Natural Language Engineering **24** (5): 763–795. © Cambridge University Press 2018
doi:10.1017/S1351324918000268

Extraction of templates from phrases using Sequence Binary Decision Diagrams

D. HIRANO[1], K. TANAKA-ISHII[2] and A. FINCH[3]

[1]*Graduate School and Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka, Japan*
e-mail: hirano@limu.ait.kyushu-u.ac.jp
[2]*Research Center for Advanced Science and Technology, University of Tokyo, Tokyo, Japan*
e-mail: kumiko@cl.rcast.u-tokyo.ac.jp
[3]*National Institute of Information and Communications Technology Kyoto, Kyoto, Japan*
e-mail: andrew.finch@gmail.com

(*Received 22 March 2017; revised 7 June 2018; accepted 7 June 2018*)

Abstract

The extraction of templates such as 'regard X as Y' from a set of related phrases requires the identification of their internal structures. This paper presents an unsupervised approach for extracting templates on-the-fly from only tagged text by using a novel relaxed variant of the Sequence Binary Decision Diagram (SeqBDD). A SeqBDD can compress a set of sequences into a graphical structure equivalent to a minimal deterministic finite state automata, but more compact and better suited to the task of template extraction. The main contribution of this paper is a relaxed form of the SeqBDD construction algorithm that enables it to form general representations from a small amount of data. The process of compression of shared structures in the text during Relaxed SeqBDD construction, naturally induces the templates we wish to extract. Experiments show that the method is capable of high-quality extraction on tasks based on verb+preposition templates from corpora and phrasal templates from short messages from social media.

1 Introduction

The extraction of frequently appearing templates such as 'regard X as Y' or 'Magnitude X earthquake at Y' is a challenging problem, since it requires an analyzer to capture the structure of the frequently appearing phrases, and detect which part should be represented by the slots. In this article, a template is defined as a frequently co-occurring subsequence of words containing one or more slots, where a slot is replaceable by another word sequence. In other words, a template in this article is a sequence of elements of length longer than one, where an element is either a word or a slot: at least one element is a slot, the other elements are words. The template itself, or the word sequences filling the slots in this article are not necessarily limited to linguistic Noun phrase (NP), Verb phrases (VPs) and PPs.

Classically, such templates have been considered in relation to collocations such as in Smadja (1993). Collocations without slots are readily extractable by finding *n*-grams that form atomic chunks. Therefore, researchers have been turning their

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

attention to the more difficult and general task of extracting templates with slots. The problem has been addressed in part in the form of the extraction of verb structures. Baldwin and Kim (2010) showed the possibility of MWEs (multi-word expressions) extraction by only using *n*-grams and co-occurrences, without using a parser.

Since templates are related to the grammatical structure of the phrases, one possibility is to use a parser. However, the kinds of text and languages that can be processed by a parser are strictly limited, since they are generally built from clean annotated corpora. In addition parse trees can be deep, and the recursive nature of parse trees means there are no limits on the depth of the structures involved, potentially leading to a heavy computational burden.

Templates, as defined at the beginning of the article, are commonly found in the social media: social media text is usually short (around 140 characters in the case of Twitter) and many distributers use automatic bots to communicate the recent database updates. We consider the extraction of these templates as one target application of our approach. Social media text can cause issues even with a well-trained parser, since their word sequences are often non-grammatical. Furthermore, the grammatical structures found in this type of text are typically shallow making a full-blown parser less appropriate for this task than a technique that focuses on the shallow structure directly. It may be possible to develop an unsupervised parsing technique to handle social media text, but a general high quality method for unsupervised parsing is still being sought (Headden, Johnson and McClosky 2009). One possible explanation for this is the difficulty inherent in analyzing the structure recursively. Another related field, shallow parsing (Abney and Abney 1991; Huang, Xu and Yu 2015) is well-established and devoted to obtaining shallow linguistic analyses. These techniques typically segment a sentence into grammatical structures. While pursuing this line may have led to fruitful results, instead, we chose to try an alternative unsupervised method, a graphical method that produces a flat analytical structure, as opposed to the potentially unlimited depth of the structures that can be created by a full parser.

Our proposed unsupervised method extracts templates through the construction of a structure equivalent to a minimum deterministic finite state automaton (minDFA) (Daciuk *et al.* 2000). Our implementation uses Sequence Binary Decision Diagrams (SeqBDDs) to allow the automata to be more memory efficient than traditional minDFA (around twenty per cent was reported in Denzumi *et al.* (2016)), and furthermore permits the graph to be constructed on-the-fly.

The technique we propose facilitates the mining of text from the social media, but can also find application in the more general field of lexicography and corpus linguistics. Patterns of word usage can play an important role for enriching dictionaries, and moreover serve language learners as in the lexical templates found in the Collins Cobuild English Dictionary (Francis, Hunston and Manning 1998). As a means of determining such templates, the key word in context (KWIC) technique is often employed to rapidly display words in their textual context, although the template identification must be done laboriously by hand.

The primary contribution of this paper is to propose an algorithm for automatically constructing graphical representations of text designed specifically for automatic

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

template extraction. The SeqBDD (or minDFA) is the starting point of this work, and we show empirically that it is capable of generating templates with high precision, but suffers from low recall. We propose a relaxed algorithm for SeqBDD construction to overcome this fundamental shortcoming. Usually, attempting to raise recall leads to a lowering of precision, but in this work, the relaxation process increases the number of templates being extracted while at the improving their quality, leading to gains in both precision and recall.

2 Related work

We are interested in extracting the templates as defined at the beginning of Introduction. Although word sequences filling the slots in this article are not necessarily limited to linguistic units, such as NP, VPs and PPs, we want the results to be meaningful linguistically. The templates studied in this article can be considered MWEs with slots, where MWEs are coined (Baldwin and Kim 2010), as 'the split configuration of the verb–particle constructions'. A detailed summary of the acquisition of MWEs is well set out in this reference. In brief, active research in MWEs was triggered by the seminal paper of Sag *et al.* (2002). In the work that followed, various attempts have been made to acquire MWEs, such as co-detection of English and Japanese templates by statistical methods (Kita *et al.* 1994), or the extraction of verbs and noun-type idiom candidates (Fazly 2006). Duan *et al.* (2006) reapply a method to extract frequent patterns from DNA sequences. Since the basic technique is based on finding longest common subsequences, the method does not handle gaps. Several different methods for extracting MWEs using a bilingual corpus have also been proposed (Zarrießand Kuhn 2009; de Medeiros Caseli *et al.* 2010).

In more recent work, (Nagy T. and Vincze 2014) focus on the extraction of verb-particle combinations, each of which has a verb and a particle. Their method can handle gaps, since it is a syntax-based method that uses a parser, but this requires a fully annotated gold standard. A more recent work attempts to extract MWEs without wildcards. Tu and Roth (2012) study the method to detect phrasal verbs using machine learning techniques. Although slots are considered in this article, the application is limited to six verbs and nineteen prepositions. Some recent studies attempt to extract from grammatically structured data, such as treebank and parsed data. Sangati and van Cranenburgh (2015) extract MWEs from frequently appearing fragments in the treebank. Since the method is able to exploit a treebank, the accuracy of the method is higher than the unsupervised methods. Another study (Martens 2010) attempts to find frequent subtrees from parsed data. Our approach differs from these in that it assumes no grammatical structure in the input.

From the opposite perspective, the proposed method is aimed at discovering the underlying structure in related phrases in an unsupervised manner, and as such it is broadly related to work in the field of unsupervised grammar induction.

In the field of natural language processing, the induction of grammars more complex than regular grammars has been actively studied. In Klein and Manning (2004), an unsupervised dependency parser was proposed that successfully induced structure from a modest amount of data. The success of this early approach stimulated further

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

research in the field. Headden *et al.* (2009) introduced valence frames and lexical information into their induction process, giving rise to a substantial improvement in performance (Gimpel and Smith 2012). The method of Klein and Manning (2004) was improved in Gimpel and Smith (2012) by introducing concave models that could be optimized without the problem of local minima. However, at present the accuracy of the state-of-the-art unsupervised methods evaluated on Penn Treebank text (Marcus, Marcinkiewicz and Santorini 1993) is below seventy per cent when the relative to the state-of-the-art supervised methods achieve over ninety-three percent (Klein and Manning 2004; Gimpel and Smith 2012). Therefore, it may be difficult to replace current supervised parsers with unsupervised parsers to use for template extraction.

A template considered in this work has the basic structure of a regular expression (that we will model using automaton-related data-structures), and therefore next we summarize previous work related to regular expression induction that directly concerns template induction. The work can be roughly partitioned into three: heuristics, regular grammar induction and methods related to finite state automata. Many have performed template induction based on heuristics, such as Han, Pei and Yin (2000), Pei *et al.* (2001), Cui, Kan and Chua (2004), and Kim (2010), just to mention a few. Most of the approaches are original but none are theoretically grounded. The remainder of the previous work related to regular expressions presented here has some mathematical framework as a basis.

Regular grammar induction algorithms are able to extract a regular grammar from a set of phrases with similar structure. These techniques have had wide application to sequences in general including the extraction of DNA patterns. (Fernau 2009) suggested a simple bottom-up regular expression induction strategy that we replicated and applied to natural language. Unfortunately, it resulted in a huge number of trivial regular expressions. Moreover, it is not straightforward to apply the method to template extraction since it has no mechanism for slot creation. Most of the previous work discussed so far is based on prefix sharing of similar phrase instances. This, however, generates huge amount of template candidates that have similar infix and suffix structures. The point of phrasal template induction is to identify shared common structures anywhere in the phrases, and abstract them into slots.

Template extraction has a broad range of potential applications. One potential application is to support lexicographers in building/enriching dictionaries for humans and machines. The application can take a dynamic form, for example, by extending the KWIC system (Hunston and Francis 2000), which shows the words immediately preceding and following the node word, up to a predetermined length. In fact, our work was partly motivated by an idea presented in 'pattern grammars' (Hunston and Francis 2000). The authors assembled the Collins Cobuild Grammar Templates dictionary Francis, Hunston and Manning (1996); Francis *et al.* (1998), and claim that texts are formed of *patterns* defined as word sequences that may contain slots (and we use the term *template* to represent this concept hereafter). The patterns allow for parsing any sentence using pattern grammar, as each word in the sentence will occur with its own pattern. Hunston and Francis (2000) show this under the heading

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

of 'pattern flow'. Although the Collins Cobuild English Dictionary was constructed manually, they suggest the possibility of automating the procedure with an extended form of the KWIC system. This work takes a first step to answer this challenge, by means of the SeqBDD. This work was motivated by this fundamental work, but the templates we extract are different from patterns of this Pattern Grammar, in which patterns are defined as sequences of phrase or clause types (e.g., 'that-clause', 'noun phrase') that are dependent on specific verbs, nouns or adjectives.

