classifier. Paths with higher weights in the classifier are considered as relevant paths to indicate hypernym relations. Using their method, they

were able to rediscover Hearst patterns and discover new patterns. In a later work, Sang and Hofmann (2009) apply the same approach of Snow et al. (2005) to compare between dependency paths and lexico-syntactic paths. They have concluded that the former performs better than the latter. More recently, Nakashole et al. (2012) have proposed a method to automatically extract hypernym relations based on dependency paths. They extract all shortest dependency paths between named entities, and then they generalize these paths by replacing words with syntactic, lexical, and semantic information.

In this paper, we also propose to get benefits of the considerable performance achieved by the dependency parser to increase the precision and recall of pattern-based approaches. Unlike previous approaches that use dependency paths to learn a supervised model (Snow et al., 2005; Sang and Hofmann, 2009) or to generalize these paths (Nakashole et al., 2012), we use an enhanced typed dependency relations to manually reformulate Hearst patterns as dependency patterns.

# 3 DEPENDENCY HEARST PATTERNS

## 3.1 Dependency Parsing

Dependency parser provides both lexical information in a sentence and its syntactic structure as direct binary grammatical relations that hold between words in the sentence (called typed dependency relation). Typed dependency relation is a grammatical binary relation between two words: head word and dependent word $Rel(Head, Dependent)$. A major advantage of dependency parser is the ability to deal with languages that have a relatively free word order. Another advantage of dependency parser is that their typed dependency relations associate words which are distant in a sentence; in this sense, dependency relations are closer to a sentence meaning (Marneffe et al., 2006).

More recently, Schuster and Manning (2016) present an enhanced English Universal Dependencies representation. They extend the representation by additional and augmented relations. An example of additional relations is the augmented modifiers, where all nominal modifiers in enhanced representation comprise the preposition e.g. nmod:such_as. Figure 1 and 2 show the enhanced typed dependency tree for sentences "I like musical instruments invented in Spain, such as guitar" and "A march, as a musical genre, is a piece of music with a strong regular rhythm" respectively. Below, some key typed dependencies required for reformulating Hearst patterns are explained:

- **nmod:such_as:** nmod refers to nominal modifier, associating a non-head noun that serves as a modifier of a head noun. "such_as" is the preposition name of "nmod". For instance, for a sentence reported above, the dependency parser provides the relation nmod:such_as(instrument, guitar).
- **cop:** cop refers to copula, i.e. a relation between a copula verb and its complement (all verbs "to be" are copula verbs). For instance, for a sentence reported above, the dependency parser provides the relation cop(is, piece).
- **nsubj:** nsubj refers to nominal subject, i.e. it represents the subject of a clause. The head in the relation is not always a verb, it can be an adjective or a noun when the verb is copula verb. For instance, for a sentence reported above, the dependency parser provides the relation nsubj(march, piece).

## 3.2 Patterns Definition

To reformulate Hearst patterns in term of enhanced dependency relations (resulting in what we name Dependency Hearst Patterns), we performed the following steps:

i. selecting for each Hearst pattern a random set of matching sentences from the music corpus (about 10 sentences for each pattern).
ii. extracting dependency relations between words of each sentence by applying the enhanced dependency parser.
iii. analyzing manually each set of parsed sentences.
iv. defining their corresponding general dependency patterns.

Table 2 shows the 6 Hearst patterns, and their corresponding dependency patterns (DHPs). Dependency pattern is an ordered set of dependency relations. A dependency pattern matches a sentence if each of its dependency relations matches in order with a dependency relation of the parsed sentence (if a pattern dependency relation of index $i$ matches a sentence dependency relation of index $j$, then the pattern dependency relation of index $i+1$ should match with a sentence dependency relation of index $j+k$, with $i, j, \& k > 0$).

To identify all hyponyms of one hypernym, as Hearst patterns, we use the conjunction dependency relation "conj($NP_y$, $NP_z$)". In other words, all NPs that

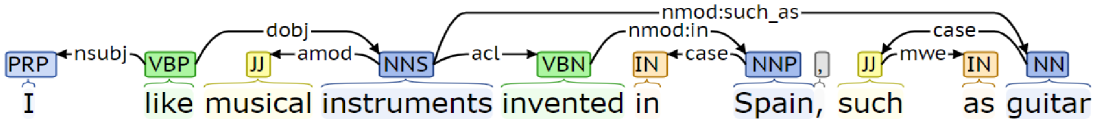


Figure 1: Enhanced typed dependency tree.

