ASPER: Attention-based Approach to Extract Syntactic Patterns denoting Semantic Relations in Sentential Context

Md. Ahsanul Kabir[a], Tyler Phillips[a], Xiao Luo[b], Mohammad Al Hasan[a]

[a] *Computer and Info Sci, Indianapolis, Indiana, USA*
[b] *Computer and Info Tech, Indianapolis, Indiana, USA*

Abstract

Semantic relationships, such as hyponym-hypernym, cause-effect, meronym-holonym etc., between a pair of entities in a sentence are usually reflected through syntactic patterns. Automatic extraction of such patterns benefits several downstream tasks, including, entity extraction, ontology building, and question answering. Unfortunately, automatic extraction of such patterns has not yet received much attention from NLP and information retrieval researchers. In this work, we propose an attention-based supervised deep learning model, ASPER, which extracts syntactic patterns between entities exhibiting a given semantic relation in the sentential context. We validate the performance of ASPER on three distinct semantic relations—hyponym-hypernym, cause-effect, and meronym-holonym on six datasets. Experimental results show that for all these semantic relations, ASPER can automatically identify a collection of syntactic patterns reflecting the existence of such a relation between a pair of entities in a sentence. In comparison to the existing methodologies of syntactic pattern extraction, ASPER's performance is substantially superior.

1. Introduction

In natural language text, often the entities in a sentence are related through various semantic relationships, such as, hyponym-hypernym, cause-effect, meronym-holonym, etc. For instance, in a sentence like `Sigmoid is a kind of activation function`, `sigmoid` and `activation function` share a hyponym-hypernym relationship. Similarly, in a sentence like `COVID-19`

`causes breathing difficulty in some patients`, there exists a cause-effect relation between `COVID-19` and `breathing difficulty`. Extracting such relationship between entities is an important task for natural language understanding. To extract semantic relationship between entities, human relies on some token-based template like Hearst pattern [1]. For instance, the template u `is a kind of` w denotes that there is a hyponym-hypernym relation between u, and w. Moreover, u `causes` w template suggests that u and w exhibit cause-effect relationship between themselves. Finally, there is a meronym-holonym relationship between u and w in the template u `comprise of` w. For a given semantic relation, there may exist many such templates in a language, but building a comprehensive list of templates for a relation is a challenging task. Besides, one will have to build a new list of templates for every new relation, so an approach for automatic extraction of such template patterns is of paramount importance.

Automatic extraction of template patterns is an important natural language processing task, as such patterns can be used to extract entity pairs exhibiting various semantic relationships [2, 3], a prerequisite for building a question-answering system [4, 5]. In the medical domain such patterns can help discover relations between disease, symptoms, and medication [2, 6]. Specifically, questions regarding the causes or the symptoms of a disease can be answered by extracting cause and effect terms from sentences in medical articles. Hyponym-hypernym patterns can also be used for ontology building [7, 8, 9, 10], and various methods are available for extracting hyponym-hypernym pairs from large corpora[11, 12, 13, 14, 15]. Although supervised learning methods can be used for some of the above NLP tasks, lack of labeled data always remains a challenge [16] for using a supervised learning method effectively. A major value proposition of template patterns is that such patterns can be used to create large (possibly noisy) labeled data, which can later be used for training of a supervised learning based model.

Developing an automated method for extracting patterns for an arbitrary semantic relation is a challenging task. While humans can easily recognize template patterns through a neuro-cognitive process that enables them to perceive a subject as a structured whole consisting of objects arranged in space or sequence, the same does not hold for a machine learning-based agent, which is better at statistical pattern recognition than template-based pattern recognition. Besides, an automated system lacks semantic understanding of the entities, so template patterns which only contain word tokens are not adequate for an automated system—a richer representation of each of the

tokens is needed for an ML-based method. So, it is no wonder that existing computational NLP and AI research have not ventured much into the automatic identification of token-based template patterns from natural language text. In this work, we enrich the tokens in a sentence using the dependency relations, and POS tags, and then apply a deep learning-based method to automatically identify a collection of template patterns of an arbitrary semantic relationship. Note that, when the tokens of a template pattern are enriched by using dependency relations and POS tags, we call them syntactic patterns; For instance, u `causes` w is a template pattern. To form a syntactic pattern, we add POS and dependency relation tags to each of the tokens. Excluding the keywords u, and w, we use lexicalization, and lemmatization of other tokens to avoid the form changes of words with respect to number, gender, tense, etc. Note that, for both the template patterns, u `causes` w and u `caused` w, u, and w are nouns, u is a `subject`, the lemmatized form of the principal verb is the `cause`, and the dependency relation of w with the `cause` is `dobj`, as w is a direct object of the verb. If we put the information together, the following, [*(u,noun,nsubj,cause),(cause,verb,dobj,w)*], is the corresponding generic syntactic pattern for both the template patterns.

In the existing literature, manual or semi-automatic approaches have been used for the extraction of template patterns. The earliest among these works was Hearst's seminal contribution [1, 17] on finding token-based template patterns for hyponym-hypernym relation through manual inspection. Similar manual approaches have also been used for extraction of token-based template patterns denoting cause-effect [16] and meronym-holonym [18] relations. But, the manual approach for pattern extraction is laborious and time-consuming. Besides, for every new semantic relationship, an independent inquiry needs to be pursued to obtain a collection of such patterns encoding that relationship.

Snow et al.[19] have proposed one of the earliest semi-automatic syntactic pattern extraction methods. However, the method is proposed considering only one kind of semantic relationship, hyponym-hypernym. From the methodological aspect, the proposed method uses a raw frequency threshold of sentential structures over the corpus for selecting a pattern, which generally produces patterns of poor quality. Subsequent to Snow et al.'s work, another semi-automatic work is proposed [20] for extracting meronym-holonym patterns. This method is also based on frequency threshold, and the authors themselves have reported that most of the extracted patterns are false positive. Though the extraction of syntactic patterns is not the focus of

most of the works, a number of works have devoted to utilize syntactic patterns for classifying whether a semantic relationship between a pair of entities exists or not [19, 21, 22, 23, 24]. Note that, extraction of syntactic patterns is orthogonal to the task of relation classification; the former extracts syntactic patterns from the sentences reflecting semantic relationship, whereas the latter classifies whether a semantic relationship between a pair of entities exists or not. In this paper, our focus is on the former task—extraction of syntactic patterns.

Machine learning based methods are also used for predicting semantic relations between a pair of entities in a sentence. Majority of these works [25, 26, 27, 28] consider the hyponym-hypernym relationship and solve a binary classification problem to identify whether such a relation holds between a given pair of entities. Such approaches are often designed to achieve high classification accuracy, but they are not capable of extracting syntactic patterns [29, 30, 31]. To summarize, automatic extraction of syntactic patterns for an arbitrary semantic relation is yet an unsolved task.

In this paper, we propose ASPER[1], a generic attention-based deep learning model that can identify syntactic patterns for any semantic relationship. ASPER follows a supervised learning approach—the model is trained through a collection of sentences; for each sentence, an ordered pair of entities are identified and a binary label is provided which denotes whether the entities are involved in a specific semantic relationship in that sentence. The output of the model is a collection of syntactic patterns which reflect the semantic relationship between entities involved in a chosen semantic relationship. By changing the training data ASPER can return syntactic patterns for any semantic relationship. To obtain the patterns of a given relationship, ASPER uses a bi-directional LSTM [32] with an attention layer [33], which highlights the part-of sentence (pattern) that are important to decide whether the identified pair of entities in the sentence are involved in that relationship. Importantly, in the data representation, ASPER does not use the embedding vectors of the entities whose relationship is inquired by the model, which compels ASPER to answer the query by discovering syntactic patterns capturing that relationship. Experiments on multiple datasets show ASPER's effectiveness.

[1]ASPER is composed of the bold letters in **A**ttention-based **S**yntactic **P**attern **E**xtraction for Semantic **R**elation

We claim the following contributions:

- We propose ASPER, a novel deep learning model which can extract syntactic patterns of a chosen semantic relationship between entities in a sentence, effectively and efficiently.
- Experiments on multiple semantic relationships, such as hyponym-hypernym, meronym-holonym, and cause-effect show that ASPER can identify most of the previously reported syntactic patterns of these relations. It can also identify a few patterns which have not been explicitly noted in earlier works.

2. Related Works

Related works are discussed in two groups. The first group comprises the works which do not extract syntactic patterns, rather perform semantic relationship classification. A subset of these works first manually collects patterns and then use them for pattern-based semantic relationship classification.

Semantic relationship classification approaches can be broadly categorized as distributional approaches, path-based approaches, and pattern-based approaches. Distributional approaches classify entity pairs based on the distinct contexts in which the two entities appear. Some of these works focus on building term embeddings for classification [34, 35, 36, 37, 38, 39, 40]. Path-based approaches instead consider contexts in which an entity-pair co-occurs. The lexico-syntactic paths which connect entity pairs in such contexts are leveraged in order to classify the pairs. We have already discussed one of the path-based approaches before [19]. An existing work [41] compared several path-based and distributional approaches and concluded that path-based approaches achieve better performance. Authors of [42, 43] proposed supervised approaches for relationship extraction in which they combine both path-based and distributional approaches in order to achieve state-of-the-art classification results. Some of the recent works among these use deep learning with attention for classification [44, 45, 46]. These works retrieve semantic embeddings and POS encodings for each term of a sentence in which an entity pair co-occurs. They also encode the proximity of each sentence term to the entities of interest. A supervised attention-based classifier is then trained to identify which terms within the sentences are important in determining if

the entity pair shares a certain type of semantic relationship. However, the attentions are not used to mine syntactic patterns, rather to validate whether the model is concentrating on the important segment of the sentence. The third group uses patterns for semantic relation classification [20, 1]. These methods can be benefited by the availability of an automated pattern extraction tool, like ASPER.

The second group of works either focus on manual or automatic pattern extraction. Within this group, some works extract patterns manually, some depend on taxonomies like WordNet, others are contingent upon sentential context. The first work which deals with hyponym-hypernym pattern extraction is carried out by Hearst [1] who manually extracts a few hyponym-hypernym syntactic patterns, now known as Hearst Patterns. A few research works later propose general pattern extraction methods which could be used for various semantic relationships. Hearst [17] designs a frequency-based approach using WordNet [47] entity pairs. Their approach simply scrapes sentences from a large corpus in which WordNet entity pairs of a certain relationship type co-occur. The syntactic patterns which are frequent across scraped sentences are then extracted. Motivated from the above works, some researches explicitly focus on automatic pattern extraction. For instance, authors of [19] first obtain dependency trees, and then apply a frequency threshold over the dependency tree edges to obtain hyponym-hypernym patterns. There exists another frequency-based unsupervised approach [48] for hyponym-hypernym relationship, which creates pattern clusters; the patterns are manually evaluated afterwards and the filtered list of patterns are then used for entity extraction. In another work on hypernym-hyponym template pattern extraction [49], authors follow an observation based semi-automated approach; however, the pattern extraction process is oblivious of syntax and dependency relation, which is important for obtaining high quality patterns. Finally, there are some research works, which focus on non-English hyponym-hypernym patterns [50, 12, 13, 51, 52].

Similarly, while meronym-holonym relation has been an important research topic in the literature due to various applications [53, 54, 55, 56, 57, 58, 59, 60], only a few research works have been performed in the area of meronym pattern extraction. Winston et. al. [61] manually developed a taxonomy of meronym patterns. For meronym-holonym pattern extraction, a Google search-based semi-automatic, frequency-based approach exists in the literature [20]. The authors report finding 1000 snippets, and 4503 unique patterns for 503 part-whole pairs. Top 300 frequent patterns

out of 4503 patterns are manually validated and they claim to get only 12 correct patterns. Generally, frequency-based, or bootstrapping methods generate a large number of noisy and false positive patterns [20, 62], which are later evaluated manually. There are some research works which emphasize on meronym-holonym patterns in other languages [63, 57]. Cause-effect relation has also received substantial attention in the existing literature [64, 65, 66, 67, 68, 69, 70, 71]. The supervised methods for cause-effect pairs extraction lack annotated dataset [64, 71] which is also true for other relations. However, for cause-effect relation, some works exist in the literature which depend on causative verbs, causal links, prepositions, and human extracted patterns [72, 22, 73]. Among them, the logical pattern-based semantic pair extraction method extracts causal patterns based on word dependencies in a given sentence over four sets of rules with define regular expressions [22]. The other method uses word vector-based similarity to find causative verbs; those verbs with some observed syntactic rules are then introduced as cause-effect patterns [72].