Another genre of application, as mentioned in the previous section, relates to social media, which has abundant repetitive patterns and shallow structure. Baldwin *et al.* (2013) shows these common characteristics threading through social media, through building and analyzing a corpus consisting of various types of social media. This suggests how patterns could be one possible key to information extraction. The identification of patterns would also create a means of compressing the data. This paper will later study the proposed method in the context of these two potential applications: the verb+preposition template extraction as an example toward an extended KWIC method, and pattern acquisition from social media.

3 The sequence BDD

Given a set of linguistic phrases with a similar structure, we want to extract the common structure as a *template*: a sequence of words and slots. The first attempt of doing this made use of a Trie structure. The Trie, however, does not allow for sharing the infix and suffix. Broadly, for template extraction, we are interested in representations that also allow for the sharing of common structures within infixes and suffixes.

Among the different data structures that allow this type of sharing, we chose the SeqBDD as our basis. The SeqBDD is a descendant of two different formal methods: of BDD and acyclic DFA.[1]

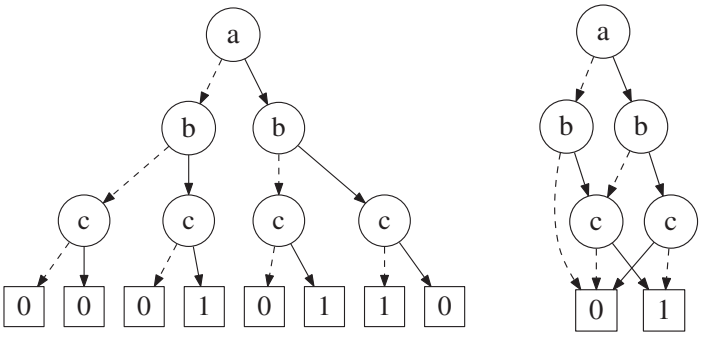
Therefore, the explanation of SeqBDD has two aspects: SeqBDD as one descendant of BDD; and the other in contrast to DFA. We provide the former in Section 3.1, and the latter in Section 3.2. The basis of the SeqBDD can be found in Loekito, Bailey and Pei (2010), Denzumi *et al.* (2016), and the definitions and algorithms necessary to replicate our method are all provided in this article.

3.1 SeqBDD as a descendant of BDD

SeqBDDs are descendants of BDDs (Bryant 1986) that are graphical structures able to express Boolean functions. For example, consider the Boolean function '$ab\bar{c} \vee a\bar{b}c \vee \bar{a}bc$'. The function can be represented as the binary decision tree structure shown in Figure 1(a). Solid edges in the graph represent truth (1-edge), and dashed edges represent falsehood (0-edge). The leaf nodes represent final truth and falsehood as 1 and 0, respectively. It is clear from the figure that the tree

[1] The reason why DFA are acyclic is that all string sets we need to model are finite. Similarly, the SeqBDD is also acyclic.

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press



(a) Binary decision tree (b) BDD

Fig. 1. A binary decision tree and corresponding BDD structures representing the Boolean function $ab\bar{c} \vee a\bar{b}c \vee \bar{a}bc$.

contains many common substructures. For example, a node labeled '*c*' is connected to the 0- and 1-terminal nodes multiple times. The study of BDD has tackled the elimination of such redundancy in binary decision trees (Bryant 1986; Knuth 2009).

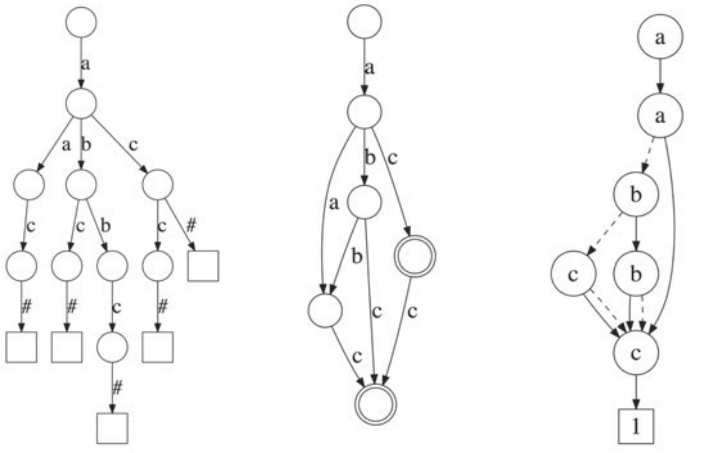
The minimization of the representation is conducted by exhaustively applying two reduction operations for merging identical subgraphs: (1) node sharing (share all equivalent subgraphs) and (2) node deletion (delete all redundant nodes where both outgoing edges lead to equivalent subgraphs) (Bryant 1986; Knuth 2009). Equivalent subgraphs are defined recursively: two nodes are in equivalent subgraphs if their labels are identical and the heads of their outgoing edges are also in the equivalent subgraphs. For example, in Figure 1(a) the central two nodes labeled '*c*' are both in equivalent subgraphs that extend down to the leaf nodes. These subgraphs are merged into the left '*c*' node at the bottom in Figure 1(b) using the node sharing operation.

As for the node deletion, the left-most node labeled '*c*' in Figure 1(a) is deleted using the node deletion operation (2) since both outgoing edges lead to the 0-terminal node. Its incoming edge (from node '*b*') is redirected to the shared equivalent subgraph of its children (the 0-terminal node). It is clear that the graph in (b) is considerably more compact than (a).

A SeqBDD (Loekito *et al.* 2010) is a BDD dedicated to representing a set of sequences. A set of sequences is treated as a Boolean function; the function being True if the sequence is a member of the set, and False otherwise. Since the number of False members are infinite, they are not explicitly shown in SeqBDD. Paths along the *solid* arrows through the graph encode members of the set. For example, Figure 2(c) represents the set: $\{ac, abc, aac, acc, abbc\}$,where '*acc*' is represented by $a \rightarrow a \dashrightarrow b \dashrightarrow c \rightarrow c \rightarrow 1$ (i.e., encode *a*, then do not encode *a*, not *b*, encode *c*, encode *c*). Note that a path can include False edges (dashes), but these do not encode elements of the sequence. A symbol in the sequence is generated from the label of the tail of a traversed edge if the edge is a True edge.

When building a SeqBDD, it is not necessary to first construct a (potentially huge) binary decision tree and then apply reduction rules to reduce it. Rather,

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press



(a) Trie (b) min-DFA (c) SeqBDD

Fig. 2. Equivalent Trie, minDFA and SeqBDD representing: $\{ac, abc, aac, acc, abbc\}$.

for SeqBDD, the final compact structure is built on-the-fly (Loekito *et al.* 2010; Denzumi *et al.* 2016), by merging two SeqBDD graphs, basically composed of node sharing and deletion rules.[2] The obtained SeqBDD graph is minimal, having the minimal number of nodes necessary to accept a given set of input.

3.2 SeqBDD versus minimal DFA

A SeqBDD is one way to represent a set of sequences in a compact manner, another is a DFA, defined as follows:

acyclic DFA : $\langle \Sigma, \Gamma, \delta, s_{start}, s_{end} \rangle$,
SeqBDD : $\langle \Sigma, V, E, s_{start}, \{0, 1\} \rangle$,

where each element in the $\langle \ldots \rangle$ tuples denotes the set of accepted elements (the alphabet so far), set of nodes, set of edges, start[3] and end states, from left to right, respectively. They differ essentially in that BDD labels a node, whereas DFA labels an edge. A Trie is one such DFA. For example, a Trie representing the set of sequences $\{ac, abc, aac, acc, abbc\}$ is shown in Figure 2(a).

Similar to the original binary decision tree, there are many common edges that can be shared, such as the edges from '*c*' to the final node. A minDFA, a DFA with the smallest number of nodes that accepts the same set of sequences, can be constructed using the algorithms, originally proposed in Daciuk *et al.* (2000). Briefly, two nodes are contracted when the labels of the incoming edges and outgoing edges are identical. For our example, the proposed algorithm generates Figure 2(b). (b) is considerably more compact than (a). We now have two different compact graphical

[2] A faster construction is proposed by use of union operator (Bryant 1986; Knuth 2009).
[3] The start states are suppressed in Figures 1 and 2, to save space.

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

representations: the SeqBDD (Figure 2(c)) and the minDFA (Figure 2(b)) for the same set of sequences.

Comparing the two graphs even in this simple toy example, we see three advantages of SeqBDD (c) over minDFA (b). First, and above all, with respect to our objective of template extraction, (b) does not show the template *aXc*, since the *c* labels are distributed over multiple edges and are not contracted. This is different in (c), where we clearly see the template. This natural ability to encode templates arises from the fundamental difference that the SeqBDD labels *nodes*.

Second, (b) is less compact compared with the same representation in the SeqBDD shown in (c), since (b) has nine labeled graphical objects, whereas (c) only has six. This can be theoretically analyzed using the mathematical notation presented at the beginning of this subsection. Let $|X|$ denotes the number of elements of a given set X. It has been recently proven (Denzumi *et al.* 2016), that minDFA requires $O(|\delta|)$, whereas SeqBDD $O(|V|)$ computational space complexity, and furthermore, $|V| \leq |\delta|$ where minDFA can be $|\Sigma|$ times larger than the former (Denzumi *et al.* 2016). In other words, a SeqBDD is theoretically never larger than the equivalent minDFA. Moreover, the article reports an empirical comparison of memory use and SeqBDD reduced memory requirements by 10–22 per cent relative to a minDFA. This is a considerable amount when considering that (as will be explained in the experimental section) a single target template can require several gigabytes of allocation.