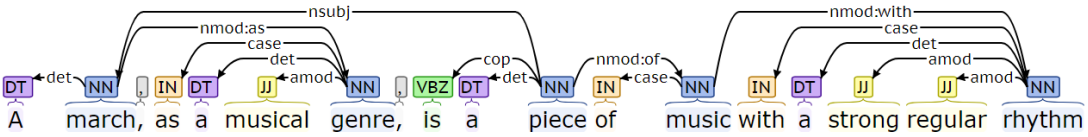


Figure 2: Enhanced typed dependency tree.

Table 2: Hearst patterns and their corresponding dependency patterns.

| Hearst Patterns | Dependency Patterns |
|---|---|
| NP$_x$ such as NP$_y$ | case($NP_yHead$, such)<br>nmod:such_as($NP_xHead$, $NP_yHead$) |
| Such NP$_x$ as NP$_y$ | amod($NP_xHead$, such)<br>case($NP_yHead$, as)<br>nmod: as($NP_xHead$, $NP_yHead$) |
| NP$_x$ including $NP_y$ | case($NP_yHead$, including)<br>nmod:including($NP_xHead$, $NP_yHead$) |
| NP$_y$ and\|or other $NP_x$ | cc($NP_yHead$, and\|or)<br>amod($NP_xHead$, other)<br>conj($NP_yHead$, $NP_xHead$) |
| NP$_x$ especially NP$_y$ | advmod($NP_xHead$, especially)<br>dep($NP_xHead$, $NP_yHead$) |
| NP$_y$ is a $NP_x$ | nsubj($NP_xHead$, $NP_yHead$)<br>cop($NP_xHead$, was\|were\|is\|are) |

are related by a conjunction with the hyponym NP ($NP_y$) are also hyponym NP ($NP_z$) of the same hypernym NP ($NP_x$). Let consider the following sentence: "I like musical instruments such as piano and guitar". Table 3 shows its enhanced dependency relations. A subset of these dependencies matches with the dependency pattern corresponding to "NP such as NP" to identify "musical instrument" as the hypernym NP of the hyponym NP "piano". Moreover, it identifies that "guitar" is also a hyponym of "musical instruments" using the conjunction relation "conj:and(piano, guitar)".

Table 3: The enhanced dependency relations.

| |
|---|
| nsubj(like, I), root(ROOT, like) |
| amod(instruments, musical), dobj(like, instruments) |
| case(piano, such), mwe(such, as) |
| nmod:such_as(instruments, piano), cc(piano, and) |
| conj:and(piano, guitar) |

## 3.3 Matching Noun Phrases in Dependency Patterns

In general, a sentence expresses semantic relations between NPs rather than between words. For instance, the hypernym relation between "instrument" and "guitar" or that between "piece" and "march" are less semantically rich (even not correct) than those respectively between "musical instrument" and "guitar" or "piece of music" and "march". The LSHPs are defined to identify hypernym relations between NPs. However, NPs are identified with few syntactic patterns based on a shallow parsing.

Although, dependency relations represent the syntactic relations between words, the defined DHPs allow identifying hypernym relations between NPs by using the notion of NPHead (Noun Phrase Head). When, DHP matches a sentence, it identifies the hypernym word ($NP_xHead$) and its hyponym words ($NP_xHead$). Then, hypernym relations are identified between the NP associated to the hypernym word and the NPs associated to the hyponym words. A NP is associated to a hyponym or hypernym word if the word and the noun phrase head (NPHead) are the same word.

A noun phrase head is a noun phrase word that complies the following criteria:

- it is a noun.
- it is not a modifier of another noun within the parsed sentence (nmod).

- it is not a compound of another noun within the parsed sentence (compound).

For instance, consider the noun phrase “a rock band”, “rock” and “band” are both nouns, but “rock”, according to dependency relations, is the “compound” of “band”, then we consider “band” as NPhead.

The extraction of NP from a sentence is done using the phrase structure tree of a dependency parser which has a good performance. Figure 3 shows the phrase structure tree of the sentence “I like musical instruments invented in Spain, such as guitar”. The noun phrases of the sentence are tagged by “NP”. In a nested noun phrase where a noun phrase comprises smallest noun phrases, we select the smallest noun phrases. For instance, “musical instrument invented in Spain” is a noun phrase that comprises the two smallest noun phrases: “musical instrument” and “Spain”; then, these two noun phrases are selected.

```
(ROOT
  (S
    (NP (PRP I))
    (VP (VBP like)
      (NP
        (NP (JJ musical) (NNS instruments))
        (VP (VBN invented)
          (PP (IN in)
            (NP (NNP Spain))))
        (, ,)
        (PP (JJ such) (IN as)
          (NP (NN guitar)))))
    (. .)))
```

Figure 3: Phrase structure tree.