Our task of syntactic pattern extraction is related to other tasks, such as named entity recognition [74, 75], as input to both tasks are sequence data. For sequential inputs, besides LSTM, transformer architecture is also used, which is based on attention mechanisms, dispensing with recurrence and convolutions entirely [76]. The transformer can substitute recurrent neural networks, i.e, LSTM for text summarization [77, 78], machine translation [79, 80], etc. The main advantage of transformer over LSTM is that the former is order-independent as the attention mechanism of the transformer allows the model to work with any place of a sequence. Order-independence of the transformer also enables parallelism for processing the input sequence. For our task, the syntactic patterns that we extract are ordered, so a lighter model like LSTM suffices. Another appeal of the transformer is that it can handle longer sequence reducing the vanishing gradient problem of RNN. From our observation, syntactic patterns are shorter for which, LSTM works well. For sequential tasks, CRF [81] is also widely used, but for CRF the tokens are labeled; but in our dataset the labels are assigned to a sentence, not to its tokens, so CRF is a poor fit for this task. We are aware that some works extract cause-effect entity pairs using CRF [82]; but the same cannot be used for extracting syntactic patterns.

3. Methods

In this section, we begin by formally defining the relationship-based syntactic pattern extraction task. We then describe the LSTM architecture of ASPER along with its input representation and loss function. Finally, we describe how ASPER extracts syntactic patterns, and provide a pseudo-code of the end-to-end system.

3.1. Problem Formulation

Given a sentence S, and a pair of entities (words or phrases) u, w in S exhibiting a specific semantic relationship R (e.g. hypernymy, meronymy, causality, etc.), the task of syntactic pattern extraction is to extract a syntactic pattern, $\mathcal{P}$, which manifests that the entity pairs (u, w) are related through the relation R. To extract such patterns, in this work, we adopt a supervised learning model. As input, the model takes a set of triplets, $\mathcal{T} = \{(u_i, w_i), S_i, y_i\}_{i=1}^{\Lambda}$, where (u_i, w_i) is a directed pair of entities, S_i is a sentence in which words u_i and w_i co-occur, and y_i is a binary label indicating if the directed entity pair (u_i, w_i) exhibits the relationship R in the contextual scope of S_i; Λ is the number of distinct triples in $\mathcal{T}$. The objective of the model is to extract all syntactic patterns $\mathcal{P}$ such that, $\mathcal{P}$ is associated with one or multiple sentences in $\mathcal{T}$ indicating that the entity pairs (u_i, w_i) in those sentences are related through the relation R.

Recall that, a syntactic pattern is a sequence of tokens, along with POS tags, and dependency relations of the tokens. For example, the sentence `LSTM is a type of neural network` that exhibits hyponym-hypernym relation between `LSTM`, and `neural network`—the template pattern is `u is a type of w`, and the syntactic pattern is —
[*(u,noun,nsubj,be),(be,aux,attr,like),(type,noun,prep,of), (of,adp,pobj,w)*].
The purpose of ASPER is to extract such syntactic patterns from the input sentences.

3.2. Model Architecture

To successfully extract a syntactic pattern that demonstrates the relationship R in entity pair (u, w) in S, we must first determine whether u and w exhibit the relationship R. To make such distinctions, we train a binary classifier using a supervised approach through a set of training triples, $\mathcal{T} = \{(u, w), S, y\}$. Since our main objective is to extract syntactic patterns from sentences, a classification model that works with sequential data is needed. In addition, the model should be able to identify the parts of

the sentence which contribute the most to making the relationship prediction decision. For these reasons, we use a bi-directional Long Short-Term Memory (Bi-LSTM) [32] augmented with an attention layer [33, 83] as our binary classifier. The Bi-LSTM model is able to leverage the sequential nature of our sentence representation. Furthermore, as a result of supervised learning, the model's attention layer will be trained to highlight the parts of the sentence that are particularly useful in determining the presence of relationship R between the entity pair (u, w). We can, therefore, observe the attention layer to identify the important sentential constructs, which can then be composed to generate the syntactic pattern, $\mathcal{P}$. For a sentence, the Bi-LSTM model takes a vector-sequence representation of the sentence and outputs a prediction of the binary label. The complete model is shown in Fig. 1.

As shown in the bottom layer of Fig. 1, the input to the Bi-LSTM is the vector sequence representation of a sentence S. This representation denoted as, $\mathbf{X}$, has K edge embeddings in a sequence, each with dimension D, where the K edges are obtained from the dependency tree of the input sentence. The vector representation of a sentence, composed of a sequence of edge embeddings, is discussed in detail in Section 3.3.

The Bi-LSTM layer, $\mathcal{L}$, takes $x_i \in \mathbf{X}$ as input and outputs two hidden state vectors. The first hidden state vector, $\overrightarrow{h_i}$, is the forward state output, and the second hidden state vector, $\overleftarrow{h_i}$, is the backward state output. Let h_i be the concatenated output of $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$. Also, we define $\mathbf{H}$, which is the concatenation of each h_i output from $\mathcal{L}$ for a single sentence representation $\mathbf{X}$.

$$\overrightarrow{h}_i = \mathcal{L}(\overrightarrow{h}_{i-1}, x_i), \overleftarrow{h}_i = \mathcal{L}(\overleftarrow{h}_{i+1}, x_i)$$

$$h_i = [\overrightarrow{h}_i, \overleftarrow{h}_i], \mathbf{H} = [h_1, h_2....h_K]$$

Recall that, the shape of a single sentence representation, $\mathbf{X}$, is $K \times D$, where K is the number of edges in the sentence representation from the dependency tree and D is the dimension of each edge representation. Therefore, when given a single sentence representation, $\mathbf{X}$, the Bi-LSTM layer, $\mathcal{L}$, produces a concatenated output, $\mathbf{H}$, of shape $K \times 2 * N_u$, where N_u specifies the size of a single hidden vector. For our experiments, we set N_u to be equal to 256.

Following the Bi-LSTM layer, $\mathcal{L}$, output $\mathbf{H}$ is used as input to the at-

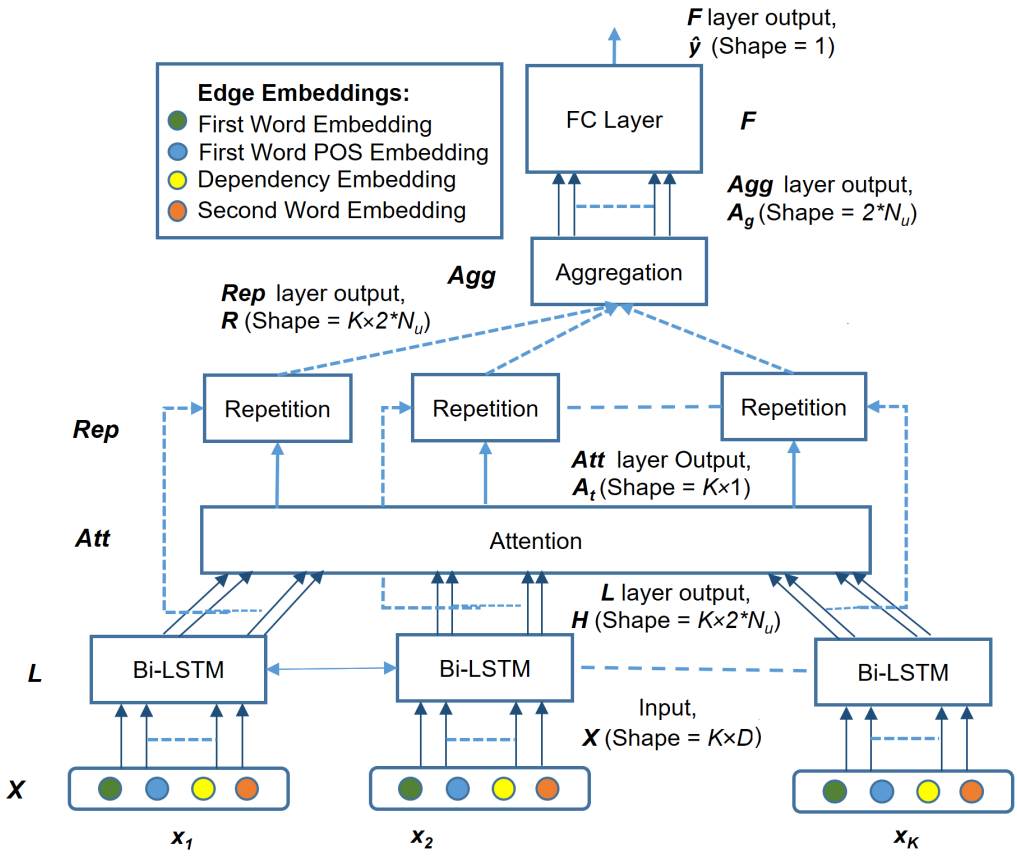


Figure 1: Description of LSTM with attention layer for binary semantic relationship classification.

tention layer, **Att**. The attention layer produces, $\mathbf{A}_t$, a vector of size $K \times 1$ where each $a_i \in \mathbf{A}_t$ is a value within a fixed range, $a_i \in [0, 1]$. Each such attention value, a_i, will encode the relative importance of edge embedding x_i in making the binary classification decision. $\mathbf{A}_t$ is computed as below.

$$Temp = \mathbf{Tanh}(\mathbf{H} * \mathbf{W_1}) * \mathbf{W_2}$$

$$\mathbf{A}_t = \mathbf{Softmax}(Temp)$$

$$R = \mathbf{A}_t * H$$

Here $\mathbf{W_1}$ is a trainable matrix of shape $2 * N_u \times 2 * N_u$, $\mathbf{W_2}$ is another

trainable matrix of shape $2 * N_u \times 1$. The shape of temporary variable *Temp* is K, on which we apply **Softmax** activation to retrieve $\mathbf{A}_t$.

Next, the model uses both $\mathbf{A}_t$ and $\mathbf{H}$ as inputs for the repetition layer, **Rep**. The repetition layer, **Rep**, outputs $\mathbf{R}$ of shape $K \times 2 * N_u$. $\mathbf{R}$ is simply the scalar multiplication of each hidden input $h_i \in \mathbf{H}$ with its corresponding scalar attention value, $a_i \in \mathbf{A}_t$.

Then, the model uses $\mathbf{R}$ as input for the aggregation layer, **Agg**. The aggregation layer simply computes the column-wise sum of $\mathbf{R}$ in order to yield the $2 * N_u$ shape output, $\mathbf{A}_g$. In short, $\mathbf{A}_g$ outputs the weighted sum of $\mathbf{H}$ where weights are the attention values.

$$\mathbf{A}_g = \mathbf{Summation}(\mathbf{R})$$

$\mathbf{A}_g$ is then used as input to a fully-connected layer with a sigmoid activation function, whose output is a scalar, $\hat{y}$, which denotes the prediction of a binary label, y, of a triplet, $t \in \mathcal{T}$.

$$\hat{y} = \mathbf{Sigmoid}(\mathbf{A_g} * \mathbf{W_3})$$

Here $\mathbf{W_3}$ is a randomly initialized weight matrix of shape $2 * N_u \times 1$.

Using these constructs, we train the binary classifier using the sentence embeddings generated from a collection of triplets, $\mathcal{T}$. We train the model using standard binary cross-entropy loss: $Loss = -\frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} y_t * log(\hat{y}_t) + (1 - y_t) * log(1 - \hat{y}_t)$ Using Early Stopping [84], we train the model until the validation loss does not decrease at the end of an epoch and then load the model parameters of the previous epoch with least validation loss.

3.3. Sentence Representation

To identify syntactic patterns from a sentence using machine learning, the sentence should be embedded in a form so that its syntactic structure is preserved. For this, we generate a dependency tree of a sentence [85] and use it as input to our learning model. The motivation is that the dependency tree of a sentence captures the syntactic structure of the sentence through a parse-tree like structure (see Fig. 2). However, the dependency tree only provides a symbolic representation, so we obtain a vector representation of it to be used as input to our model. Given an N-word sentence, S, we obtain a dependency tree of S by using a dependency parser.