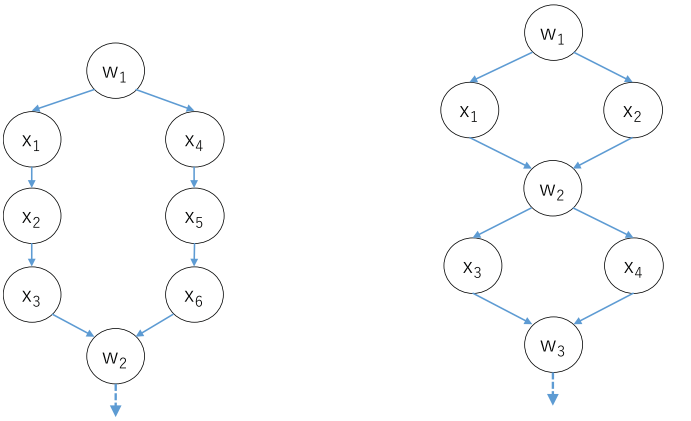
Third, the algorithm proposed in Daciuk *et al.* (2000) is entitled *incremental*, but it is applied to a *given graph* incrementally. In other words, the algorithm of Daciuk *et al.* (2000) is not online. This can be very costly for a large-scale natural language application, since a huge Trie graph structure must be rendered in memory prior to applying the algorithm (Daciuk *et al.* 2000). This contraction requires $O(|\delta|)$, computational time complexity where δ is the number of edges. In contrast, as noted earlier, the application of Loekito *et al.* (2010) and Denzumi *et al.* (2016) effects a direct, incremental construction of the minimal representation.

4 Relaxed SeqBDD

As will be shown in the evaluation section, a straightforward application of SeqBDD as a means to extract a set of templates leads to a result of low coverage (recall), although the quality of output templates is high. This is a consequence of the strictness of the node sharing operation. The SeqBDD faithfully represents the supplied dataset, but it is not clear whether this data structure is optimal with respect to template extraction.

An example is given in Figure 3, where the w_j are lexical items and x_i are the potential slots. The original SeqBDD can learn the structure in the left figure from just two training examples (i.e., $\{w_1x_1x_2x_3w_2, w_1x_4x_5x_6w_2\}$). To learn the structure shown in the right figure, however, it requires examples of four different forms to appear (i.e., $\{w_1x_1w_2x_3w_3, w_1x_1w_2x_4w_3, w_1x_2w_2x_3w_3, w_1x_2w_2x_4w_3\}$). If even one of these phrases did not appear in the input set, then the graph structure gets separated, as in for example the structure in the left figure and the potential slot before and after w_2 cannot be captured. In general, when a template has n slots,

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press



(a) SeqBDD constructed with two phrases

(b) SeqBDD constructed with four phrases

Fig. 3. (Colour online) (a) is when two phrases $\{w_1x_1x_2x_3w_2, w_1x_4x_5x_6w_2\}$ are given as input. (b) is when four phrases $\{w_1x_1w_2x_3w_3, w_1x_1w_2x_4w_3, w_1x_2w_2x_3w_3, w_1x_2w_2x_4w_3\}$ are given as input.

a number of corresponding phrases that must be included in the input set grows combinatorially with respect to the variety of the items around the n slots. Although such combination is possible in natural language, the huge number of required occurrences cannot be actually observed in real data.

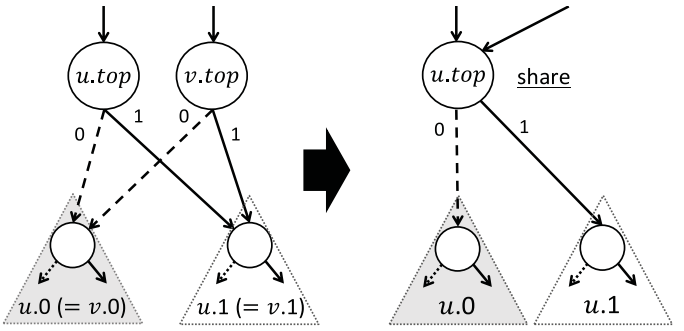
In order to cope with this strictness, we relax the algorithm given by (Denzumi *et al.* 2011): We call the proposed graph structure *Relaxed SeqBDD* and explain it below. Since we cannot expect all examples to occur in reality, we instead look at *False* phrases represented by the graph. The set of False phrases represented by the SeqBDD include the following two types:

(1) Phrases that did not occur.
(2) Phrases that should not occur.

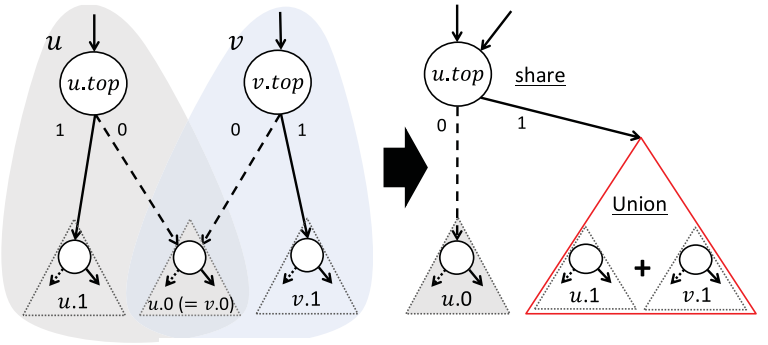
In SeqBDD, the (1) is not represented, whereas the latter is omitted: if represented using a less compact form of original BDD, they are connected to 0-terminal nodes. The issue is that possible phrases that did not appear get included in (2) by the standard process of SeqBDD construction.

The idea of the Relaxed SeqBDD is to relax the algorithm so that all the phrases that share 0-terminal nodes get merged during construction. To realize this, only the node sharing rule of the construction algorithm needs to be modified: The node sharing rule for the SeqBDD is described in Section 3.1, and is intuitively depicted in Figure 4. In the figure of original algorithm (a), one can see that nodes are only shared when *both* their True and False subgraphs are equivalent, which is called *zero suppression reduction rule*, in the study of BDD. In the proposed Relaxed SeqBDD as shown in Figure 4(b), this rule is replaced by the relaxed version that shares the subgraphs having the common subgraph leading to the 0-terminal node. As a means to satisfy this constraint, the True children are combined using the union operator

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press



(a) Original sharing rule



(b) Relaxed sharing rule

Fig. 4. (Colour online) Original and relaxed sharing rules.

proposed in Loekito *et al.* (2010), where its algorithm is provided in Appendix A. The combination process results in the sharing of the structures of both True children, and is illustrated in Figure 4(b); in the figure, the two True subgraphs of u and v are combined using the union operation.

Unlike the original node sharing rule, there is danger of creating a cyclic graph with the proposed method (for example, in Figure 3 right figure, a cyclic graph can get generated if node w_3 was unified with node w_1). As a consequence sharing is not performed if it would result in a cyclic graph. This new rule allows the BDD structure to represent word sequences that have not occurred in the data. Our hypothesis is that this will make the structure more suitable for template extraction, because observations are necessary to make the generalizations leading to templates.

The idea is formally defined as follows as a modification of the original theory of SeqBDD (Denzumi *et al.* 2011). In order to show clearly where the modifications are, we follow a similar descriptive form to the algorithms presented in the article. The modification of the definition is shown mathematically first and then we show which part of the algorithm must be modified.

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

Algorithm 1 *Reduce* : SeqBDD Reduction algorithm

Require: v {acyclic graph}
Ensure: The reduced graph
1: **if** $v = 0$- **or** 1-*terminal node* **then**
2: **return** v
3: **else**
4: **return** *Get_node*($v.top$, *Reduce*($v.0$), *Reduce*($v.1$))
5: **end if**

Algorithm 2 *Get_node* : Original

Require: $v.top, g_0, g_1$
Ensure: Graph processed with node sharing and zero suppression reduction rules
1: **if** $g_1 = 0$-*terminal node* **then**
2: **return** g_0 {zero suppression reduction rule}
3: **end if**
4: $u \leftarrow$ *hash_table*($\langle v.top, g_0, g_1 \rangle$)
5: **if** $u \neq$ *null* **then**
6: **return** u {node sharing rule Section 3.1}
7: **end if**
8: $v.0 \leftarrow g_0$
9: $v.1 \leftarrow g_1$
10: *hash_table*($\langle v.top, g_0, g_1 \rangle$) $\leftarrow v$ {*hash_table* : Hash subgraphs globally}
11: **return** v

Mathematically, a subgraph v is characterized by the tuple $\langle v.top, v.0, v.1 \rangle$, where $v.top$ denotes the top node of the subgraph v, $v.0$ and $v.1$ denote the subgraphs under v, through 1-edge (called True subgraph) and 0-edge (called False subgraph), respectively. In the original (Denzumi *et al.* 2011), the subgraphs u and v are shared in the SeqBDD when:

$$\langle v.top, v.0, v.1 \rangle == \langle u.top, u.0, u.1 \rangle,$$

where the Relaxed SeqBDD shares u and v when:

$$\langle v.top, v.0 \rangle == \langle u.top, u.0 \rangle.$$

Given this simple modification, the algorithm of the *Reduce* procedure shown in Algorithm 1 (which in fact calls Algorithm 2) can be modified as in Algorithm 3.

The sharing of two graphs occurs by providing a subgraph v to the function *Reduce*, as in Algorithm 1. The graph reduction process is performed by recursively calling the *Reduce* function (line 5). The *Reduce* function uses a *Get_node* function that implements the graph sharing and zero suppress reduction rule (see Section 3.1). To realize the Relaxed BDD, we only have to modify this *Get_node* function, where Algorithm 2 shows the original *Get_node* algorithm and Algorithm 3, the proposed algorithm. The differences are shown in red in Algorithm 3.

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

Algorithm 3 *Get_node* : Proposed Relaxed Version

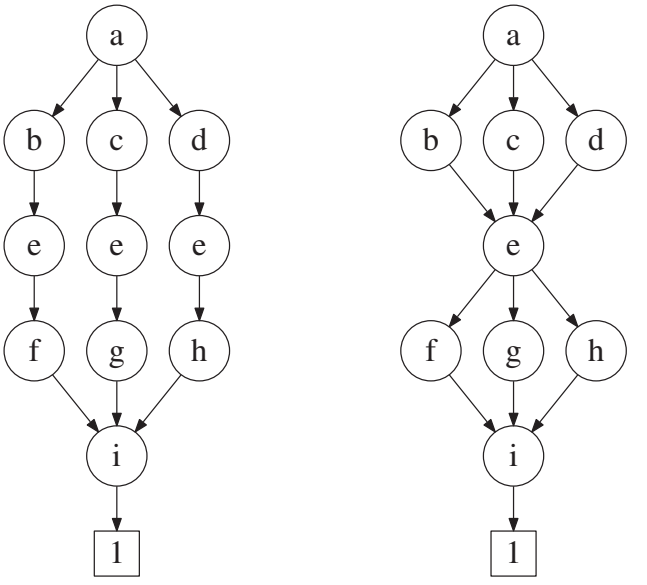
Require: $v.top, g_0, g_1$
Ensure: Graph processed with relaxed node sharing and zero suppression rules
1: **if** $g_1 = 0$-*terminal node* **then**
2: **return** g_0 {zero suppress reduction rule}
3: **end if**
4: $u \leftarrow hash_table(\langle v.top, v.0\rangle)$
5: **if** $u \neq null$ **then**
6: **if** $\neg\exists$ path from $u.top$ to $v.top$ **then**
7: $u.1 \leftarrow Union(g_1, u.1)$
8: **return** u {relaxed node sharing rule Section 4}
9: **end if**
10: **end if**
11: $v.0 \leftarrow g_0$
12: $v.1 \leftarrow g_1$
13: $hash_table(\langle v.top, g_0\rangle) \leftarrow v$ {Note: hash key differs from Algorithm 2}
14: **return** v

The *Get_node* function acquires the top node of the subgraph v, and the two subgraphs that are processed by the *Reduce* function. Until line 4, the two algorithms are the same (for processing the zero suppress reduction rule). Then, the sharing is processed by acquiring the existing common graph as u. The graph is stored in the *hash_table*, that maintains a table of constructed graphs (as pointers), using node and subgraphs as keys. The key point of the difference of the proposed Algorithm 3 from the original Algorithm 2 lies in this *hash_table* function. In the original, the *hash_table* stores a structure given the triplet $\langle v.top, v.0, v.1\rangle$, whereas the modified method is only given the pair $\langle v.top, v.0\rangle$.