## 4 CORPUS AND DATA

In our experiment, we rely on a music Corpus provided by the SemEval2018 organizers for the task of hypernym discovery (Camacho-Collados et al., 2018). The corpus is a concatenation of three music-specific corpora: ELMD corpus (Oramas et al., 2016), a corpus of Wikipedia music branch, and a corpus of album customer reviews from Amazon (Oramas et al., 2017). The task organizers also provide the corpus with training and testing data. Data corresponds to a list of hyponyms, and each hyponym is associated to a list of its hypernyms. A hyponym is either a concept or a named entity. In our work, we do not address “is-instance” relation extraction, thus we focus on hyponyms of concepts and ignore named entities.

We extract from both the training and testing data all hypernym relation couples between hyponym concepts and hypernym concepts. Then, we call a sentence a positive sentence if a hypernym couple occurs in the sentence and a sentence with no occurrence of any couple negative one. Indeed, if a pattern matches a positive sentence and extracts its hypernym couple, its recall and precision increase. If it doesn’t match a positive sentence, the recall and precision decrease (for more detail, see section 5.1). Our early analysis reveals that several positive sentences (i.e. containing a couple of Hyper/Hypo) do not convey the expected meaning. We name such sentences as ”non semantically positive sentences”. Consequently, we perform corpus cleaning to filter them. Thus, a sentence is considered as not semantically positive if:

- hyponym or hypernym term is not the head word of the noun phrase. For instance, let consider this sentence “Starting in the 1980, country music began a slow rise in American main pop charts”; the couple in this sentence is (country music, pop), and the hypernym “pop” isn’t the head word of its associated NP “American main pop charts”.
- hyponym and hypernym are related by a conjunction. For instance, let consider this sentence “The contradictions may stem from different definitions of the terms ragtime and jazz”; the couple (ragtime, jazz) is related by conjunction relation “and”.
- hyponym and hypernym occur within brackets and but not within the same bracket. For instance, let consider this sentence “By the 7th century , the koto ( a zither ) and the biwa ( a short-necked lute ) had been introduced into Japan from China”; each term of the couple (zither, lute) occurs within distinct bracket.

## 5 EVALUATION RESULTS AND ANALYSIS

### 5.1 Gold Standard Evaluation

In order to evaluate Dependency Hearst patterns (DHPs), we compare their precision and recall to the ones for lexico-syntactic Hearst patterns (LSHPs), and an extended set of lexico-syntactic Hearst patterns (ExtLSHPs). LSHPs and ExtLSHPs were implemented using Python[2] (Seitner et al., 2016). We also implement DHPs with Python.

First, we apply corpus cleaning to remove all non-semantically positive sentences. Second, we select

[2]LSHPs and ExtLSHPs exist in this link: https://github.com/mmichelsonIF/hearst_patterns_python

from the cleaned corpus 5000 balanced sentences, 2500 positive sentences (PS) and 2500 negative ones (NS). Third, we match DHPs, LSHPs, and ExtLSHPs with the 5000 selected sentences. The matching categorizes sentences according to the table 4 below, and these categories are useful for defining precision and recall. Fourth, we measure precision, recall, and F-measure of DHPs, LSHPs, and ExtLSHPs.

$$Precision = \frac{TM}{TM + FM}$$

$$Recall = \frac{TM}{TM + FNM}$$

$$F - measure = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Table 4: Sentence Matching Categories

| | Expected meaning YES | Expected meaning NO |
|---|---|---|
| **Semantically Positive** | TM | FNM |
| **Negative** | FM | TNM |

Table 5 shows precision, recall, and F-measure results for DHPs, LSHPs, and ExtLSHPs. DHPs show better results than LSHPs and ExtLSHPs by a slight improvement in term of precision and a considerable improvement in term of recall. The results confirm that dependency patterns perform better than lexico-syntactic patterns. And unexpectedly, the recall of DHPs (6 dependency patterns) is better than the recall of ExtLSHPs (59 lexico-syntactic patterns). This result is due to the high existence of sentences in the corpus that match DHP corresponds to the Hearst pattern "NP is a NP" and do not match the corresponding lexico-syntactic pattern. As clearly shown in the table 6, the recall of DHP corresponds to the pattern "NP is a NP", is approximately twice greater than that of LSHP.