In our implementation, we have employed two dependency parsers - Spacy and Stanza. We have utilized Spacy 2.2.3 [86]'s *en_core_web_sm*[2] package which is trained on a dataset of English sentences consisting of internet blogs, news, and comments. Additionally, the parser is ClearNLP parser, trained on OntoNotes corpus[3]. In contrast to Spacy, Stanza 1.2.2 [87] is trained on a total of 112 datasets of 66 human languages including the Universal Dependencies treebanks and other multilingual corpora.

While both the parsers produce different acyclic directed dependency tree, the tokens and their dependency relations may differ. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote any such dependency tree. Fig. 2 presents an example of such a dependency tree from Spacy. As shown in this figure, each vertex, $v_i \in \mathcal{V}$ is a tuple representing a word (or phrase) from S and the part-of-speech (POS) tag (e.g. noun, verb, adverb, adjective, etc.) of that word ($|\mathcal{V}| = N$). The edge-set, $\mathcal{E}$, is the set of all directed edges in the dependency tree with cardinality $|\mathcal{E}| = M$ ($M < N$). Each dependency edge e_{ij} links a parent vertex v_i to a child vertex v_j, and is labeled by the type of syntactic dependency (attribute, coordinating conjunction, compound, etc.) between the words at the two end-vertices of the edge. However, Stanza can also generate a similar type of dependency tree as shown in Fig 3.

In general, the dependency tree of a sentence, $\mathcal{G}$, may contain many vertices and edges which do not contribute to conveying if entity pair (u, w) share relationship R. For example, consider the sentence: `The cat, a type of animal, enjoys laying around and eating`. In this sentence, the first half of the sentence is critical in establishing that the `cat` is a type of `animal`. Clearly though, `enjoys laying around and eating` plays no role in establishing a semantic relationship between `cat` and `animal`. In our sentence representation, we discard such vertices and associated edges. Specifically, we preserve all vertices and edges which are along the shortest path connecting word pair u, and w. We also preserve descendants of u and w along with the edges which connect u and w to their descendants. Next, we organize the edges of the filtered tree into a fixed ordering, in which the edges in the shortest path between u and w come first, followed by the descendant edges of u and w.

In Fig. 2, the spacy dependency tree of the sentence, `Like most mammals,`

[2]https://spacy.io/models/en

[3]https://catalog.ldc.upenn.edu/LDC2013T19

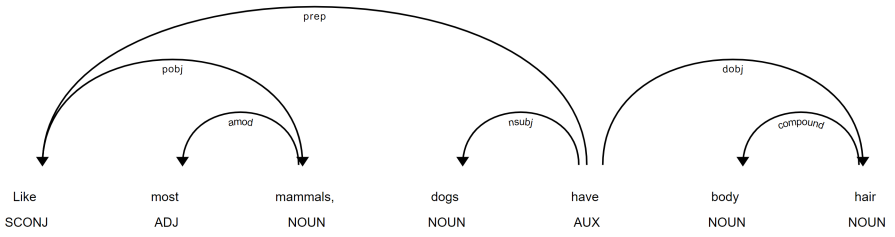


Figure 2: Spacy dependency tree with part-of-speech tags.

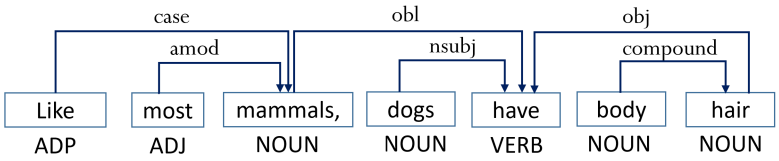


Figure 3: Stanza dependency tree with part-of-speech tags.

`dogs have body hair` is shown. Note that, the entities `mammals`, and `dogs` are provided for pattern extraction. All the edges along the shortest path from `mammals` to `dogs` are parts of ASPER's sentence representation. To make the patterns general, the entities `mammals`, and `dogs` are replaced with w, and u respectively. However, `most`, which is the descendant of `mammals`, is part of the desired syntactic pattern, `Like most w, u`. That is why the descendants of u and w are also important. But, neither all words on the shortest path, nor the descendants are part of the pattern. Similar argument goes for Stanza dependency tree in Fig. 3. That's why ASPER is an attention-based approach, so that the important words and edges for patterns can be extracted. For instance, for the above example, the token `most` needs to get attention by the attention mechanism. Likewise, ASPER's attention mechanism can deal with other determiners and modifiers as well in this way. Finally, once the edges are selected by attention mechanism, a frequent itemset mining algorithm is necessary to find the frequent edges which will be the desired syntactic pattern.

To generate the representation of a sentence, we embed each of the selected edges in sorted order and compose the resulting ordered edge representations, x_k, into a single vector sequence representation of $\mathcal{S}$, $\mathbf{X}$. The embedding of an edge $e_k = (v_i, v_j)$ (defined with variable x_k) is composed of

the following:

1. semantic embedding of the root word (or phrase) corresponding to the parent vertex v_i,
2. encoding of POS tag corresponding to root word (or phrase) corresponding to parent vertex v_i,
3. one-hot encoding of syntactic dependency between v_i and v_j, and
4. semantic embedding of the word (or phrase) corresponding to child vertex v_j.

Zero vectors of appropriate dimension are used for the semantic embedding of both the entities u and w. This forces ASPER to use only syntactical structural information entailing from sentence structure for predicting the relation between u and w, ignoring semantic information from these entity pairs. For the words, except u and w, we use 512-dimensional universal sentence encoder (USE) vectors [88]. Note that, one may use other choices, such as word2vec, or Glove, instead of USE. For the POS tags, we use two representations with Spacy both having 18 dimensions. First, which we refer as **One-Hot-POS Rep**, use one-hot encoding of the POS tags. Second, which we refer as **(Continuous-Prob-POS Rep)** concatenates the probabilities of all the POS tags for a token obtained from Spacy. For syntactic dependency types, we use one-hot-embedding which forms 58-dimensional vectors using Spacy. In contrast, using Stanza we have only one representation using one-hot encoding. Note that using Stanza we have 20 dimensions for POS tags and 60 dimensional one-hot vectors for dependency types. Therefore, any edge embedding has a fixed dimension D although D varies among representation methods. Finally, we fix vector sequence $\mathbf{X}$ to a fixed-length K (the number of edges) by either removing edge embeddings from the end of the sequence or adding zero-padding vectors of size D. This ensures that any sentence representation, $\mathbf{X}$, is of a fixed size, $K \times D$. Clearly, our sentence representation, $\mathbf{X}$, is agnostic to the relationship R, so it is capable of encoding an arbitrary semantic relationship between a given entity pair u and w.

3.4. Pattern Extraction Pipeline and Pseudo-code

After training the supervised learning model as discussed in Section 3.2, the model can be used for classifying whether an unseen pair of entities (within the context of a sentence) shares a relationship or not. This works

for an arbitrary semantic relationship as long as we can gather training data for that relationship. Since the entity pairs, u and w, are represented with zero vectors, model is oblivious of the semantic meaning of the entities; as a result the model is forced to predict the relationship by using the dependency edges and their syntactic augmentation. Our assumption is that important edges receive high attention value, and hence for each sentence (with positive label) we construct a candidate edge-list comprising of edges receiving high attention values. Now, to construct syntactic patterns, we apply frequent itemset mining (FIM) algorithm, ECLAT [89], over the candidate edge-set of the positively labeled sentences of $\mathcal{T}$, considering each edge as an item. FIM outputs a collection of edge-sets, such that each of the edge-sets exceeds the minimum frequency threshold (*supp*) over the sentences, i.e., they must appear in at least *supp*% of sentences. Frequent edge-sets, the output of FIM constitute the desired syntactic patterns, as each of these edge-set represents a syntactical unit of the sentence which receives high attention, and also appears in many positively labeled sentences (high support). The support (*supp*) value is a hyper-parameter of ECLAT algorithm, which we tune. The higher the *supp*, the more strict the quality control, resulting in a smaller number of false positive patterns. On the other hand, the lower the *supp*, the higher the chance that a pattern will be discovered.

ECLAT builds the frequent itemset iteratively. In the first iteration, it obtains itemsets of length one (consisting of a single edge) and filters any edge which does not have the desired minimum support. In the k'th iteration, it constructs candidate itemsets of size $k+1$, computes their support and filters the candidates which do not have the minimum support. The process completes when an iteration yields candidates such that none of them are frequent. The output edge-sets of ECLAT algorithm are numerous, because if a set of edges is extracted by ECLAT algorithm, all the subsets of that set will also be extracted by ECLAT as the subsets will also meet *supp* threshold. So, we keep only the maximal subset of edges. For all such edge-sets, we maintain the edge order and introduce that as a pattern.

The pseudo-code of ASPER is given in Algorithm 1. For a triplet $t(u, w, S)$ in a given collection $\mathcal{T}$, we first train the model and predict the label $\hat{y}$ and the attention values, $\mathbf{A}_t$, associated to the edges of t (Line 3-7). Next, we consider each $t \in \mathcal{T}$, for which the model predicts positively i.e., confirms the existence of relationship R in entity pair (u, w) in the sentence S (Line 8) . For the qualified triples, we normalized the attention values of their edges, and by using an importance threshold, *att* (a value between 0

Algorithm 1: Pattern Extraction

```
1  Extract_Pattern (𝒯, Model, att_thresh, supp)
2      edge_accum = []
3      for t ∈ 𝒯 do
4          {(u, w), S, y} = t
5          E = SentenceRepresentation(S, u, w)
6          X = Embed(E)
7          ŷ, A_t = Model(X)
8          if ŷ == 1 then
9              A_t = log(A_t)
10             edge_index_mask = A_t[A_t > att * max(A_t)]
11             edge_accum.append(E[edge_index_mask])
12         end
13     end
14     𝒫 = ECLAT(edge_accum, supp)
15     return 𝒫
16
```

and 1), filter out the edges of lesser relative importance (Line 11). As the attention values are on an exponential scale (output of a Softmax function), before applying the threshold, we take the logarithm of the attention values and then use min-max normalization to scale the attention values between 0 and 1 (Line 9-10). Corresponding to each triple, we accumulate an edge-set considering only the important edges (Line 11). Then frequent pattern mining algorithm is used to obtain a syntactic pattern-set (Line 14).

4. Experiments and Results

As ASPER is relation-agnostic, we validate its performance in extracting syntactic patterns for multiple relations; specifically, we choose hyponym-hypernym, cause-effect, and meronym-holonym relationships, as these three are well-studied semantic relations in the literature. We also compare the performance of ASPER with Snow's method [19], the only semi-automatic method (to be best of our knowledge) that extracts syntactic patterns. However, Snow's method works only for the hyponym-hypernym relation, so we compare with this method for results on this relation. For the other relations that we experimented with, we are not aware of a method, barring from

Table 1: Dataset Statistics

		# Pairs			# Sent		
Relation	**Dataset**	**Train**	**Val**	**Test**	**Train**	**Val**	**Test**
Hyponym-Hypernym	LEX	20,335	1,350	6,610	104,117	1,305	33,701
	RND	49,475	3,534	17,670	236,859	3,435	89,883
Cause-Effect	SemEval	6,914	1,053	2,838	7,157	1,166	2,861
	ADE	7,917	1,625	6,340	8,162	1,636	6,503
Meronym-Holonym	BLESS	11,151	3,225	10,163	23,412	3,889	19,512
	Phi	4,638	812	2,853	7,938	1,587	6,352
	SemEval	7,025	982	2,111	6,930	1,027	2,569

manual methods [16, 18], so for these relations, we show results on ASPER only.

4.1. Datasets

We use six datasets for validating the performance of ASPER. The statistics of the datasets are shown in Table 1, and some examples from each dataset are shown in Table 2. Among these, LEX and RND are used for hyponym-hypernym pattern extraction; SemEval and ADE datasets are used to perform cause-effect pattern extraction; and SemEval, Bless, and Phi's datasets are used for meronym-holonym pattern extraction tasks. Note that we use SemEval dataset for two relations.

Our problem formulation requires context sentences for the entity pairs, but four of the six datasets do not have any context sentence associated with the entity-pair. We obtain context sentences from Wikipedia. For this, we download the latest Wikipedia dump and extract all the sentences. Then, if a pair of entities co-occur in a sentence, we extract and associate that sentence with the entity pair. Note that, in this way, a given pair can be associated with multiple sentences.