Figure 4 (a) and (b) shows the difference of the sharing rules of the original triplet $\langle v.top, v.0, v.1\rangle$ and the relaxed pair $\langle v.top, v.0\rangle$. In the original version of Algorithm 2, if a common part u already exists, then, u is returned; if not g_0 and g_1 are linked as $v.0$ and $v.1$. In the modified version of Algorithm 3, the procedure is the same, but in addition, the cyclic condition is examined (line 6), and if no cycles exists, unification is conducted (line 8). This *Union* function is a predefined function to unify two subgraphs which is provided in Appendix 8. The operator is exactly the same as that defined in Denzumi *et al.* (2011).

In order to show the effect of the Relaxed SeqBDD, Figure 5 illustrates two graphs, a SeqBDD and a Relaxed BDD, generated for the three input sequences $\{abefi, acegi, adehi\}$. In (a), the original SeqBDD is shown, where the e in the middle is not merged. With SeqBDD, to create the figure as in (b), the input must be $\{abefi, abegi, abehi, acegi, acefi, acehi, adegi, adefi, adehi\}$, where all combinations of prefix $\{ab, ac, ad\}$ and suffix $\{fi, gi, hi\}$ should occur. The SeqBDD therefore does not incorporate any unseen but possible input. RelaxedBDD allows the generation of (b), from the three inputs, assuming the possibility of occurrence of six unseen inputs. Of course this is a double-edged sword: the algorithm will certainly decrease

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press



(a) Original SeqBDD (b) Relaxed SeqBDD

Fig. 5. Original and Relaxed SeqBDDs for the input data of $\{abefi, acegi, adehi\}$.

the precision of template extraction of $aXeYi$, but will increase the recall. The effectiveness of this modification can only be verified with real evaluation, which we provide in Sections 6 and 7.

5 Post-processing of a relaxed SeqBDD to obtain templates

Given a set of word sequences, a Relaxed SeqBDD is constructed by using an on-the-fly algorithm, thus explained. Then, we apply the following procedure to obtain the final templates out of the Relaxed SeqBDD.

The appropriate form of input word sequence depends on the problem as will be shown in the following two evaluation sections. In our work, the input word sequences are labeled with POS tags, and the SeqBDD is constructed based on the POS tags (Σ is the set of POS tags). The words are nevertheless preserved in the nodes as auxiliary information, in the form of lists containing the words at nodes that were contracted through node sharing operations during BDD construction. Let $\mathscr{W}_u$ be the set of words at node $u \in V$, $f(w)$ be the relative frequency of $w \in \mathscr{W}_u$.

Fully constructed SeqBDDs are post-processed as follows.

0. **POS-tagging:** To build a relaxed SeqBDD, all word sequences must be POS tagged.
1. **Delete Falsehood:** The 0-edges are simply deleted, since they are no longer necessary.
2. **Weight edges:** The edges are then weighted by the absolute frequency of the number of phrases that goes through the edge. In order to reduce the

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

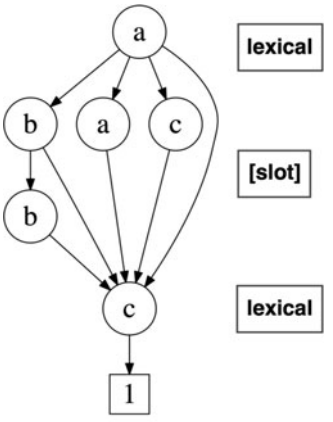


Fig. 6. Post-processed SeqBDD representing: $\{ac, abc, aac, acc, abbc\}$ same as Figure 2.

complexity of the graph, all edges with frequency one are removed (Figure 7(a)).

3. **Obtain paths:** Obtain all *paths* connecting to the starting status of s_{start}.[4] Weight each path with the *minimum* frequency of the edge included in the path (Figure 7(b)).
4. **Find slots:** Each node u along a path is labeled as 'lexical' if $\max_{w \in \mathscr{W}_u} f(w) \geq \theta$, 'slot' otherwise, where θ is set to best suit the problem. This transforms a path into a template with slots (Figure 7(c)).
5. **Merge templates:** Merge identical templates and weight the final template by the sum of the weights of the merged templates (Figure 7(d)).

The output consists of a frequency-ranked list of *templates*. For example, Figure 2(c) is transformed into Figure 6 through procedures **1** and **2** above, where the template '*aXc*' is now readily visible: the output nodes and edges of the real examples are labeled by the weights. Procedure **4** defines which nodes are slots.

The overall procedure above is equivalent to determining the paths by maximal flow (Cormen, Leiserson and Rivest 1990). Since the edges $e \in E$ of the SeqBDD are weighed by frequency, the maximal flow in every path is naturally limited by the minimum frequency edge (Procedure **3**). We believe that this is a natural procedure to extract a template given a SeqBDD graph, but other possible strategies exist and remain future research.

We show how this post-processing proceeds using Figure 7. Let us assume that there are five input phrases[5] :

[4] There are nodes that are not connected to s_{start}, caused by elimination of branches at step 2.

[5] Many NPs in this slot could be longer than this, but here short toy examples are given only for explanation.

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

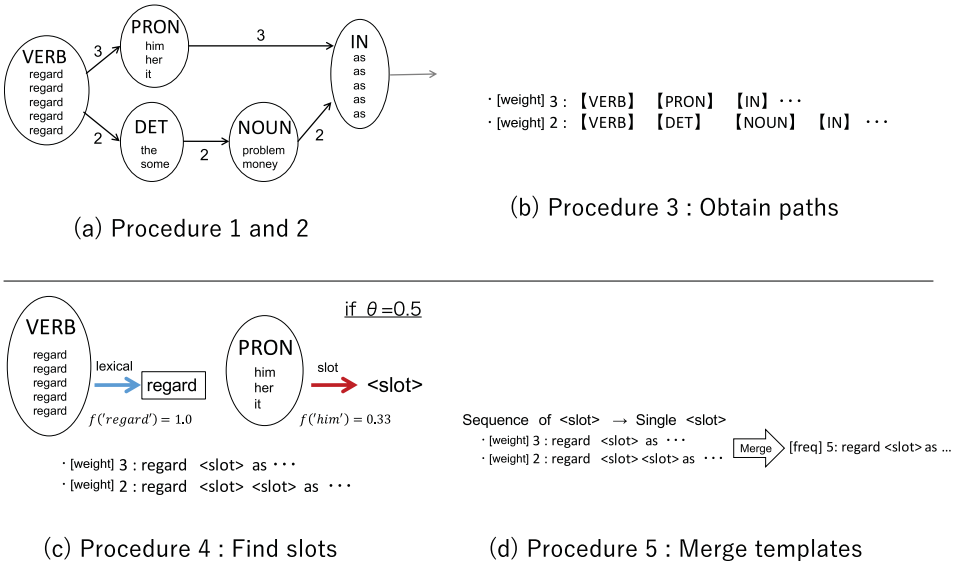


Fig. 7. (Colour online) SeqBDDs post-processing examples.

regard him as ···
regard her as ···
regard it as ···
regard the problem as ···
regard some money as ···

The Relaxed BDD is constructed and by eliminating unnecessary information through procedures **1** and **2**, and the graph would look as that in (a): the graphs are aligned horizontally here (due to space limitations), nodes indicate shared POS, and edges are annotated with frequencies. Examination of paths through procedure **3** gives potential template candidates as in (b). Then lexical/slot nodes are judged with the relative frequency function f as in (c) through procedure **4**. Finally, the equivalent templates are merged to obtain the the final results through **5** as in (d).

6 Verb+preposition template extraction

We first present a large-scale experiment that attempts to extract verb templates out of given sentences. A verb+preposition template here is a template that involves a verb, such as 'regard X as Y'. This test intends to evaluate the performance of the method using a clean dataset, acquired from a dictionary.

For this purpose, the verb+preposition templates were collected by searching Oxford Advanced Learner's Dictionary for phrasal verbs with slots. This extraction was based on the annotation provided by the Dictionary, for example, 'regard X as Y' where X and Y are slots. We considered verbs with multiple slots, only, and excluded verbs that only have one slot. The slots can be located at the end of the template as in 'regard X as Y'. To collect such instances, a verb list was acquired from the British national corpus, we scanned the Learner's dictionary for every verb,

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

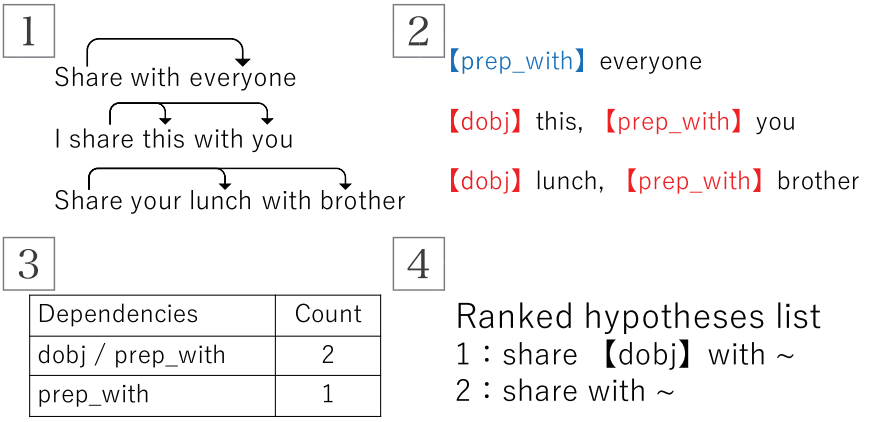


Fig. 8. (Colour online) Extracting parser-based template hypotheses.

and extracted the verb+preposition templates with slots using regular expressions. We obtained a set of 976 templates.