Table 5: Precision, recall, and F-measure of DHPs, LSHPs, and ExtLSHPs.

| Gold Standard | Precision | Recall | F-measure |
|---|---|---|---|
| DHPs | **0.1521** | **0.0793** | **0.1042** |
| LSHPs | 0.1486 | 0.0455 | 0.0697 |
| ExtLSHPs | 0.1416 | 0.0647 | 0.0888 |

**Low Recall.** The results confirm that approaches relying on limited number of patterns suffer from low recall. The reason for this low recall is the existence of many positive sentences that do not match the patterns (FNM). For instance,"In popular belief, fado is a form of music characterized by mournful tunes and lyrics". Such sentences are frequent in the corpus, and they are useful to extract new patterns. For example, the latter sentence is useful to extract the pattern "NP is a form of NP".

**Low Precision.** The results also show a low precision for all set of patterns (DHPs, LSHPs, and ExtLSHPs). To understand the cause of this result, we analyze the negative sentences that match with the patterns (FM). We discover the occurrence of many hypernym couples in the sentences that are not covered by the gold standard provided by the SemEval2018 organizers. For instance, in this sentence "Often other small percussion instruments such as krap or chap are used", the hypernym couples (krap, percussion instrument) and (chap, percussion instrument) are true hypernym couples, but they are not covered by the gold standard.

Table 6 shows precision, recall, and F-measure results for each pattern belonging to DHPs and LSHPs. Some DHPs are better than corresponding LSHPs, while others are worse. This is due to the fact that DHPs are not completely better than LSHPs as clarified in the section 5.3.

## 5.2 Qualitative Analysis

In order to show the benefits of DHPs over LSHPs, we select and analyze some sentences truly matching DHPs, while not matching LSHPs or falsely matching them. In contrast to DHPs, we find that LSHPs do not perform well with some complex sentences, where some words occur between the noun phrase pair and not belonging to the pattern. For instance, in sentence "A march, as a musical genre, is a piece of music with a strong regular rhythm"; the phrase "as a musical genre" that occurs between "march" and "piece of music" prevents LSHPs to match the sentence, while DHPs correctly match it. Such sentences with a similar case are frequent, and especially that corresponds to the pattern "NP is a NP". Another example showing the efficient of DHPs over LSHPs is "I like musical instruments invented in Spain, such as guitar"; where a wrong hypernym couple is identified by LSHPs, while DHPs identify the correct couple thankful to the syntactic information obtained from the dependency parsing "nmod:such_as(instruments, guitar)".

## 5.3 Error Analysis

As mentioned above, DHPs are not completely better from LSHPs. DHPs are also prone to make errors when applied on complex sentences. Most of these errors are due to parsing errors. These parsing errors cause DHPs to either match positive sentences with wrong hypernym couple identification or never

Table 6: Precision, recall, and F-measure for each pattern of DHP and LSHP.

| Pattern | DHP | | | LSHP | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F-measure | Precision | Recall | F-measure |
| NP is a NP | 0.2941 | 0.042 | **0.0735** | 0.2618 | 0.02 | 0.0372 |
| NP such as NP | 0.2105 | 0.0128 | 0.0241 | 0.2705 | 0.0132 | **0.0252** |
| such NP as NP | 0.0 | 0.0 | 0 | 0.0 | 0.0 | 0 |
| NP and other NP | 0.3846 | 0.002 | **0.004** | 0.25 | 0.0008 | 0.0016 |
| NP including NP | 0.2667 | 0.0032 | 0.0063 | 0.3333 | 0.0052 | **0.0102** |
| NP especially NP | 0.0 | 0.0 | 0 | 0.2857 | 0.0008 | **0.0016** |

match the sentence. For instance, let consider the following complex sentence "Songwriters often play both piano, a key instrument for arranging and composing, and popular pop or rock instruments such as guitar". The parser wrongly outcomes the relation "nmod:such_as" between "pop" and "guitar", which cause DHPs to identify wrong hypernym couple (pop, guitar). Instead, LSHPs match this sentence and identify a true hypernym couple (Rock instrument, guitar).

### 5.4 Manual Evaluation

As mentioned in section 5.1, we notice the existence of several sentences labeled as false matched (FM) because the couple identified by the pattern is not covered by the gold standard; however, the couple is a true hypernym. Consequently, we randomly select 20% of sentences for each set of negative sentences matching with DHPs, LSHPs and ExtLSHPs respectively. Then, we manually label each sentence by true or false: true if the sentence conveys true hypernym relation, otherwise false. We find that 47.5% of negative sentences matching with DHPs can be reclassified as positive sentences if couples are added to the gold standard, 44.4% for LSHPs, and 40% for ExtLSHPs. The results confirm that DHPs, LSHPs, and ExtLSHPs are prone to make errors and identify wrong hypernym couples due to the complexity of the sentence or due to the fact that patterns are not always reliable to identify hypernym couples. Finally, we highlight a special case where the pattern "NP is a NP" identifies inversed hypernym couples, e.g. the couple (main instrument, side drum) in the sentence "The main instrument is the Side drum". Table 7 shows the results of precision, recall, and F-measure of DHPs, LSHPs, and ExtLSHPs using gold standard and manual evaluation. The results also confirm that dependency patterns perform better than lexico-syntactic patterns.