It is important to understand that not every sentence has a pattern even if the sentence contains an entity-pair. On some occasions, sentences merely list a pair of entities, but do not imply a relationship between them in the sentential context. For instance, a row from the RND dataset is {*anthemis, genus, True*} which is shown in Table 2. We parse a sentence from Wikipedia for this row which is *Anthemis is a genus of aromatic flowering plants in the family Asteraceae*. Note that the sentence preserves a pattern between the corresponding entity pair. In contrast, let there be another instance of a

Table 2: Instances from Datasets

Dataset	u	w	Sentence	Label
LEX	**aarau**	**place**	**Aarau** *retain their* **place** *in the Swiss Super League.*	True
	cotmanhay	**england**	**Cotmanhay** *is a village in Derbyshire,* **England**.	False
RND	**anthemis**	**genus**	**Anthemis** *is a* **genus** *of aromatic flowering plants in the family Asteraceae*	True
	chuck biscuits	**black flag**	*It is the only official* **Black Flag** *release to feature* **Chuck Biscuits** *on drums*	False
SemEval	**musle fatigue**	**muscle pain**	**Musle fatigue** *is the number one cause of* **muscle pain.**	True
	castle	**museum**	*The* **castle** *was inside a* **museum**.	False
ADE	**clozapine**	**td**	*Several case reports have suggested that* **clozapine** *could also cause* **TD**	True
	castle	**museum**	*The* **castle** *was inside a* **museum**.	False
BLESS	**microphone**	**mics**	*The* **microphone** *is made of four* **mics**	True
	warrior	**face covering**	*The* **warrior** *wore a white cloak with a brown* **face covering**.	False
Phi	**committee**	**five members**	*The* **committee** *consists of* **five members**	True
	zula	**village**	**Zula** *is a* **village** *in central Eritrea.*	False

positive pair from the LEX dataset be {*aarau, place, True*}. One extracted sentence of this is *Aarau retain their place in the Swiss Super League.* Note that although the pair exhibits hyponym-hypernym relation, the sentence does not actually imply that. Yet in our dataset the sentence is a positive instance. This does not pose a significant problem because the frequent itemset method does not extract any pattern for this case as the candidate sequence of edges for this sentence is infrequent. For the true negative pairs the extracted sentences are unlikely to contain any pattern. More details of these datasets are provided below.

- **LEX & RND:** These datasets are obtained from [90]. They list a set of entity pairs with a label denoting whether the entity pair have a hyponym-hypernym relation (positive) or not (negative) without context sentences for an entity-pair. As discussed above, we use Wikipedia for obtaining context sentences for an entity-pair. Since multiple Wikipedia sentences can be associated with a given entity pair, for both the datasets, we allow at most five sentences to be associated with an entity pair. Both LEX and RND datasets are balanced having the same number of positive and negative sentences. Also, these datasets are already split into train, test, and validation partitions which we respected. In LEX dataset, disjoint entity pairs are used in train and test partition; while RND is split randomly, so the same entity pair may appear in training, validation, and test partitions, but with distinct sentences.

- **Bless:** We use this dataset for evaluating meronym-holonym pattern extraction. It was used in [91] for classifying different semantic relationships. It does not have any context sentence, so we extract sentences

from Wikipedia for these pairs. Since this dataset has entity pairs for many relations, we consider meronym-holonym entity pairs as positive class and others as negative class. For both positive and negative entity pairs, we allow at most 3 sentences for each pair. Finally, we maintain positive and negative sentence ratio as 1:1; split the dataset into training, test and validation maintaining 50%, 40%, 10%, respectively.

- **Phi:** This dataset is used in this paper [24], in which authors (Phi et al., whose name is used for naming this dataset) used word embedding for extracting different kinds of meronym-holonym relationships between entities. We use this dataset for evaluating meronym-holonym pattern extraction. This dataset contains only positive pairs with different kinds of part-whole relationships, such as component-of (11.2%), member-of (22.21%), stuff-of (18.89%), participates-in (15.23%), etc., as labels. For the negative pair sentence instances, we borrow from Bless dataset. We maintain a positive negative sentence ratio of 1:1 so that the dataset is balanced. Finally, we split the dataset randomly for training, test, and validation partitions maintaining 50%, 40%, and 10% instances respectively.

- **SemEval:** This is a well-used dataset, built by combining the SemEval 2007 Task 4 dataset [92] and the SemEval 2010 Task 8 datasets [93]. A row for SemEval datasets contains a term pair, and a sentence containing this pair which may exhibit a semantic relation. The SemEval 2007 Task 4 possesses 7 semantic relations whereas the SemEval 2010 Task 8 describes 9 relations. However, only two relations are common between these two, i.e., Cause-Effect and Meronym-Holonym relations. The datasets include predefined train and test partitions. For building validation partition, we borrow from the train partition. Train, test and validation partitions are then merged to concatenate into a single dataset. Note that, the merged dataset contains 14 relations. Now to create a dataset for Cause-Effect relation only, the sentences containing Cause-Effect relation are treated as positive sentences. The negative sentences are sampled randomly from other relations. Meronym-Holonym dataset is created in a similar manner; each row for both of these datasets contains a term pair, a sentence exhibiting the relation, and a binary label to denote whether the sentence preserves the relation. Moreover, for both the datasets, the ratio of positive and negative

sentences is 1:5, so the datasets are somewhat imbalanced, unlike other datasets. Finally, the percentage of the training, test, and validation sentences is 60%, 30%, and 10% respectively.

- **ADE:** The adverse drug effect (ADE) dataset [94] includes a set of predefined positive and negative examples. In the case of the positive examples, the cause-effect entity pair is given along with a corresponding sentence. On the other hand, the negative examples are only a collection of sentences that do not exhibit the cause-effect semantic relationship. In order to have entity pairs for each negative sentence, we randomly obtain two noun phrases from each negative sentence. This dataset is balanced in terms of the number of sentences and training, test, and validation partitions contain 50%, 40% and 10% data respectively.

4.2. Hyper-Parameters Discussion

For training using LSTM model we need to define a set of parameters. They are: (1) K (the maximum number of dependency tree edges in the sentence representation); (2) *batchSize* (total number of train instances in a batch); (3) the size of the hidden layer (N_u) in a Bi-LSTM unit; and (4) the learning rate. We fix the hidden layer size at 256, without tuning. We use Adam optimizer with its default learning rate (0.01), and early stopping. We tune K from the value between 30 and 40, and tune *batchsize* from the values $\{128, 256, 512\}$.

For LEX, RND, SemEval (Cause-Effect), ADE, BLESS and Phi datasets using `One-Hot-POS Rep`, we find the best results for $K = 30$, but for the ADE, and SemEval (Meronym-Holonym) $K = 35$. While using `Continuous-Prob-POS Rep`, the best results are found for $K = 30$ in LEX, RND, SemEval(both), ADE, BLESS datasets. For Phi dataset, $K = 35$ achieves the best result.Additionally, using Stanza for ADE, Phi datasets $K = 35$ achieves the best performance. But for all other datasets, using Stanza we find the best score for $K = 30$. For all datasets, *batchSize* equal to 128 produces the best result for both the representations with Spacy for all datasets except for the RND dataset, which requires *batchSize* equal to 256 to achieve the best result for `Continuous-Prob-POS Rep`. The reason that for the ADE, and SemEval datasets $K = 35$ provides the best result may be due to the fact that the dependency edges found for these datasets are comparatively larger. Note that K is contingent upon the maximum number of dependency

tree edges in the sentence representation. If K is kept large extra padding is added in the representation of sentence which may affect the performance of the model, otherwise pattern information will be missing because some edges will be skipped by a lower K value. However, for all the datasets using Stanza $batchSize = 256$ achieves the best score. The tuning of $batchSize$ in all our experiments seem to have little effect on performance change.

To extract patterns from the important dependency edges by using frequent itemset mining, we use *supp* (minimum support threshold in percentage) as a hyper-parameter. Another hyper-parameter is *att* (Attention threshold) which is used to filter the important edges. *att* is tuned for the values between 0.1 to 0.9 at 0.1 interval. We get good patterns for $att = 0.6$ for all the datasets using both representations of Spacy except Phi where $att = 0.1$ works well. For all the datasets, typically the most important edges are the edges which have at least 60% attention value of overall maximum value of that sentence. However, Phi dataset is an anomaly of this claim, as best *att* is 0.1 for this. One way to explain this can be – to find the patterns from Phi dataset we need to explore more edges which are comparatively less important. In contrast, using Stanza for RND, LEX, and SemEval(both) datasets $att = 0.5$ achieves the best patterns. For all other datasets except Phi $att = 0.6$. For Phi dataset using Stanza, we got $att = 0.2$ for the better patterns.

supp is tuned using a validation set from values between 0.1% to 3.0% at 0.1 interval; the patterns that we obtain from the validation set are manually scanned to choose the optimal values of *supp*. For a small value of *supp*, we find noisy and incomplete patterns, which do not qualify as syntactic patterns of a relation. Alternatively, if those values are too large, we find too few patterns. We find that a small support threshold works the best as they obtained larger patterns, denoting a full syntactic pattern, conveying a semantic relationship. For LEX and RND datasets, the optimum *supp* values are 0.28% and 1.3% using `One-Hot-POS Rep`; while the best *supp* values are 0.25% and 1.2% for `Continuous-Prob-POS Rep`. For the SemEval (Cause-Effect), ADE, and SemEval (Meronym-Holonym), the optimum *supp* values are 0.3%, 0.5%, 0.4% using `One-Hot-POS Rep`. For `Continuous-Prob-POS Rep`, the best *supp* values are 0.4%, 0.5%, 0.4% for the same datasets. Finally, for Bless and Phi datasets, the best *supp* values are 0.3%, 1.0%, respectively using `One-Hot-POS Rep` and 0.4%, 0.9% using `Continuous-Prob-POS Rep`. On the contrary, for SemEval (both), ADE, and Bless datasets using Stanza the optimum *supp* value is 0.4%. However, for Lex, RND and Phi datasets

the optimum *supp* values are 0.25%, 1.3%, and 0.9% respectively.

We perform an ablation study over *supp* (results shown in Section 4.7). One observation is that the optimum support values for both `One-Hot-POS Rep`, and `Continuous-Prob-POS Rep` are almost similar. The reason is that both methods are approximately similar except the difference of adding probabilities in stead of one hot encoding values. The optimum support values for all the methods are typically in the range 0.1% to 1.3%. The reason for this small support values is that patterns do not appear frequently among sentences. A large *supp* value will skip many edges which make the extracted patterns incomplete. Moreover, we need to provide a substantial number of edges to algorithm 1 to ensure extraction of complete patterns. Note that edges can be filtered and merged by algorithm 1 for complete and significant number of patterns. However, *supp* values less than 0.1% typically provide noisier edges to algorithm 1, which makes the algorithm extract edges which are not part of the patterns.Similar logic applies for the experiments with Stanza as well. However, we need to remember that the patterns found by Stanza are different than those with Spacy. This is due to the fact that the tokens, dependency relation differ between these two parsers.

4.3. Pattern Evaluation

Evaluating a pattern extraction is a difficult task as the ground truth for a pattern extraction method is not available. Existing works, manual or semi-automated only perform a qualitative evaluation. In this work, we have proposed two quantitative metrics for evaluating the performance of pattern extraction. For both the evaluation metrics, we use syntactic patterns to check whether two patterns match. We discuss the evaluation metrics below.