For each template, the sentences containing all words in the template were obtained from the English Gigaword Corpus (fifth edition; LDC2011T07), consisting of 180 million sentences lemmatized and POS tagged using the Stanford CoreNLP tools (Manning *et al.* 2014). The sentences that included the verb were obtained and then filtered using regular expressions so that a span between the slots was at most five words. For example, for the template 'regard X as Y', all sentences with the words 'regard' and 'as' were extracted, under the constraint that the number of words between 'regard' and 'as' was at most five. Similar constraints have often been used in previous work, notably (Baldwin and Kim 2010). The proposed method and the parser-based method that we compare with (defined in the following) were then used to discover the templates from these *search phrases*.

For the proposed method, we extracted ranked lists of template hypotheses from the search phrases, using the method described in Section 5, after the sentences in the search phrases were reduced to sets of subsequences following the main verb. Since there is no established baseline for this task, another plausible method using a parser was created for comparison. This method using a parser is an unfair baseline, since the parser-based method uses vastly more resources, requiring a parser trained on a structurally annotated corpus, whereas our method uses only tagged corpus. In fact, these are two different methods which could even complement each other for the task of phrase structure induction. Hereafter, we call this alternative method to be compared with our method, the parser-based method.

In this parser-based method, first the search phrases for all of the 976 templates being studied were dependency parsed. The parser was the Stanford Parser, based on universal dependencies. The procedure is conducted by providing the whole sentence to the parser. Figure 8(1) shows three such sentences for the template 'share X with Y'. Figure 8(2) shows the dependencies of the verb in the template. In Figure 8(3), template hypotheses for the parser-based method are created from sentences that contain the same dependencies, for example, the second two sentences in the figure

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

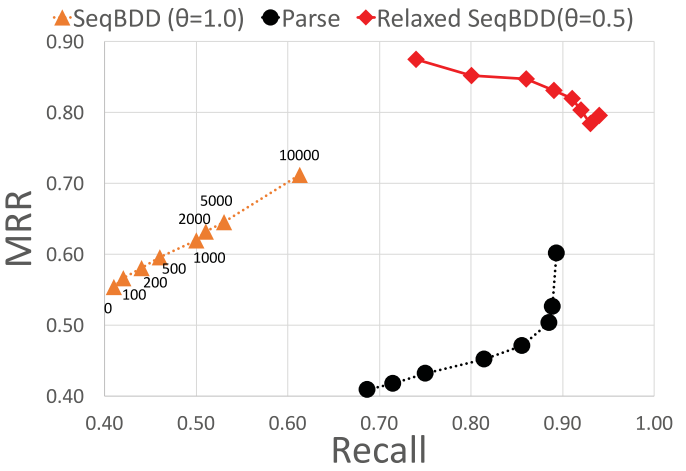


Fig. 9. (Colour online) verb+preposition template extraction from the Gigaword corpus.

both contain the dependencies: `[dobj]` and `[prep_with]` and these give rise to a single template hypothesis with a count of 2. The hypothesis counts reflect the number of sentences that share the same dependencies, and are used to rank the template hypotheses (Figure 8(4)).

The performance was evaluated by mean reciprocal rank (MRR) (Manning, Raghavan and Schütze 2008) and recall, to judge the quality of the lists of extracted template hypotheses. Recall is the proportion of the correct verb template that were extracted. As for what corresponds to the precision, since the output templates are ranked and the rank must be evaluated, we used MRR. MRR is the average of reciprocal rank of the correct verb+preposition template in the output list. Let Q be the number of test cases, for each of which the procedure extracted a template at rank r_i $(i = 1, \ldots, Q)$.

$$MRR = \frac{1}{Q} \sum_{i=1}^{Q} \frac{1}{r_i} \tag{1}$$

When the correct template was not extracted for a set of search phrases, its rank was considered to be ∞. In this experiment, $Q = 976$.

The results are shown in Figure 9. To observe the effect of dataset size, dataplots show the average MRR and recall of templates that had *more than* a specific cutoff threshold number of search phrases. The cutoff values used were: 0, 100, 200, 300, 500, 1 k, 2 k, 5 k, and 10 k. These values are annotated only on the 'SeqBDD($\theta = 1.0$)' orange curve on the graph; on all other curves the cutoff variables run in the same manner.

We first examine the parser-based method and the two variants of the proposed method as follows:

Parse in black dotted line. The parser-based method.
SeqBDD ($\theta = 1.0$) in orange dotted line. (For θ, see Section 5.)
Relaxed SeqBDD ($\theta = 0.5$) in red line with triangles.

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

It can be seen in Figure 9 that the SeqBDD method is able to achieve higher levels of MRR than the parse-based baseline, but at a cost in terms of its recall. The proposed Relaxed SeqBDD variant is able to achieve considerably higher levels of MRR in all cases than both methods without losing ground in terms of recall. It is impressive how higher MRR was acquired: this is due to the relaxed algorithm that integrates template candidates scattered in the original SeqBDD. We observed that for the Relaxed SeqBDD there is a trade-off between MRR and Recall, however, the Relaxed SeqBDD was able nonetheless able to achieve MRR levels greater than the other methods even when the level of recall was at its maximum value of ninety-five per cent. It is clear from the figure that the baseline and SeqBDD methods depend heavily on the amount of data, whereas the proposed method appears to be far less sensitive to the data set size, and is able to achieve respectable MRR and recall on even the smallest subset of the data used in our experiments. Note, that the parser-based method used a parser trained on a structurally annotated corpus, whereas our method used only a tagged corpus.

The end-to-end runtime performance,[6] from raw text until the final output for the proposed SeqBDD method was 72.1 sec and 866.7 MB per template. The Relaxed SeqBDD method required 247.1 sec for processing and 1.5 GB of memory. The baseline was the fastest, taking only 51.8 sec for processing, but required 2.3 GB of memory due to the resources required for parsing.

7 Template extraction from twitter

In this section, we are concerned with a more real-world application: the extraction of templates from short text messaging. Twitter, for example, is used for commercial objectives, and it is relatively common for the information in tweets to be generated using templates, as in the following examples:

5.1 earthquake, Kermadec Islands region. 2016-06-23 07:44:37 UTC at epicenter (24m ago, depth 55km).
5.5 earthquake, Scotia Sea. 2016-06-23 00:05:39 at epicenter (20m ago, depth 10km).
5.1 earthquake, Southern Mid-Atlantic Ridge. 2016-06-21 17:47:36 at epicenter (19m ago, depth 10km).
6.1 earthquake, Northern Mid-Atlantic Ridge. 2016-06-21 13:26:35 at epicenter (31m ago, depth 10km).
5.3 earthquake, South of the Fiji Islands. 2016-06-21 23:40:40 at epicenter (24m ago, depth 565km).

The automatic extraction of such templates will facilitate the analysis and extraction of structured information. The task in this experiment is to output the templates for a given set of tweets. Clearly, this example is output by a machine bot: other more real examples are provided in Appendix B.

For this purpose, we collected corpora of Tweets in English and Japanese.

In the case of English, a set of Twitter accounts was mined starting from a single Twitter account by recursively following the followers of the accounts. A set of tweets was collected for each account traversed. In the case of Japanese, a list of tweet sets for accounts was downloaded directly from the web.[7] Since we are

[6] In this work, we used the Python language on a PC cluster with CPU of Intel E5 3.4GHz and 512GB of RAM running on CentOS.
[7] URL:http://seesaawiki.jp/w/wikkkiiii/

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

interested in evaluating template extraction, we downloaded the ones which were judged to contain templates by the baseline procedure. Extraction from random data is unlikely to be fruitful, since any extraction algorithm will not work unless many similar expressions are contained in the original data. Therefore, we have to filter the dataset first.

The baseline procedure is based on heuristics of substring matching, described as follows. For a set of tweets for a Twitter account, *k*-means clustering (Macqueen 1967) was applied with the similarity between two tweets being the word term frequency multiplied by inverse document frequency (TF-IDF) vector (Deerwester *et al.* 1990). The procedure partitions the set of tweets into K clusters. Templates were constructed for each cluster as follows. First, the common words that appear in more than half of the tweets were identified and preserved as words in the template; then single slots were created to replace the sequences of remaining words. The most frequent template was output for each cluster, leading to K templates being output for each set. Here, we set the value of the K to 20, that presented the highest F-score in our final evaluation. If any of the templates contained one slot between two lexical items, then the list of tweets for the set was said to contain a template. The length of the expression filling the slot could be longer than five words, different from the previous evaluation task of verb+preposition template.

This baseline was used to produce the baseline output, and also to create the gold standard explained as follows. First, using this baseline, we collected a set of 600 tweet sets (one set per Twitter account) of English tweets, and another set of 600 tweet sets of Japanese tweets: for every set, the template suggested by the baseline contained more than one slot, and we excluded a set that had short templates with only one slot. Then, trivial noisy tweets, such as replies and re-tweets together with those with URLs in the tweets were deleted or replaced automatically by using regular expressions. They are able to form (unwanted noise) templates, and they can affect performance, depending on their frequency of occurrence in the tweets.

After cleaning, we constructed two types of gold standard from this pre-processed data. The two annotators[8] manually scanned every tweet set, in which the candidate template is highlighted by the baseline method. First, every Twitter set is annotated with whether it contained a template or not. Second, among those which contained a template, we randomly chose hundred sets and annotated them with the templates; each account was annotated with multiple possible templates, that were found in many tweets included in the account.

This set of hundred sets is defined as *Template-annotated corpora*, and the rest is defined as *Hand-classified corpora*. Note that a set of the former always contains a

[8] One annotator, who is unrelated to this project, was hired specifically for the annotation task. The other annotator is the first author. The two annotators initially worked separately. Given the baseline, the annotators were able to observe the appearance of templates existing in each of the datasets and they worked by modifying the appearance and borders of the templates. The agreement of the two annotators was pretty high, due to the baseline highlights, even though they were often implausible. Then, the results were discussed by the two annotators to reach a complete agreement, and the annotators rectified their errors and omissions. The overall result was checked by the second author. This procedure was conducted before the actual experiments of SeqBDD, without knowing its output.