Table 7: Precision, recall, and F-measure of DHPs, LSHPs and ExtLSHPs with manual evaluation.

| Gold Standard + Manual Evaluation | Precision | Recall | F-measure |
|---|---|---|---|
| DHPs | **0.3327** | **0.1586** | **0.2148** |
| LSHPs | 0.3269 | 0.0949 | 0.1471 |
| ExtLSHPs | 0.295 | 0.1259 | 0.1765 |

## 6 CONCLUSION AND FUTURE WORK

In this paper, we have reformulated Hearst patterns by using dependency relations. We have evaluated resulting patterns (named Dependency Hearst patterns - DHPs) by using precision and recall on a cleaned music corpus. We have compared their precision and recall to those of lexico-syntactic Hearst patterns (LSHPs) and their extended set (ExtLSHPs), showing a better performance than both of them.

For defining DHPs, we select 10 random sentences matching Hearst patterns (section 3.2). This step can be iterated by using new sentences matching with the new pattern DHP until patterns do not change by the iteration.

Although DHPs show a better performance than LSHPs and ExtLSHPs, their recall is still low. Our analysis of the results puts in evidence several useful sentences for extracting patterns other than Hearst patterns. Our future work will be focused on the discovery of new patterns for hypernym extraction using a dependency representation of a sentence and sequential pattern mining (Agrawal and Srikant, 1995). Sequential pattern mining is efficient to extract the frequent sequential patterns as valid patterns for hypernym extraction.

## ACKNOWLEDGEMENTS

This research was partially supported by VOCAGEN project.

# REFERENCES


Agrawal, R. and Srikant, R. (1995). Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, ICDE '95, pages 3–14, Washington, DC, USA. IEEE Computer Society.

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

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

Fellbaum, C. (1998). Wordnet: An electronic lexical database. *Cambridge, MA: MIT Press*.

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

Hearst, M. A. (1998). Automated discovery of wordnet relations. *WordNet: an electronic lexical database*, pages 131–152.

Jacques, M.-P. and Aussenac-Gilles, N. (2006). Variabilit des performances des outils de tal et genre textuel. cas des patrons lexico-syntaxiques. 47:11–32.

Kamel, M., dos Santos, C. T., Ghamnia, A., Aussenac-Gilles, N., and Fabre, C. (2017). Extracting hypernym relations from wikipedia disambiguation pages : comparing symbolic and machine learning approaches. In *IWCS*.

Klaussner, C. and Zhekova, D. (2011). Pattern-based ontology construction from selected wikipedia pages. pages 103–108.

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

Marneffe, M.-C. D., MacCartney, B., and Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, pages 449–454.

Nakashole, N., Weikum, G., and Suchanek, F. (2012). Patty: A taxonomy of relational patterns with semantic types. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1135–1145, Stroudsburg, PA, USA. Association for Computational Linguistics.

Oramas, S., Espinosa-Anke, L., Sordo, M., Saggion, H., and Serra, X. (2016). Elmd: An automatically generated entity linking gold standard dataset in the music domain. In *Language Resources and Evaluation Conference (LREC 2016)*, pages 3312–3317, Portorož (Eslovenia).

Oramas, S., Nieto, O., Barbieri, F., and Serra, X. (2017). Multi-label music genre classification from audio, text, and images using deep features. *CoRR*, abs/1707.04916.

Orna-Montesinos, C. (2011). Words and patterns: Lexico-grammatical patterns and semantic relations in domain-specific discourses. 24.

Ponzetto, S. P. and Strube, M. (2011). Taxonomy induction based on a collaboratively built knowledge repository. *Artificial Intelligence*, 175(9):1737 – 1756.

Ritter, A., Soderland, S., and Etzioni, O. (2009). What is this, anyway: Automatic hypernym discovery. In *AAAI Spring Symposium - Technical Report*, pages 88–93.

Sang, E. T. K. and 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 '09, pages 174–182, Stroudsburg, PA, USA. Association for Computational Linguistics.

Schuster, S. and Manning, C. D. (2016). Enhanced english universal dependencies: An improved representation for natural language understanding tasks. In *LREC*.

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

Snow, R., Jurafsky, D., and Ng, A. (2005). Learning syntactic patterns for automatic hypernym discovery. *MIT Press*, pages 1297–1304.