Evaluation on Sentence

Our first evaluation method builds ground truth by manually extracting patterns directly from the sentences in a dataset. Unfortunately, such an effort is time consuming and difficult for large datasets. So such an evaluation is only possible by sampling a subset of sentences in a dataset. So, given a potentially large test dataset, we first choose a random subset of sentences (around 1000) from the positive class (where the entity pair in the sentence exhibit the relation). For each of these sentences, we manually extract the pattern and make a ground truth pattern set over a sample of the dataset. If $\mathcal{P}_t$ is the total pattern set and $\mathcal{P}_o$ is the obtained pattern set over the

same sentences in the sample of the dataset, the following equations define precision and recall of pattern extraction by a method.

$$prec = \frac{|\mathcal{P}_o \cap \mathcal{P}_t|}{|\mathcal{P}_o|}, rec = \frac{|\mathcal{P}_o \cap \mathcal{P}_t|}{|\mathcal{P}_t|}$$

A problem with the previous evaluation metric is that it is computed over a random sample of sentences in the dataset, not the entire dataset. In fact, it is impractical to extract patterns manually over all the sentences in a dataset. But for any semantic relation, there generally exist a finite number of important frequent patterns, and it is easier to validate these patterns without observing them in the sentential context. In this evaluation method, we manually evaluate the precision of extracted patterns (over the entire dataset) by a method without evaluating them in the sentences. In other words, all the correctly predicted patterns in an extracted pattern set are considered to be the ground truth, and precision is computed as the ratio of correctly predicted patterns over all the extracted patterns. If we have more than one pattern extraction methods, we collect all the correctly predicted patterns by all of the methods and consider that to be the ground truth pattern-set and report precision on the basis of this set. Evaluation on a pattern is easier because the number of patterns is generally less than a hundred for a given semantic relation, and manual evaluation of a pattern is still possible without considering it in the sentential context. Let $\mathcal{P}'_t$ be set of collected patterns in the ground dataset and $\mathcal{P}'_o$ be the obtained pattern set by a specific method. Then, we define the precision and recall of the method with similar equations as before.

$$prec = \frac{|\mathcal{P}'_o \cap \mathcal{P}'_t|}{|\mathcal{P}'_o|}, rec = \frac{|\mathcal{P}'_o \cap \mathcal{P}'_t|}{|\mathcal{P}'_t|}$$

However, note that in this kind of evaluation, a method is not penalized for not discovering a pattern as long as no other competing methods is able to discover that pattern.

4.4. Quantitative Pattern Extraction Results

In this section, we first discuss the performance of ASPER using both representations with Spacy for its ability to extract patterns for three distinct relationships: Hypernym-Hyponym, Cause-Effect, and Meronym-Holonym and seven datasets; two for Cause-Effect, two for Hyponym-Hypernym, and

Table 3: Pattern Extraction Results of ASPER Evaluated on Sentence: One-Hot-POS Rep (Labeled as Rep 1 on Left), Continuous-Prob-POS Rep (Labeled as Rep 2 on Right)

		Rep 1					Rep 2		
Relation	**Dataset**	**Prec**	**Rec**	**F_1**	**Relation**	**Dataset**	**Prec**	**Rec**	**F_1**
Hyponym-Hypernym	**Lex**	0.78	0.7	0.74	**Hyponym-Hypernym**	**Lex**	0.74	0.67	0.7
	RND	0.88	0.72	0.80		**RND**	0.84	0.71	0.77
Meronym-Holonym	**Bless**	0.52	0.58	0.54	**Meronym-Holonym**	**Bless**	0.49	0.53	0.51
	Phi	0.62	0.67	0.64		**Phi**	0.62	0.66	0.64
	Semeval	0.69	0.73	0.71		**Semeval**	0.65	0.72	0.68
Cause-Effect	**ADE**	0.69	0.61	0.65	**Cause-Effect**	**ADE**	0.65	0.59	0.62
	Semeval	0.71	0.71	0.71		**Semeval**	0.68	0.73	0.7

three for Meronym-Holonym relations. In Table 3, we present the results for all the datasets showing the precision, recall, and F_1 metrics using `One-Hot-POS Rep` on the left; and `Continuous-Prob-POS Rep` on the right side for sentence-based evaluation. Overall the datasets and various relations, ASPER's performance using `One-Hot-POS Rep` is the best for detecting patterns for Hyponym-Hypernym relation with an F_1 score of 0.74 and 0.80 on Lex and RND datasets respectively. The poorest performance of ASPER for both representations was for the Meronym-Holonym pattern with an F_1 score 0.54, 0.64 for Bless and Phi datasets. On the other hand the Semeval (Meronym-Holonym) dataset achieves an F_1 score 0.71. The reason for the best performance for Hyponym-Hypernym relation is possibly due to well-established patterns for expressing this relation in a sentence. For the other two relations, the syntactic patterns are more fluid and hence, hard to recognize by an automated method. That means, even if a pair holds a semantic relation, only a few sentences have a syntactic pattern. We observe this from sampled test dataset which is labeled manually. A similar argument holds for ADE cause-effect dataset. The performances on both Semeval sub-datasets are comparatively better. This dataset was created for competition and many of the sentences in this dataset are constructed with true cause-effect patterns. Finally, although there are already sentences for Phi dataset, the sentences do not always have consistent syntactic patterns. The right side of Table 3 shows the sentence based pattern extraction results using `Continuous-Prob-POS Rep`. The performance of `Continuous-Prob-POS Rep` never outperforms that of `One-Hot-POS Rep`. One possible reason for that can be the probability values do not actually add information in the pattern extraction task; rather at

times those values can add noise. For instance, if a POS label for a token is *NN*, the probability of that token being *ADP* is probably a noise. The result also indicates that, the POS tags are important. If ASPER is integrated into other languages, an accurate dependency parser, and POS tagger would be needed for discovering syntactic patterns.

Table 4: Pattern Extraction Results of ASPER Evaluated on Sentence: One-Hot-POS-Stanza Rep on Left, Evaluated on Pattern: One-Hot-POS-Stanza Rep on Right

	Evaluated on Sentence				**Evaluated on Pattern**		
Relation	**Dataset**	**Prec**	**Rec**	**F_1**	**Relation**	**Dataset**	**Prec**
Hyponym-Hypernym	**Lex**	0.8	0.7	0.75	**Hyponym-Hypernym**	**Lex**	0.75
	RND	0.86	0.73	0.79		**RND**	0.84
Meronym-Holonym	**Bless**	0.47	0.58	0.52	**Meronym-Holonym**	**Bless**	0.5
	Phi	0.62	0.68	0.65		**Phi**	0.6
	Semeval	0.69	0.73	0.71		**Semeval**	0.66
Cause-Effect	**ADE**	0.66	0.6	0.63	**Cause-Effect**	**ADE**	0.64
	Semeval	0.7	0.69	0.7		**Semeval**	0.66

However, in our experiments with Stanza, we obtained comparable performance to the `One-Hot-POS Rep` of Spacy. Table 4 shows the experimental results, with the left side providing the results evaluated on sentences and the right side providing the pattern evaluated performance. Comparing the results in Table 4 with other results, we observe that Stanza performs slightly better on the Hyponym-Hypernym relation. However, for the Meronym-Holonym relation, the performance is almost the same, and for the Cause-Effect relation, the performance deteriorates. Therefore, we cannot draw a definitive conclusion regarding which representation between Spacy and Stanza provides better patterns.

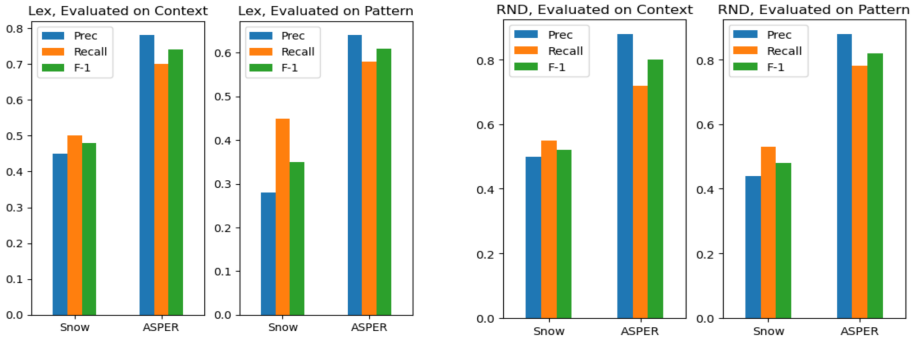
In Table 5, we show the results using the evaluated pattern approach for both representations. The finding is very similar to the results in Table 3. Note that for the evaluation metric based on pattern, only precision is shown. This is due to the fact that for pattern-based evaluation when only one extraction method is used, we have no knowledge about false-negative, so recall cannot be computed.

We compare the performance of ASPER using `One-Hot-POS Rep` with [19]'s work, which works only for Hyponym-Hypernym pattern extraction.

Table 5: Pattern Extraction Results of ASPER Evaluated on Pattern: One-Hot-POS Rep (Labeled as Rep 1 on Left), Continuous-Prob-POS Rep (Labeled as Rep 2 on Right)

Rep 1		
Relation	**Dataset**	**Prec**
Hyponym-Hypernym	**Lex**	0.81
	RND	0.88
Meronym-Holonym	**Bless**	0.54
	Phi	0.64
	Semeval	0.68
Cause-Effect	**ADE**	0.68
	Semeval	0.73

Rep 2		
Relation	**Dataset**	**Prec**
Hyponym-Hypernym	**Lex**	0.78
	RND	0.83
Meronym-Holonym	**Bless**	0.54
	Phi	0.58
	Semeval	0.68
Cause-Effect	**ADE**	0.58
	Semeval	0.68



(a) Pattern Extraction Results in Lex Dataset

(b) Pattern Extraction Results in RND Dataset

Figure 4: Hyponym-Hypernym Pattern Extraction Results

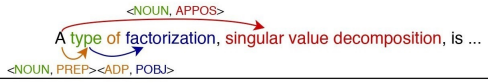
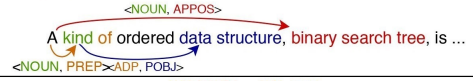
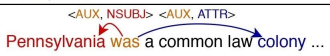
So we show comparison results on Lex and RND datasts for the Hyponym-Hypernym pattern extraction task. This comparison result is shown in the bar charts of Figure 4 using precision, recall and F_1 values of both the pattern evaluation metrics. Both the methods are tuned for the highest F_1 score. As we can see from the bar chart, for both the datasets (Lex on the Left, RND on the right), with respect to both evaluation metrics, ASPER beats Snow's method significantly. In fact, precision, recall, and F_1 of Snow's method are substantially lower (50% lower) than ASPER for both evaluation metrics in both datasets. Although we could not compare ASPER with other methods for meronym-holonym and cause effect patterns extraction due to scarcity of

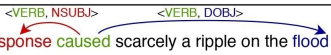
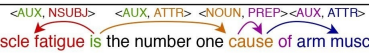
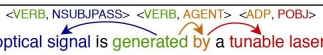
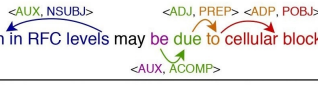
enough automatic pattern extraction works, the results in Table 5, and Table 3 clearly indicate that ASPER performs well on pattern extraction for other relation.

The first column of Figure 5 shows some of the extracted patterns for all the semantic relations using Spacy we work within this research. While there are 29 human readable patterns are shown for hyponym-hypernym relation, the number of syntactic patterns for this relation is 16. Out of this 13 are extracted in Lex dataset, and 14 are extracted in RND. Similarly for meronym-holonym relation, 22 unique syntactic patterns are extracted; among those 12, 14, and 15 patterns are extracted from Bless, Phi, and Semeval datasets respectively. Finally, for cause-effect semantic relation out of 22 unique syntactic patterns, 15 comes from ADE and 16 comes from Semeval datasets. However, our experiments with Stanza led to the discovery of 17 syntactic dependency patterns for the hyponym-hypernym relation, which can be transformed into 28 human-readable patterns. These patterns include 13 from the Lex dataset and 15 from the RND dataset. In the case of the meronym-holonym relation, we found 22 syntactic dependency patterns, with all three datasets (Bless, Phi, and Semeval) contributing 14 patterns each. For the cause-effect relation, we found 22 syntactic dependency patterns, with ADE contributing 17 patterns and Semeval contributing 15 patterns. However, we did not uncover any new human-readable patterns for any of the semantic relations using Stanza. As a result, we only showcase the patterns obtained from Spacy in this research paper.

4.5. Qualitative Pattern Extraction Results

In Figure 5, we show some of the patterns extracted by ASPER. The first column shows the human readable patterns; the second column shows the entity pairs which exhibit the semantic relationship, and finally, the third column shows the sentences with the corresponding semantic pairs and the syntactic patterns extracted by ASPER.