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

Table 1. *Summary information and statistics for the Twitter data-sets*

Tweets-Set	Template-Annotated	Hand-Classified
Languages and the number of sets	Each 100 English and Japanese	Each 500 English and Japanese (39 English and 23 Japanese do not have templates)
Label	Manually annotated templates (a maximum of 5)	Manually classified whether a template exist or not
The number of tweets range/Ave.	from 478 to 3200 /2578 tweets per sets	from 261 to 3200 /2576 tweets per sets
Experiment	Both Pattern Classification and Extraction	Only Pattern Classification

template, whereas an account of the latter does not necessarily have one. Relatedly, since the baseline procedure is used to collect the data, the recall of the baseline system is 1.0.

Each set of tweets from the template-annotated corpora was annotated with a maximum of five templates, created by an independent human annotator (not one of the authors). For example, the set of tweets in the example above was annotated with the template:

<slot> earthquake , <slot> . <slot> at epicenter (<slot> m ago , depth <slot> km) .

A summary of the template-annotated and hand-classified data is shown in Table 1. Typical sources included news, weather and public information feeds. A set of tweets consisted of all tweets from a single source. The total size of the corpora was over three million tweets.

It can be argued that the dataset is biased in favor of the baseline, since it was created using the baseline. Without use of a tool that gives a hint to human annotators wherein a tweet set contains a template, however, the annotation task was too difficult and time consuming. Since we cannot use the proposed method for this task, we used the baseline, even though the resulting comparison will be in favor of the baseline. However, as will be shown, the proposed method outperforms the baseline even on such data. Therefore, we believe that the evaluation serves to demonstrate the effectiveness of our approach.

We used thus constructed evaluation

Two types of experiment were performed. The first used all the 600 data both in English and Japanese. We call this experiment the *Pattern Classification Experiment*. The task is a straightforward classification task to identify whether or not templates exist in the tweets, and we therefore used the standard evaluation method of precision/recall/f-score to evaluate the performance. The results are shown in Table 2.

The precision of the baseline is 0.96 for Japanese and 0.93 for English, indicating that the heuristic used to extract the candidate template data was of high quality. Nonetheless, applying the proposed Relaxed SeqBDD method to this data was able

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

Table 2. *Template extraction results for all the 600 set of data*

		Baseline	SeqBDD	Relaxed SeqBDD
Japanese	P	0.964	0.978	0.983
	R	1	0.816	0.985
	F	0.982	0.889	0.984
English	P	0.935	0.954	0.975
	R	1	0.836	0.978
	F	0.966	0.891	0.976

to give respectable gains in both precision and f-score at a small cost in terms of recall. It is also clear from the table that Relaxed BDD is superior to the method based on the original SeqBDD in all respects.

The second experiment evaluated the quality of the templates that were actually extracted, using the Template-annotated corpora (hundred each in English and Japanese). We call this experiment the *Pattern Extraction Experiment*. Since the baseline outputs K templates, only the best K templates were considered also for SeqBDD methods. Similar to the previous work in Figure 9, these ranked lists were evaluated by MRR of the best ranked correct template appeared in the output list of length K. Here, in formula (1), $Q = 200$. If there are multiple correct templates extracted, the reciprocal of the best rank is used to obtain the average, and if no correct template was found, then the rank is set as ∞.

The results are shown in Figure 10, English and Japanese altogether. There are three lines: black, orange dashed and red solid, representing the baseline, SeqBDD and Relaxed SeqBDD, respectively. Each line shows the values of MRR and R, while varying K. In all cases, the recall of the Relaxed SeqBDD was considerably higher that the SeqBDD method, this was expected and was caused directly by the relaxation of the algorithm allowing it to be applicable to more cases in the data. There is of course a trade-off with MRR, although this trade-off proved to not to be too detrimental, and in fact for all values of K greater than 3, the Relaxed SeqBDD achieves higher levels of both MRR and recall than the other methods.

The end-to-end runtime performance, from raw text until the final output for the proposed SeqBDD method was 37.3 sec using 836 MB of memory for a single tweet-set, the Relaxed SeqBDD method took 210.5 sec and used 1.4 GB, whereas the baseline took 30.9 sec and used 2.3 GB.

Some example extraction results for the Twitter data are shown in the Appendix B. The patterns are successfully acquired for Twitter examples like B.1 or B.2, where the pattern frame structure is solid, and the length of the expression filling the slot is short.

On the other hand, the algorithm fails to extract the correct patterns when the length of expression filling the slot is large, such as in the example of B.4, due to the fact that the Sequence BDD graph becomes too sparse. For example, patterns such as ‘How to * (URL)’ or ‘Outsourcing Human Resouces * (URL)’ could have very long expression in *. Moreover, if the pre-processing of POS tag fails, then all that

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

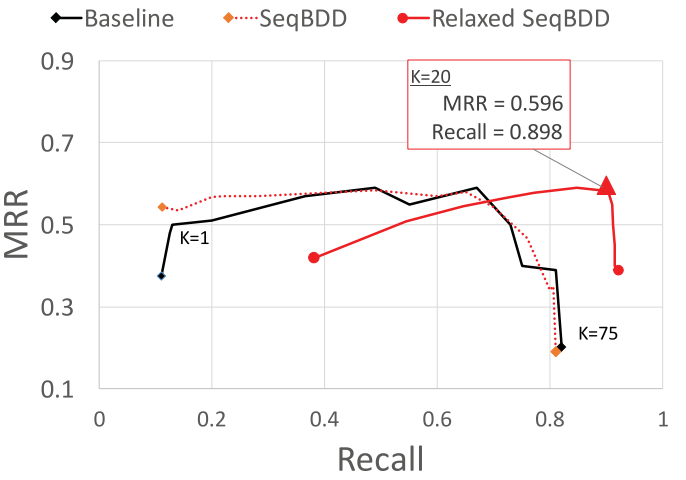


Fig. 10. (Colour online) Extraction of templates from twitter using template-annotated corpora.

follows also fails, as shown in the example of B.3. The development of methods for overcoming these issues remains future work.

8 Conclusion

This paper presents a novel method, based on an original relaxed algorithm for building SeqBDDs, for inducing a template—a subsequent phrase with slots—, given a set of phrases having common structure. The SeqBDD whose construction process we relax, is equivalent to a minDFA but more efficient and well-suited for template extraction. We proposed a relaxed form of the SeqBDD as a means to represent sets of word sequences graphically in a compressed form that naturally discovers structures in the text. The proposed enhancement to the SeqBDD construction process, relaxes the node sharing rule to facilitate a greater degree of node sharing. The structures inherent in the SeqBDD constructed from the text are then extracted to produce templates.

We evaluated the proposed relaxed method on two template extraction tasks: verb+preposition templates, and tweet templates. In verb+preposition template extraction, the original SeqBDD was able to achieve considerably higher MRR than a parser-based method, but had lower recall. When relaxing the SeqBDD, the method was able to substantially exceed the performance of parser-based method both in terms of MRR and recall. In the more real-world task of tweet template extraction using more than three million tweets, our experiments show that the proposed relaxed SeqBDD method was able to outperform the original SeqBDD by substantially increasing recall while at the same time maintaining high levels of precision.

This work represents only a first step on the road to inducing templates. This paper has shown empirically that relaxing the SeqBDD to coalesce very similar, but slightly different, structures greatly enhanced its utility for template extraction. We

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

believe further study along these lines would be fruitful in the future. In addition, we intend to focus on improving the efficiency of the algorithm in future work to allow it to scale for use on large datasets. This requires a further approximation of the method, and should be evaluated from the point of view of both accuracy and speed.

In the future, our work could be linguistically extended from the viewpoint of the Pattern Grammar, our original motivation; the current work was limited by being simplified in order to be realized as a computational procedure. Since the Pattern Grammar has been supplemented recently by work on Construction Grammars from Cognitive Linguistics, one possible future path might be to reformulate the output templates in relation to more formal linguistic notions.

Finally, as for the downstream applications of our method, an implementation of KWIC that produces templates would be helpful for lexicographers and corpus linguists. Moreover, we believe that a module for social network system (SNS) and blog data analysis would prove useful from industrial perspective.

References

Abney, S., and Abney, S. P. 1991. Parsing by chunks. In *Principle-Based Parsing*, pp. 257–278. Netherlands: Kluwer Academic Publishers.

Baldwin, T., Cook, P., Lui, M., Mackinlay, A., and Wang, L. 2013. How noisy social media text, how diffrnt social media sources. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP-2013)*. Mark: John Blitzer.

Baldwin, T. and Kim, S. N. 2010. Multiword expressions. In N. Indurkhya, and F. J. Damerau (eds.), *Handbook of Natural Language Processing*, 3rd ed. Boca Raton, FL: CRC Press, Taylor and Francis Group.

Bryant, R. E. 1986. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers* **35**(8): 677–691.

Cormen, T. H., Leiserson, C. E., and Rivest, R. L. 1990. *Introduction to Algorithms*. USA: MIT Press.

Cui, H., Kan, M.-Y., and Chua, T.-S. 2004. Unsupervised learning of soft patterns for generating definitions from online news. In *Proceedings of the 13th International Conference on World Wide Web (WWW-2004)*, 90–99.

Daciuk, J., Mihov, S., Watson, B. W., and Watson, R. E. 2000. Incremental construction of minimal acyclic finite state automata. *Computational Linguistics: Special Issue on Finite-State Methods in NLP* **26**: 3–16.

de Medeiros Caseli, H., Ramisch, C., das Graças Volpe Nunes, M., and Villavicencio, A. 2010. Alignment-based extraction of multiword expressions. *Language Resources and Evaluation: Special Issue on Multiword Expression: Hard Going or Plain Sailing* **44**(1–2): 59–77.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* **41**(6): 391–407.

Denzumi, S., Yoshinaka, R., Arimura, H., and Minato, S.-I. 2016. Sequence binary decision diagram: Minimization, relationship to acyclic automata, and complexities of boolean set operations. *Discrete Applied Mathematics* **212**(C): 61–80. Elsevier Sicence Publishers B. V. Amsterdam, The Netherlands, The Netherlands.

Denzumi, S., Yoshinaka, R., Minato, S.-I., and Arimura, H. 2011. Efficient algorithms on sequence binary decision diagrams for manipulating sets of strings. Technical Report, DCS, Hokkaido U., TCS-TR-A-11-53.