Pattern	Hyponymy-Hypernymy	Sentence
u, a class of w,	(StyleGAN, generative adversarial network)	
the w of u	(Aasu, village)	
a type of w, u,	(singular value decomposition, factorization)	
a kind of w, u,	(binary search tree, data structure)	
u was a w	(Pennsylvania, colony)	

Pattern	Cause-Effect	Sentence
u caused w	(response, flood of commentary)	
u is cause of w	(muscle fatigue, arm muscle pain)	
w generated by u	(tunable laser, optical signal)	
w influenced by u	(immunoprotective effects, tumorigenicity of clones)	
w is due to u	(cellular blockade, reduction in RFC levels)	

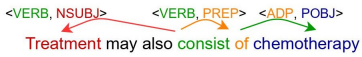
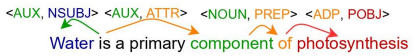
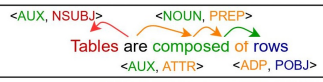
Pattern	Meronym-Holonym	Sentence
u consist of w	(treatment, chemotherapy)	
w component of u	(photosynthesis, water)	
u compose of w	(tables, rows)	
w part of u	(CPU, ALU)	
w element for u	(life, water)	

Figure 5: Example extracted syntactic patterns

For example, the second row of Hyponym-Hypernym patterns in Figure 5, contains the pattern `the w of u`. The word `Aasu` is a hyponym of the hypernym, `village`. For this hyponym-hypernym pair, the sentence `A hiking trail leads to the village of Aasu` is extracted from Wikipedia from which ASPER identifies the dependency edges `village` → `the`, `village` → `of` and `of` → `Aasu`; from the attention values and itemset mining. If we replace `Aasu` with u, and `village` with w we get the hyponym-hypernym pattern, `the w of u` in the first column, which is not reported by Hearst and Snow [1, 19]. Along with finding new hyponym-hypernym patterns, ASPER re-discovers most of the Hearst patterns except some rare ones. For instance, the pattern `u is a special case of w` is infrequent, and ASPER failed to extract it.

On the contrary, ASPER can extract some new patterns which are not reported before. The third row of cause-effect patterns in Figure 5; `w generated by u` is a pattern that is not reported by [21], and the fourth row, `w influenced by u` is not used by [22] for classification. For meronym-homonym, `w element for u` is not used by [23].

4.6. Usability of Syntactic patterns

We also perform experiments to show the utility of syntactic patterns in extracting entities from sentences in an unsupervised fashion. For this experiment, we choose cause-effect relation as this relation is studied well in medical literature for extracting cause and effect entities involving disease, symptoms, etc [95]. Note that, the supervised entity recognition tasks for extracting cause and effect phrases depend on labeled dataset [71]. However, labelling sentence token for a pattern is a time consuming task. In contrast, the syntactic pattern-based approach can extract cause-effect entities effectively, which we want to demonstrate through the results of this experiment. We run this experiment on SemEval dataset.

To find the cause-effect terms, at first, all the noun phrase pairs of a sentence are collected using Spacy 2.2.3 [86] and *en_core_web_sm* package. Secondly, for each pair, dependency edges on the shortest path are collected in the same fashion as described in Subsection 3.3. Now if any syntactic pattern of cause-effect relation matches with the dependency edge-list of a pair of noun phrases, the noun phrases are considered as cause and effect terms. To illustrate with an example let us consider the following sentence *"Most AE-COPD cases are attributed to bacterial or viral respiratory infections."*. If only noun phrases are extracted from this sentence, we have one possible

option u = *bacterial or viral respiratory infection*, and w = *Most AE-COPD cases* where u, w represent probable cause and effect term respectively. The edges in the shortest path from u to w in the dependency tree parsed with Spacy are: [(w, noun, nsubj,attribute),(attribute,verb,prep,to),(to,adp,pobj,u)]

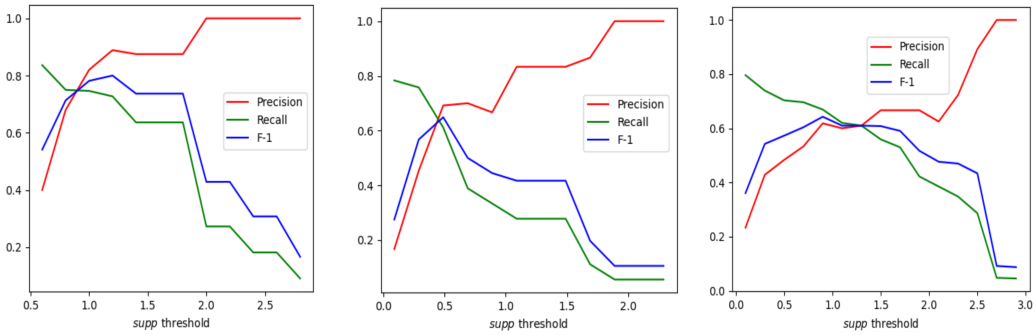
w be attributed to u is a cause-effect template pattern; its syntactic pattern matches with the Spacy shortest path edges for u to w. From this, we can extract *bacterial or viral respiratory infection* as a cause term and *Most AE-COPD cases* as an effect term. We can also allow partial match by considering a fraction of matching edges, where the fraction can be decided through a tunable threshold *thr*. For this experiment, we tuned *thr* is between [50%, 100%] with an interval of 5%. The less this threshold is, the more noisy the extracted candidate pairs are. For our dataset, we find the best result for $thr = 100\%$.

We compare the performance with existing pattern-based approaches, i.e., logical rule-based approach [22] and word vector mapping-based approach [72]. For the rule-based approach, a collection of cause-effect rules are used to extract cause effect candidates in an unsupervised manner. Unlike syntactic patterns, these rules consist of different causative verbs in active or passive form, with or without the preposition. These rules are matched in given sentences to obtain cause and effect phrase candidates. However, not all the candidates they extract contain causal relationship. So, in a second step, they use a supervised binary classification to filter out false positive pairs. To train the classification model, they use the train partition of SemEval dataset, and classify based on that trained model. The word vector mapping based method is proposed for building a causal graph from medical corpora, but this method can extract cause-effect terms as well [72]. It is an unsupervised method that uses regular expression based-dependency parsing. Then pre-trained Skip-Gram method of Word2Vec [96] is used to discover causative verbs with cosine similarity. From those causative verbs and regular expression-based Parts of Speech parsing the authors extract cause-effect terms from sentences. Extracted cause-effect terms are then used to form a causal graph. We use the causality extraction ability of this method and introduce it as one of the causality extraction baseline methods. Note that, we choose the above baseline methods for fair comparison given that the ASPER's syntactic pattern based entity extraction and both the baseline methods are unsupervised.

Table 6 shows the performance of our method along with the three baseline methods discussed above. As we can see from the results, syntactic

Table 6: Performance of PatternCausality and other methods in SemEval Dataset

Method	**Prec**	**Rec**	**F_1**
Logical-Rule Based	0.71	0.47	0.57
Word Vector Mapping Based	0.74	0.47	0.58
Syntactic Pattern from ASPER	**0.75**	**0.51**	**0.61**



(a) RND Ablation (b) ADE ablation (c) Phi ablation

Figure 6: Ablation Study with ASPER (One-Hot-POS Rep)

patterns from ASPER has better results than the competitive methods.

4.7. Ablation Study

The main hyper-parameters of ASPER which affect its pattern extraction performance are *supp*, and *att*. If *supp* is fixed, and *att* is set to a lower value, the probability of getting noisy edges is high. On the contrary, higher *att* values lead to missing of the important edges. As the number of patterns hardly change in this process, we show ASPER's performance using `One-Hot-POS Rep` (which performs best) over varying *supp* values keeping *att* unchanged. The findings are shown in Figure 6. In this Figure, for each plot, support values are shown along with the x-axis and performance values (precision, recall, F_1) are shown along the y-axis. From all the three plots, the F_1-score values increase as *supp* increases reaching the peak, then gradually decreases. With larger *supp* precision always increases, as with higher support more stringent requirements are imposed for the selection of a pattern. On the other hand, the recall curves always go downward direction since the number of predicted patterns decreases as *supp* increases.

5. Future Works and Conclusion

We present ASPER, a novel deep learning model which can extract syntactic patterns shared between entity pairs within a sentential context to convey a semantic relation. It works for any relation, it can predict the existence of a relation, and it can also extract syntactic patterns of that relation—a unique feature that no existing method can offer. We demonstrate ASPER's performance on multiple relations, each on multiple datasets—both benchmarked, and our own creation. The experimental results show that ASPER can extract all known syntactic patterns of a relation, including a few new patterns which are not explicitly stated in the previous works.

Among the weakness, ASPER has a high dependency on the collection of sentences used for extracting syntactic patterns, and the value of support threshold used in ECLAT. So, ASPER often fails to extracts patterns that are rare, and sufficient support for them is absent in the dataset. To overcome this challenge, one need to tune the support threshold thoroughly by using a validation dataset. ASPER may sometimes find only partial patterns or a false positive pattern, however this issue can easily be fixed through human validation. The syntactic patterns that ASPER returns are meant to be used as templates for entity pair extraction from sentences. In that task, several issues may arise. For instance, pattern matching may erroneously declare that the token "insects" holds "is-a" relationship with mammal from a negative statement like *It is not the case that insects are a type of mammal*. In another case, syntactic pattern matching on the sentence, *Deficiency in Vitamin D can cause increased mortality rate in Covid-19 patients* may erroneously extract that "Vitamin D is a cause of mortality", whereas the culprit is "Deficiency in Vitamin D", not Vitamin D itself. So, one needs to use more sophisticated methodologies than simple syntactic pattern matching for entity extraction.

One future work of this research can be to use ASPER for extracting syntactic patterns of other languages. Considering the fact that ASPER is domain-neutral and it only uses dependency tree parser, it can easily be adapted for any other language, assuming that there is an accurate dependency parser for that language and some annotated data. Another future work is to apply ASPER's output patterns for extracting entities from long and complex sentences. Such keywords can then be used as labeled (possibly noisy) training data for a more sophisticated supervised learning models for entity extraction.

Authors are committed to reproducible research, and will release code, and ground truth datasets once the paper is accepted.

6. Acknowledgement

This material is based upon work supported by the National Science Foundation (NSF), USA under Grant No. IIS-1909916.

References

[1] M. A. Hearst, Automatic acquisition of hyponyms from large text corpora, in: Proceedings of the 14th Conference on Computational Linguistics - Volume 2, COLING '92, Association for Computational Linguistics, USA, 1992, p. 539–545. doi:10.3115/992133.992154.
URL https://doi.org/10.3115/992133.992154

[2] R. Patel, Y. Yang, I. Marshall, A. Nenkova, B. Wallace, Syntactic patterns improve information extraction for medical search, 2018.

[3] S. Volkova, D. Caragea, W. H. Hsu, J. Drouhard, L. Fowles, Boosting biomedical entity extraction by using syntactic patterns for semantic relation discovery, in: 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Vol. 1, 2010, pp. 272–278. doi:10.1109/WI-IAT.2010.152.

[4] P. McNamee, R. Snow, P. Schone, J. Mayfield, Learning named entity hyponyms for question answering, in: Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II, 2008.
URL https://www.aclweb.org/anthology/I08-2112

[5] V. Jijkoun, J. Mur, M. de Rijke, Information extraction for question answering: Improving recall through syntactic patterns, in: COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics, 2004, pp. 1284–1290.

[6] K. Ravikumar, M. Rastegar-Mojarad, H. Liu, Belminer: adapting a rule-based relation extraction system to extract biological expression language statements from bio-medical literature evidence sentences, Vol. 2017, Narnia, 2017.

[7] C. Klaussner, D. Zhekova, Lexico-syntactic patterns for automatic ontology building, in: Proceedings of the Second Student Research Workshop associated with RANLP 2011, 2011, pp. 109–114.

[8] S. Ghadfi, N. Béchet, G. Berio, Building ontologies from textual resources: A pattern based improvement using deep linguistic information, in: WOP, 2014.

[9] H. Poon, P. Domingos, Unsupervised ontology induction from text, in: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Uppsala, Sweden, 2010, pp. 296–305.
URL https://www.aclweb.org/anthology/P10-1031

[10] Y. Chasseray, A.-M. Barthe-Delanoe, S. Negny, J.-M. Le Lann, Knowledge extraction from textual data and performance evaluation in an unsupervised context, Information Sciences 629 (2023) 324–343. doi:https://doi.org/10.1016/j.ins.2023.01.150.

[11] E. T. K. Sang, Extracting hypernym pairs from the web, in: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, 2007, pp. 165–168.

[12] S. Yildirim, T. Yildiz, Automatic extraction of turkish hypernym-hyponym pairs from large corpus, in: Proceedings of COLING 2012: Demonstration Papers, 2012, pp. 493–500.

[13] G. Sahin, B. Diri, T. Yıldız, Pattern and semantic similarity based automatic extraction of hyponym-hypernym relation from turkish corpus, in: 2015 23nd Signal Processing and Communications Applications Conference (SIU), IEEE, 2015, pp. 674–677.

[14] C. Wang, Y. Fan, X. He, A. Zhou, Predicting hypernym–hyponym relations for chinese taxonomy learning, Knowledge and Information Systems 58 (2019) 585–610.

[15] S. A. Caraballo, Automatic construction of a hypernym-labeled noun hierarchy from text, in: Proceedings of the 37th annual meeting of the Association for Computational Linguistics, 1999, pp. 120–126.

[16] R. Girju, D. I. Moldovan, Text mining for causal relations, in: FLAIRS Conference, 2002.

[17] M. A. Hearst, Automated discovery of wordnet relations, in: WordNet: An Electronic Lexical Database and Some of Its Applications, 1998.

[18] M. Berl, E. Charniak, Finding parts in very large corpora, 2002. doi:10.3115/1034678.1034697.

[19] R. Snow, D. Jurafsky, A. Y. Ng, Learning syntactic patterns for automatic hypernym discovery, in: L. K. Saul, Y. Weiss, L. Bottou (Eds.), Advances in Neural Information Processing Systems 17, MIT Press, 2005, pp. 1297–1304.

[20] W. van Hage, H. Kolb, G. Schreiber, A method for learning part-whole relations, Vol. 4273, 2006, pp. 723–735.

[21] C. KHOO, J. Kornfilt, R. ODDY, S.-H. Myaeng, Automatic extraction of cause-effect information from newspaper text without knowledge-based inferencing, Vol. 13, 1998, pp. 177–186. doi:10.1093/llc/13.4.177.

[22] A. Sorgente, G. Vettigli, F. Mele, Automatic extraction of cause-effect relations in natural language text., Vol. 2013, Citeseer, 2013, pp. 37–48.

[23] N. Sheena, S. Jasmine, S. Joseph, Automatic extraction of hypernym & meronym relations in english sentences using dependency parser, Vol. 93, 2016, pp. 539–546. doi:10.1016/j.procs.2016.07.269.

[24] V. Phi, Y. Matsumoto, Integrating word embedding offsets into the espresso system for part-whole relation extraction, in: PACLIC, 2016.

[25] M. Baroni, A. Lenci, How we BLESSed distributional semantic evaluation, in: Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics, Association for Computational Linguistics, Edinburgh, UK, 2011, pp. 1–10.
URL https://www.aclweb.org/anthology/W11-2501

[26] S. Necşulescu, S. Mendes, D. Jurgens, N. Bel, R. Navigli, Reading between the lines: Overcoming data sparsity for accurate classification

of lexical relationships, in: Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics, Association for Computational Linguistics, Denver, Colorado, 2015, pp. 182–192. doi:10.18653/v1/S15-1021.
URL https://www.aclweb.org/anthology/S15-1021

[27] E. Santus, F. Yung, A. Lenci, C.-R. Huang, EVALution 1.0: an evolving semantic dataset for training and evaluation of distributional semantic models, in: Proceedings of the 4th Workshop on Linked Data in Linguistics: Resources and Applications, Association for Computational Linguistics, Beijing, China, 2015, pp. 64–69. doi:10.18653/v1/W15-4208.
URL https://www.aclweb.org/anthology/W15-4208

[28] E. Santus, A. Lenci, T.-S. Chiu, Q. Lu, C.-R. Huang, Nine features in a random forest to learn taxonomical semantic relations, in: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), European Language Resources Association (ELRA), Portorož, Slovenia, 2016, pp. 4557–4564.
URL https://www.aclweb.org/anthology/L16-1722

[29] Z. Yu, H. Wang, X. Lin, M. Wang, Learning term embeddings for hypernymy identification, in: Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI'15, AAAI Press, 2015, p. 1390–1397.

[30] I. Sanchez, S. Riedel, How well can we predict hypernyms from word embeddings? a dataset-centric analysis, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, Association for Computational Linguistics, Valencia, Spain, 2017, pp. 401–407.
URL https://www.aclweb.org/anthology/E17-2064

[31] K. A. Nguyen, M. Köper, S. Schulte im Walde, N. T. Vu, Hierarchical embeddings for hypernymy detection and directionality, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 233–243. doi:10.18653/v1/D17-1022.
URL https://www.aclweb.org/anthology/D17-1022

[32] S. Hochreiter, J. Schmidhuber, Long short-term memory, Vol. 9, 1997, pp. 1735–1780.

[33] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, 2015, 3rd International Conference on Learning Representations, ICLR 2015 ; Conference date: 07-05-2015 Through 09-05-2015.

[34] M. J. Hussain, H. Bai, S. H. Wasti, G. Huang, Y. Jiang, Evaluating semantic similarity and relatedness between concepts by combining taxonomic and non-taxonomic semantic features of wordnet and wikipedia, Information Sciences 625 (2023) 673–699. doi:https://doi.org/10.1016/j.ins.2023.01.007.

[35] D.-C. Can, H.-Q. Le, Q.-T. Ha, N. Collier, A richer-but-smarter shortest dependency path with attentive augmentation for relation extraction, in: NAACL, 2019.

[36] P. Li, K. Mao, Knowledge-oriented convolutional neural network for causal relation extraction from natural language texts, Vol. 115, Elsevier, 2019, pp. 512–523.

[37] S. Wu, Y. He, Enriching pre-trained language model with entity information for relation classification, 2019, pp. 2361–2364. doi:10.1145/3357384.3358119.

[38] J. Lee, S. Seo, Y. Choi, Semantic relation classification via bidirectional lstm networks with entity-aware attention using latent entity typing, Vol. 11, 2019, p. 785. doi:10.3390/sym11060785.

[39] P. Shi, J. Lin, Simple bert models for relation extraction and semantic role labeling, 2019.

[40] P. Xu, D. Barbosa, Connecting language and knowledge with heterogeneous representations for neural relation extraction, 2019.

[41] S. Roller, D. Kiela, M. Nickel, 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,

Melbourne, Australia, 2018, pp. 358–363. doi:10.18653/v1/P18-2057.
URL https://www.aclweb.org/anthology/P18-2057

[42] V. Shwartz, Y. Goldberg, I. Dagan, Improving hypernymy detection with an integrated path-based and distributional method, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 2389–2398. doi:10.18653/v1/P16-1226.
URL https://www.aclweb.org/anthology/P16-1226

[43] V. Shwartz, E. Santus, D. Schlechtweg, Hypernyms under siege: Linguistically-motivated artillery for hypernymy detection, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, Association for Computational Linguistics, Valencia, Spain, 2017, pp. 65–75.
URL https://www.aclweb.org/anthology/E17-1007

[44] L. Wang, Z. Cao, G. de Melo, Z. Liu, Relation classification via multi-level attention cnns, 2016, pp. 1298–1307. doi:10.18653/v1/P16-1123.

[45] Y. Shen, X. Huang, Attention-based convolutional neural network for semantic relation extraction, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, The COLING 2016 Organizing Committee, Osaka, Japan, 2016, pp. 2526–2536.
URL https://www.aclweb.org/anthology/C16-1238

[46] G. Ji, K. Liu, S. He, J. Zhao, Distant supervision for relation extraction with sentence-level attention and entity descriptions, in: AAAI, 2017.

[47] G. A. Miller, Wordnet: A lexical database for english, Vol. 38, ACM, New York, NY, USA, 1995, pp. 39–41. doi:10.1145/219717.219748.
URL http://doi.acm.org/10.1145/219717.219748

[48] D. Davidov, A. Rappoport, Unsupervised discovery of generic relationships using pattern clusters and its evaluation by automatically generated SAT analogy questions, in: Proceedings of ACL-08: HLT, Association for Computational Linguistics, Columbus, Ohio, 2008, pp. 692–700.
URL https://aclanthology.org/P08-1079

[49] V. B. Mititelu, Automatic extraction of patterns displaying hyponym-hypernym co-occurrence from corpora, 2006.

[50] K. Sabirova, A. Lukanin, Automatic extraction of hypernyms and hyponyms from russian texts., in: AIST (supplement), Citeseer, 2014, pp. 35–40.

[51] M. N. Nityasya, R. Mahendra, M. Adriani, Hypernym-hyponym relation extraction from indonesian wikipedia text, in: 2018 International Conference on Asian Language Processing (IALP), IEEE, 2018, pp. 285–289.

[52] V.-T. Bui, P.-T. Nguyen, V.-L. Pham, Hypernymy detection forÂ vietnamese using dynamic weighting neural network, in: A. Gelbukh (Ed.), Computational Linguistics and Intelligent Text Processing, Springer Nature Switzerland, Cham, 2023, pp. 234–247.

[53] A. Artale, E. Franconi, N. Guarino, L. Pazzi, Part-whole relations in object-centered systems: An overview, Data & Knowledge Engineering 20 (3) (1996) 347–383, modeling Parts and Wholes. doi:https://doi.org/10.1016/S0169-023X(96)00013-4.

[54] J. Bernauer, Analysis of part-whole relation and subsumption in the medical domain, Data & Knowledge Engineering 20 (3) (1996) 405–415, modeling Parts and Wholes. doi:https://doi.org/10.1016/S0169-023X(96)00016-X.

[55] A. C. Varzi, Parts, wholes, and part-whole relations: The prospects of mereotopology, Data & Knowledge Engineering 20 (3) (1996) 259–286, modeling Parts and Wholes. doi:https://doi.org/10.1016/S0169-023X(96)00017-1.

[56] G. Hinton, How to represent part-whole hierarchies in a neural network, Neural Computation (2022) 1–40.

[57] G. Sahin, Classification of turkish semantic relation pairs using different sources, International Journal of Computer Engineering and Information Technology 8 (10) (2016) 196.

[58] M. Stará, P. Rychlỳ, A. Horák, Evaluation of automatically constructed word meaning explanations, arXiv preprint arXiv:2302.13625 (2023).

[59] L. Hollink, G. Schreiber, B. Wielinga, Patterns of semantic relations to improve image content search, Journal of Web Semantics 5 (3) (2007) 195–203. doi:https://doi.org/10.1016/j.websem.2007.05.002.

[60] G. Sahin, Extraction of hyponymy, meronymy and antonymy relation pairs: A brief survey, International Journal on Natural Language Computing (IJNLC) 6 (2) (2017).

[61] M. E. Winston, R. Chaffin, D. Herrmann, A taxonomy of part-whole relations, Vol. 11, 1987, pp. 417–444. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1207/s15516709cog1104, doi:10.1207/s15516709cog1104.

[62] M. Pennacchiotti, P. Pantel, A bootstrapping algorithm for automatically harvesting semantic relations, in: Proceedings of the Fifth International Workshop on Inference in Computational Semantics, 2006.

[63] A. B. Nikulásdóttir, M. Whelpton, Automatic extraction of semantic relations for less-resourced languages, in: Proceedings of the Workshop" Wordnets and other Lexical SemanticResources-between Lexical Semantics, Lexicography, Terminology and Formal Ontologies", NODALIDA, 2009, pp. 1–6.

[64] A. Ahne, V. Khetan, X. Tannier, M. I. H. Rizvi, T. Czernichow, F. Orchard, C. Bour, A. Fano, G. Fagherazzi, et al., Extraction of explicit and implicit cause-effect relationships in patient-reported diabetes-related tweets from 2017 to 2021: Deep learning approach, JMIR Medical Informatics 10 (7) (2022) e37201.

[65] N. Asghar, Automatic extraction of causal relations from natural language texts: a comprehensive survey, arXiv preprint arXiv:1605.07895 (2016).

[66] O. Hassanzadeh, D. Bhattacharjya, M. Feblowitz, K. Srinivas, M. Perrone, S. Sohrabi, M. Katz, Causal knowledge extraction through large-scale text mining, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 13610–13611.