Duan, J., Lu, R., Wu, W., Hu, Y., and Tian, Y. 2006. A bio-inspired approach for multi-word expression extraction. In *Proceedings of the COLING/ACL on Main Conference Poster*

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

Sessions (COLING-ACL-2006). Association for Computational Linguistics, Stroudsburg, PA, USA, 176–82.

Fazly, A. 2006. Automatically constructing a lexicon of verb phrase idiomatic combinations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, 337–344.

Fernau, H. 2009. Algorithms for learning regular expressions from positive data. *Journal of Information and Computation* **207**: 521–541.

Francis, G., Hunston, S., and Manning, E. 1996. *Collins COBUILD Grammar Patterns 1: Verbs*. London: Harper Collins.

Francis, G., Hunston, S., and Manning, E. 1998. *Collins COBUILD Grammar Patterns 2: Nouns and Adjectives*. London: HarperCollins.

Gimpel, K. and Smith, N. A. 2012. Concavity and initialization for unsupervised dependency parsing. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 577–581.

Han, J., Pei, J., and Yin, Y. 2000. Mining frequent patterns without candidate generation. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD-2000)*. ACM, 1–12.

Headden, III, W. P., Johnson, M., and McClosky, D. 2009. Improving unsupervised dependency parsing with richer contexts and smoothing. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-2009)*. Association for Computational Linguistics, 101–109.

Huang, Z., Xu, W., and Yu, K. 2015. Bidirectional LSTM-CRF models for sequence tagging. CoRR abs/1508.01991.

Hunston, S., and Francis, G. 2000. *Pattern Grammar: A Corpus-driven Approach to the Lexical Grammar of English*. Studies in Corpus Linguistics 4. John Benjamins Publishing Company.

Kim, C. 2010. Text: Automatic template extraction from heterogeneous web pages. *IEEE Computer Society*, 23 (4): 612–626.

Kita, K., Kato, Y., Omoto, T., and Yano, Y. 1994. A comparative study of automatic extraction of collocations from corpora: mutual information vs. cost criteria. *Journal of Natural Language Processing 1,* 1: 21–33.

Klein, D. and Manning, C. D. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*.

Knuth, D. 2009. *The Art of Computer Programming*, vol. 4, Fascicle 1. Addison-Wesley.

Loekito, E., Bailey, J., and Pei, J. 2010. A binary decision diagram based approach for mining frequent subsequences. *Knowledge and Information Systems* **24**(2): 235–268.

Macqueen, J. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, 281–297.

Manning, C. D., Raghavan, P., and Schütze, H. 2008. *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of the Association for Computational Linguistics System Demonstrations*, 55–60.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics* **19**(2): 313–330.

Martens, S. 2010. Varro: an algorithm and toolkit for regular structure discovery in treebanks. In *Proceedings of the Coling 2010: Posters*. Coling 2010 Organizing Committee, Beijing, China, 810–8.

István Nagy T., and Vincze, V. 2014. Vpctagger: Detecting verb-particle constructions with syntax-based methods. In *Proceedings of the 10th Workshop on Multiword*

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

Expressions (MWE). Association for Computational Linguistics, Gothenburg, Sweden, 17–25.

Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., and Hsu, M. 2001. Prefixspan: mining sequential patterns by prefix-projected growth. In *Proceedings of the 17th International Conference on Data Engineering*. IEEE Computer Society, 215–224.

Sag, I. A., Baldwin, T., Bond, F., Copestake, A., and Flickinger, D. 2002. Multiword expressions: a pain in the neck for NLP. In *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 1–15.

Sangati, F., and van Cranenburgh, A. 2015. Multiword expression identification with recurring tree fragments and association measures. In *Proceedings of the MWE@ NAACL-HLT*, 10–18.

Smadja, F. 1993. Retrieving collocations from text: Xtract. *Computational Linguistics* **19**(1): 143–177.

Tu, Y., and Roth, D. 2012. Sorting out the most confusing english phrasal verbs. In *Proceedings of the 1st Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval-2012)*. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 65–9.

Zarrieß, S. and Kuhn, J. 2009. Exploiting translational correspondences for pattern-independent MWE identification. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications (MWE-2009)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 23–30.

Appendix A. Union operation of SeqBDD

This section explains the Union operator as proposed in (Loekito *et al.* 2010). This Union operator is used in line 7 of Algorithm 3.

Given two SeqBDDs P, Q, The operation to produce $P \cup Q$ is shown in Algorithm 4.

First, if either of P and Q are 0-terminal nodes, then the one which is not the 0-terminal node is output. If P and Q are equivalent, then P is output.

In other cases, the top nodes of the two graphs are compared. If they are the same, then the new top is the $P.top$ and the true/false subgraphs are constructed by merging P and Q's true/false subgraphs, respectively, where the merging is conducted using the Union operator.

Suppose that $P.top$ is alphabetically smaller than $Q.top$, then $P.top$ is smaller than any nodes represented in Q. Therefore, the new top is assigned to be $P.top$, and P's true subgraph ends with the 1-terminal node. P's false subgraph must be merged with Q where this operation is conducted recursively using the Union operator.

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

Algorithm 4 *Union*

Require: P, Q :SeqBDD
Ensure: $P \cup Q$:SeqBDD
1: **if** $P = 0$-*terminal node* **then**
2: return Q
3: **else if** $Q = 0$-*terminal node* **then**
4: return P
5: **else if** $P = Q$ **then**
6: return P
7: **else if** $(w \leftarrow hash_table(\langle P, Q \rangle)\ exists)$ **then**
8: return w
9: **end if**
10: {Compare top node's label using alphabetical order}
11: **if** $P.top = Q.top$ **then**
12: $w \leftarrow Get_node(P.top,\ P.0 \cup Q.0,\ P.1 \cup Q.1)$
13: **else if** $P.top < Q.top$ **then**
14: $w \leftarrow Get_node(P.top,\ P.0 \cup Q,\ P.1 \cup 0\text{-}terminal\ node)$
15: **else if** $P.top > Q.top$ **then**
16: $w \leftarrow Get_node(Q.top,\ P \cup Q.0,\ 0\text{-}terminal\ node \cup Q.1)$
17: **end if**
18: $hash_table(\langle P, Q \rangle) \leftarrow w$
19: **return** w

Appendix B. Twitter pattern extraction example

The following subsections provides four examples for the Twitter evaluation, good/bad and English/Japanese.

A slot of a template is indicated by a POS sequence in parenthesis and the number in parenthesis [] at the end of the line shows the usage frequency of the template.

B1. Good example: English

Input:@weatherchannel (total 3,227 tweets)

- #Severe thunderstorm watch for portions of western ND and northwestern SD is in effect until 3 a.m. CDT. #NDwx #SDwx https://t.co/qOoPdBwWHN
- Florida's transition to the wet season is kicking into gear this week: https://t.co/zXCGhxHkoR https://t.co/qodQw0WFqt
- In honor of #MelanomaMonday & @OutruntheSunInc, stay safe w/these facts about #melanoma. https://t.co/lNnZzk70C9 https://t.co/BfZW6YuciY
- Line of #severe t-storms w/ high winds, perhaps brief #tornado, pushing into #Houston metro. https://t.co/EU10HpTSOK https://t.co/Vm1bj5X5Xk
- New #severe t-storm watch until 11a CT includes #Houston. Damaging winds main threat. https://t.co/rUFHq68e8h

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

• Rinse...repeat. Yet another setup for local flash #flooding in #Texas this week. https://t.co/iqPzZhX386 #txwx https://t.co/8eYPftAZ5e
• Science is about to make cooking a whole lot easier. From chopping to cleaning up, this robot has dinner covered. https://t.co/S0eLUABpa4
• Severe T-Storm Watch for portions of Southwest Texas until 10 p.m. CDT. #TXwx https://t.co/Zizty3vfRH
• Tornado Warning for Aitkin, Itasca and St. Louis Counties in MN until 7:30 PM CDT https://t.co/TzipGRIIa5
• Tornado Warning for Baltimore and Howard Counties in MD until 2:00 PM EDT https://t.co/GkZEDFI9ga
• Tornado Warning for Berkeley and Morgan Counties in WV until 5:00 PM EDT https://t.co/ufrUraPlIP
• Tornado Warning for Cheyenne County in CO until 2:45 PM MDT https://t.co/cW9FalG2TX
• Tornado Warning for Clay, Jasper and Richland Counties in IL until 8:30 PM CDT https://t.co/f4UT9Vemul
• Tornado Warning for Fannin and Lamar Counties in TX until 6:45 PM CDT https://t.co/B6MEBnVNC6
• Tornado Warning for Harrison and Marion Counties in TX until 1:45 AM CDT https://t.co/fFHwLqXQ8n
• Tornado Warning for Ochiltree and Roberts Counties in TX until 8:45 PM CDT https://t.co/86k9zvmjkY
• Tornado Warning for Tillman County in OK until 2:00 AM CDT https://t.co/aC8vnA4PUk
• We are tracking more #snow in the West & a midweek #severe threat for the #Plains...get latest forecast NOW on @AMHQ https://t.co/qUB7LQivQE

Output:@weatherchannel (total 98 patterns)

1: Tornado Warning for 【NNP】 County in 【NNP】 until 【CD】 PM CDT (URL) [440]
2: Tornado Warning for 【NNP】 , 【NNP】 , 【NNP】 and 【NNP】 Counties in 【NNP】 until 【CD】 PM CDT (URL) [36]
3: Tornado Warning for 【NNP】 County County in 【NNP】 until 【CD】 PM CDT (URL) [33]
4: 【VBN】 [9]
5: Tornado Warning for 【NNP】 County and 【NNP】 Counties in 【NNP】 until 【CD】 PM CDT (URL) [9]
6: IN 【DT】 【NN】 [9]

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

7:【CD】【NNS】[8]
8: Severe weather threat [7]
9: LIVE NOW [7]
10: Tornado of # [6]
11: Tornado【NN】【VBZ】[6]

B2. Good example: Japanese

Input:@Fukkocho_JAPAN (total 407 tweets)