[67] M. Bhandari, M. Feblowitz, O. Hassanzadeh, K. Srinivas, S. Sohrabi, Unsupervised causal knowledge extraction from text using natural lan-

guage inference (student abstract), in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 15759–15760.

[68] P. Wei, J. Zhao, W. Mao, Effective inter-clause modeling for end-to-end emotion-cause pair extraction, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 3171–3181.

[69] R. Xia, Z. Ding, Emotion-cause pair extraction: A new task to emotion analysis in texts, arXiv preprint arXiv:1906.01267 (2019).

[70] C. S. Khoo, J. Kornfilt, R. N. Oddy, S. H. Myaeng, Automatic extraction of cause-effect information from newspaper text without knowledge-based inferencing, Literary and linguistic computing 13 (4) (1998) 177–186.

[71] T. Dasgupta, L. Dey, R. Saha, A. Naskar, Automatic curation and visualization of crime related information from incrementally crawled multi-source news reports, in: Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations, Association for Computational Linguistics, Santa Fe, New Mexico, 2018, pp. 103–107.
URL https://www.aclweb.org/anthology/C18-2023

[72] N. An, Y. Xiao, J. Yuan, Y. Jiaoyun, G. Alterovitz, Extracting causal relations from the literature with word vector mapping, Vol. 115, 2019, p. 103524. doi:10.1016/j.compbiomed.2019.103524.

[73] P. Nakov, Noun compound interpretation using paraphrasing verbs: Feasibility study, 2008, pp. 103–117. doi:10.1007/978-3-540-85776-1_10.

[74] A. Ritter, S. Clark, M. Mausam, O. Etzioni, Named entity recognition in tweets: An experimental study, 2011, pp. 1524–1534.

[75] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, 2016, pp. 260–270. doi:10.18653/v1/N16-1030.

[76] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 2017.

[77] Z. Haoyu, Y. Gong, Y. Yan, N. Duan, J. Xu, J. Wang, M. Gong, M. Zhou, Pretraining-based natural language generation for text summarization, 2019.

[78] Y. Liu, M. Lapata, Text summarization with pretrained encoders, 2019.

[79] M. Di Gangi, M. Negri, M. Turchi, Adapting transformer to end-to-end spoken language translation, 2019, pp. 1133–1137. doi:10.21437/Interspeech.2019-3045.

[80] M. Di Gangi, M. Negri, R. Cattoni, R. Dessi, M. Turchi, Enhancing transformer for end-to-end speech-to-text translation, 2019.

[81] J. Lafferty, A. Mccallum, F. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, 2001, pp. 282–289.

[82] Z. Li, Q. Li, X. Zou, J. Ren, Causality extraction based on self-attentive bilstm-crf with transferred embeddings, Vol. 423, 2019. doi:10.1016/j.neucom.2020.08.078.

[83] M. Tang, P. Gandhi, M. A. Kabir, C. Zou, J. Blakey, X. Luo, Progress notes classification and keyword extraction using attention-based deep learning models with bert, ArXiv abs/1910.05786 (2019).

[84] R. Caruana, S. Lawrence, C. L. Giles, Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping, in: T. K. Leen, T. G. Dietterich, V. Tresp (Eds.), Advances in Neural Information Processing Systems 13, MIT Press, 2001, pp. 402–408.

[85] M. Covington, A fundamental algorithm for dependency parsing, 2001.

[86] M. Honnibal, I. Montani, spaCy 2.2.3: Industrial-strength natural language processing, in: https://spacy.io/, 2020.

[87] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, C. D. Manning, Stanza: A Python natural language processing toolkit for many human languages, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2020.

[88] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. St. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, B. Strope, R. Kurzweil, Universal sentence encoder for English, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 169–174. doi:10.18653/v1/D18-2029.
URL https://www.aclweb.org/anthology/D18-2029

[89] M. J. Zaki, Scalable algorithms for association mining, Vol. 12, 2000, pp. 372–390. doi:10.1109/69.846291.

[90] V. Shwartz, Y. Goldberg, I. Dagan, Improving hypernymy detection with an integrated path-based and distributional method, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 2389–2398. doi:10.18653/v1/P16-1226.
URL https://www.aclweb.org/anthology/P16-1226

[91] M. Baroni, A. Lenci, How we blessed distributional semantic evaluation, 2011, pp. 1–10.

[92] R. Girju, P. Nakov, V. Nastase, S. Szpakowicz, P. Turney, D. Yuret, SemEval-2007 task 04: Classification of semantic relations between nominals, in: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), Association for Computational Linguistics, Prague, Czech Republic, 2007, pp. 13–18.
URL https://www.aclweb.org/anthology/S07-1003

[93] I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, D. Ó Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, S. Szpakowicz, SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals, in: Proceedings of the 5th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Uppsala, Sweden, 2010, pp. 33–38.
URL https://www.aclweb.org/anthology/S10-1006

[94] H. Gurulingappa, A. Mateen, A. Roberts, J. Fluck, M. Hofmann-Apitius, L. Toldo, Development of a benchmark corpus to support

the automatic extraction of drug-related adverse effects from medical case reports, Vol. http://dx.doi.org/10.1016/j.jbi.2012.04.008, 2012. doi:10.1016/j.jbi.2012.04.008.

[95] X. L. M. Ahsanul Kabir, AlJohara Almulhim, M. A. Hasan, Informative causality extraction from medical literature via dependency tree based patterns, 2022.

[96] T. Mikolov, G. Corrado, K. Chen, J. Dean, Efficient estimation of word representations in vector space, 2013, pp. 1–12.

7. Appendix

Table 7 shows all the hyponym-hypernym human readable patterns extracted by ASPER. However, the first three template patterns we show are syntactically similar. We group the patterns for the sake of presentation. Most of the patterns shown in a group can be generalized by syntactic syntactic patterns. The ability to generalize different patterns is a strength for syntactic patterns. That is why the number of syntactic patterns we found are actually less than human readable patterns. Table 8, and 9 show the cause-effect and meronym-holonym patterns retrieved by ASPER respectively.

Table 7: All Hyponym-Hypernym patterns extracted by ASPER

Pattern	**u**	**w**	**Sentence**
u, a class of w	Core 2 Duo	microprocessor	Core 2 Duo, a class of early Desktop microprocessor had much lower core frequency and approximately the same FSB frequency and level 2 cache size as Pentium D microprocessors
a class of w, u	Core 2 Duo	microprocessor	A class of early Desktop micro-processor, Core 2 Duo, had much lower core frequency and approximately the same FSB frequency and level 2 cache size as Pentium D microprocessors.
u be a class of w	Core 2 Duo	microprocessor	Core 2 Duo is a class of early Desktop microprocessor which had much lower core frequency and approximately the same FSB frequency and level 2 cache size as Pentium D microprocessors.
u, a family of w a family of w, u u be a family of w	Vinyasa	yoga	Vinyasa, a family of yoga is dynamic and ever-flowing.
u, a type of w a type of w, u u be a type of w	system software	computer software	The system software, a type of computer software is designed for running the computer hardware parts and the application programs
u, a kind of w a kind of w, u u be a kind of w	panda	bear	Panda, a kind of bear is found only in China.
w, including u w which/that include u w include u	Asiatic black bear	bear	Some species of bears, including Asiatic black bears and sun bears, are also threatened by the illegal wildlife trade.
w, such as u w, for example u w, like u like many w, u	sheep	domesticated animal	Domesticated animals, such as sheep or rabbits, may have agricultural uses for meat, hides and wool.
the w of u	Aasu	village	The village of Aasu along with Aoloau are jointly called O Leasina
u be w u be the w u be a w	panda	bear	The giant pandas are true bears, and part of the family Ursidae
u become w	kizzy	singer	In 2005, Kizzy became the lead singer of the "Bo Winiker Orchestra" with whom she performed for Bill Clinton, Glenn Close and with whom she gained critical acclaim for performing songs in Hebrew.
w named u w called u	ponikve	village	Like other villages named Ponikve and similar names, it refers to a local landscape element.
w as u	Emperor	band	Since the 1990s, Norway's export of black metal, a lo-fi, dark and raw form of heavy metal, has been developed by such bands as Emperor, Darkthrone, Gorgoroth, Mayhem, Burzum and Immortal.
w "u"	Clarens	village	A commission was appointed in 1912 to finalize negotiations, and a decision was made to name the village "Clarens" in honour of President Paul Kruger influence in the area.

Table 8: All Cause-Effect patterns extracted by ASPER

Pattern	**u**	**w**	**Sentence**
w caused by u u cause w u be a cause of w w be attributed to u u be causes of w	sorafenib treatment	severe interstitial pneumonia	In this article, we describe a japanese patient with severe interstitial pneumonia probably caused by sorafenib treatment for metastatic renal cell carcinoma.
w induced by u u induce w	mizoribin administration	A case of siadh	A case of SIADH induced by mizoribin administration.
u lead to w	peripheral neuropathy	linezolid	However, peripheral neuropathy and bone marrow depression led to linezolid withdrawal in seven patients, and neuropathy may not be fully reversible in all patients.
w be associated with u	sulfasalazine	pulmonary infiltrates	Pulmonary infiltrates and skin pigmentation are associated with sulfasalazine. .
w related to u	flecainide	interstitial hypoxaemiant pneumonitis	We describe a case of interstitial hypoxaemiant pneumonitis probably related to flecainide in a patient with the LEOPARD syndrome, a rare congenital disorder. .
u result in w w be result of u	flucloxacillin	fatal hepatic injury	It is well-recognized that flucloxacillin may occasionally result in fatal hepatic injury.
w from u	exertion	satisfaction	I have always drawn satisfaction from exertion, straining my muscles to their limits. .
w be triggered by u u trigger w	earthquake	tsunami	A large tsunami is triggered by the earthquake spread outward from off the Sumatran coast.
w come from u	fear	blockage	Sometimes the blockage comes from fear, as for a CEO who hates public speaking but must give frequent speeches. .
w be the effect of u w, the effect of u	acupuncture	pain relief	Pain relief is the effect of acupuncture which lasts for an extended period of time, sometimes months after the needle was removed.
u produce w w produced by u	Ambient vanadium pentoxide dust	irritation	Ambient vanadium pentoxide dust produces irritation of the eyes, nose and throat.
u promote w	antiwar demonstrators	positive values	He created and advocated flower power,"a strategy in which antiwar demonstrators promoted positive values like peace and love to dramatize their opposition to the destruction and death caused by the war in Vietnam."
u generate w w generated by u	tunable laser	optical signal	The optical signal is generated by a tunable laser.
u influence w w influenced by u	tumorigenicity of clones	immunoprotective effects	The tumorigenicity of clones may be influenced by immunoprotective effects.
w due to u w because of u	Incorrect design	Failure in physical containment	Failures in physical containment may occur due to incorrect design.

Table 9: All Meronym-Holonym patterns extracted by ASPER

Pattern	**u**	**w**	**Sentence**
u consist of w u comprise of w	treatment	chemotherapy	Treatment may also consist of chemotherapy
w part of u w element for u w source of u w component of u w constituent of u	Lowe Group	Lowe	Lowe is part of the Lowe Group, one of the three large subsidiaries of Interpublic.
u made of w	Gene	DNA	Genes are made of DNA
w block of u	Muscle	Protein	Protein is the building block of muscle
u have w	The commission	seven members	The commission shall have seven members.
u group of w	arthropods	invertebrates hypoxaemiant pneumonitis	Arthropods are a group of invertebrates.
u mixture of w	Concrete	cement	Concrete is a mixture of cement.
u have number of w	Arrays	elements	Arrays can have any number of elements.
u combination of w	green	blue	Green is a combination of blue and yellow. .
u collection of w	society	individual	Society is now a collection of individuals
u branch of w	Chinese medicine	Medical Qigong	Medical Qigong is a branch of traditional Chinese medicine
w of u	government	member	The prime minister shall inform all members of the government
w ingredient in u	Ephedrine	Ephedra	Ephedra is a key ingredient in Ephedrine
w with u	peacock	overgrown beak	I have an Indian Blue peacock with an overgrown beak
w member of u	United Nations	Israel	Israel is a member of the United Nations.
u include w	Symptom	vomiting	Symptoms can include vomiting