• 「復興局からの復興だより」を更新しました。(福島県) https://t.co/hyNgANUo2a
• 2月14日、髙木復興大臣は宮城県石巻市及び南三陸町を訪問し、第2回東北観光アドバイザー会議を開催しました。 https://t.co/8NzHXQFGMW https://t.co/5Gs9KmL5D0
• 7月18日、小泉復興大臣政務官がイトーヨーカドー・アリオ西新井店で行われた「第12弾 東北かけはしプロジェクト 東北の"今"を見よう!! PRイベント」に出席しました。http://t.co/6jpTR2Bl7w
• 気仙沼支所だより「つちおと」(第 42 号)を発行しました。https://t.co/26dGQhB5zv
• 「東日本大震災に係る政府の対応」として、発災直後の対応と現在の取組みを示しました。あわせて、関係組織等についても紹介しています。https://t.co/ZCSCcm2OJJ
• 「東日本大震災5周年復興フォーラム~新たなステージ 復興・創生へ~」の申込みの受付を開始しました。 https://t.co/EtlZK6Dgy6
• つちおと情報館(福島県)の掲載情報を追加・更新しました。http://t.co/uAOdhsLwzr
• 5月19日、竹下復興大臣の記者会見が行われました。http://t.co/pGjIbHOp4t
• 9月8日、竹下復興大臣の記者会見が行われました。http://t.co/7nZ7zbjACw
• 被災者支援(健康・生活支援)総合対策を公表しました。http://t.co/0hPMr70kBL
• Twitterイベント 東北版「#観光地ウィーク」を開催します! 2月22日から26日までの5日間、東北のおすすめ観光スポットや実際に訪れての感想等をツイートいただくことで参加いただけます。是非、ご参加ください! https://t.co/BZHswiUxMU
• 1月20日、竹下復興大臣の記者会見が行われました。http://t.co/HjuYtvlq4T
• 9月15日、竹下復興大臣の記者会見が行われました。http://t.co/Ez8BXOTIcI
• 竹下復興大臣が「東京銀座・いきなり市場2015」を訪問しました。(2月18日) http://t.co/9SgCvymPbK http://t.co/aaaHAx7wD1

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

• 「復興局からの復興だより」を更新しました。（福島県）https://t.co/us7gj4Fu79
• 平成27年度税制改正の大綱概要について http://t.co/81a0MFrlrm
• 気仙沼支所だより「つちおと」（第43号）を発行しました。https://t.co/26dGQhB5zv
•「復興局からの復興だより」を更新しました。http://t.co/hyNgANUo2a
• 生活支援相談員等の実績が活かされる仕組みについて公表しました。社会福祉士国家試験の受験資格に必要な実務経験へ参入します。http://t.co/IYxuRtB7AL
• 平成２８年３月２７日（日）、福島県福島市で「原子力災害からの福島復興再生協議会」を開催し、髙木復興大臣及び長島復興副大臣、若松復興副大臣が出席しました。https://t.co/dF1YaseGDe
•「企業連携プロジェクト支援事業」により商品化した日本酒「親子の二十歳酒」の販売開始および会津若松市の新成人への無料配布について http://t.co/nMgDJESyva
• 9月1日、竹下復興大臣の記者会見が行われました。http://t.co/vVKcU0yBAL
• 復興庁企業連携メールマガジンＶｏｌ.154を発行しました。（11月16日発行）https://t.co/E8YFM9uSXB
• 気仙沼支所だより「つちおと」（最新号）を掲載しました。http://t.co/26dGQhB5zv
• 2月3日、竹下復興大臣の記者会見が行われました。http://t.co/N7YLtsk28H
• 平成28年４月１日(金)、髙木復興大臣の記者会見が行われました。https://t.co/l8PMLEWWRx
•「新しい東北」復興ビジネスコンテストの受賞団体に対する短期研修（経済同友会協力）が実施されます。http://t.co/gx6TBKGF20
•「新しい東北」先導モデル事業事例集を発刊しました。http://t.co/tjcQjt1C5Z http://t.co/Ijomw2xOvX
•「住まいの復興工程表」を平成27年3月末現在から9月末現在に更新しました。この工程表は、災害公営住宅及び防災集団移転促進事業等の面整備事業で供給される 民間住宅等用宅地の整備に係る工程表をとりまとめたものです。https://t.co/DGUfMY9WxA
•「被災地域企業新事業ハンズオン支援事業」支援案件の募集を開始しました。※本事業は、被災地域における新産業の創出につながる新たな事業を支援する事業です。http://t.co/7M0ETT4SaQ
• 竹下復興大臣が「宮城牡蠣の家」オープニングセレモニーに出席しました。（1月8日） http://t.co/xCCVtrs6lA

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

Output:@Fukkocho_JAPAN (total 21 patterns)

1: 復興 庁 企業 連携 メール マガジン Ｖｏｌ．【数名詞】を 発行 し ました 。（ 【数名詞】月 【数名詞】日 発行 ）(URL) [13]
2: つ ち おと 情報 館（ 【固有名詞】県 ）の 掲載 情報 を 追加・更新 し ました 。(URL [11]
3: つ ち おと 情報 館（ 【固有名詞】県 ）を 更新 し ました 。(URL) [7]
4:【副詞可能名詞】【数名詞】日 、 髙木 復興 大臣 の 記者 会見 が 行われ ました 。(URL) [7]
5:【副詞可能名詞】【数名詞】日 に 髙木 復興 [4]
6:【副詞可能名詞】【数名詞】【数名詞】日 【副詞可能名詞】【数名詞】【数名詞】日（ 【一般名詞】）、 髙木 復興 大臣 は【固有名詞】県 を 訪問 し ました 。(URL) (URL) [3]
7:【一般名詞】【一般名詞】調査 速報 版（ 【固有名詞】【接尾名詞】[3]
8:「 復興 局 支援 コーディネート [3]
9:【副詞可能名詞】【数名詞】【数名詞】日 、 竹下 県 [3]
10:【一般名詞】【サ変接続名詞】[3]

B3. Bad example (incorrect template obtained): English

Input:@virtualemp

• 3 Computing Technologies That Will Change the Face of Business World – http://t.co/f2KeWQbHRc
• 5 project management hacks to make your project see the light of day – http://t.co/TgpWAFVHon
• 5 Secrets to Increase #Employee Engagement With Technology https://t.co/Jq8Y0q6iWG #retention #work #business https://t.co/gN110xSUk4
• 6 Consumer Psychology Hacks You Need to Know NOW! - Neuromarketing https://t.co/uFXBkUoSCe #socialmedia #marketing #smallbiz
• A freelancer can be a handy resource when the work you have is short-term and low-budget http://t.co/g9aYxXLh6U
• Apart from cost effectiveness, there are certain pertinent advantages of #outsourcing.
• Facebook Acquires Online Shopping Search Engine TheFind – http://t.co/56GUwUyVh7
• Google's SDKs Enable iOS, Android Mobile Apps to Work Offline – http://t.co/F2BvkuvNfs
• Hire Full Time Remote C++ Programmers To Work Dedicatedly For You: http://t.co/WGFQIeSzOB
• Hiring remote Photoshop experts is the best way to get customized solutions since these are your very own resources. http://t.co/yyKsIcObJ4

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

• How To Manage Your Remote Development Team https://t.co/omAllH2DWd #business #development #management https://t.co/iqxovtPBKv
• How to measure the success of your digital marketing campaign. Find out at http://t.co/aQAWHndmEL
• How to Merge your Business Processes with Online Marketing – http://t.co/ifIqwGcGES
• How to navigate the dynamic digital marketing world https://t.co/j4FuDd0B12 #marketing #socialmedia #smallbiz
• http://t.co/Rqf6R7vZGH Benefits of hiring dedicated Blackberry application developers with http://t.co/SiomE2CdQk
• Infosys shares surge 9% on founder Narayana Murthy's return http://t.co/KdJrzW8KdB
• Let's check out the facts why a remote dedicated payroll expert is a better resource than a freelancer. http://t.co/f1LMM5vBIa
• Microsoft and Amazon leading cloud computing performance: Edmund Shing- https://t.co/vUWrVR7OjV
• Must add tools for your #startup #enterprenuer https://t.co/FauI57Abg7
• Outsourcing Helps You Recruit Professional Lawyers https://t.co/T7n8jW2L63 #Outsource #Lawyers #smallbusiness
• Outsourcing Human Resources: The Best Business Decision You Will Ever Make. Read on – http://t.co/NCHkgYjpxQ
• Outsourcing Human Resources: The Best Employment Decision You Will Ever Make. Find out why this is the case. Click http://t.co/Ugv8izBZeQ
• Philippines' outsourcing dream crashes – unable to meet talent demand http://t.co/YEIWXXPwR7 #outsourcingtrends
• Precise information about Dot Net team model of http://t.co/9z6sNop6jp, have a look! http://t.co/ChW5TtIaQ2
• Procter & Gamble is doing a major shift in marketing as over 100 brands will be taken off the shelves: http://t.co/W9gsSRzcsx
• The benefits of hiring a remotely working dedicated interior designer: http://t.co/0CeJKwfDSL
• Twitter Inks Deal with Ad Giant WPP to Expand Data-Driven Marketing: http://t.co/NcB4ZK3Whq
• VE Opens Its Third Office – in Birmingham, UK http://t.co/9C8VtWVHbI #whatsnewatve
• Virtual Employee Pvt Ltd, an innovative outsourcing platform has been awarded ISO 9001:2008 Certification. http://t.co/sC9XShsLkU
• Why Is a Dedicated Remote Resource the Best Way to Outsource Your Financial Services?: http://t.co/4XMDIqVROC

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

• Why is India the best outsourcing destination for website development? http://t.co/Lsgit0QMP1

Output:@virtualemp

1:【VBG】to [11]
2:【NNP】, [11]
3: The【JJ】【NN】【NN】[10]
4:【NNP】【VBZ】【JJ】[9]
5: Did [9]
6:【NNS】new [8]
7: How【PRP】【MD】【VB】[8]
8:【NNP】【VBZ】【NNP】[8]
9:【CD】【JJ】【NNS】to【VB】[7]
10:【VBP】you【VBG】[7]

B4. Bad example (incorrect template obtained): Japanese

Input:@sony_jpn (total 1,773 tweets)

• h.ear × WALKMAN OKAMOTO'S コラボモデル 登場！ https://t.co/4JwhrCjACI https://t.co/viaPyTV0wd
• My Sony では、11月のプレゼントキャンペーンを実施中です！ まだ、My Sony ID を登録していない方は、新規登録(無料) ですぐに応募できます。 https://t.co/FLSRXzTvZv https://t.co/HIAUEYwVMA
• Twitter や Ustream など新しいコミュニケーションが増えています。未来のコミュニケーションはどんな風に変わっていくのでしょう？ http://ckpt.jp/YrTb8v
• Xperia スマートフォンをお持ちの方を対象に抽選で 1,000 名様にソニーポイント 2,000 ポイント、または、250 組 500 名様に「アメイジング・スパイダーマン 2 」劇場鑑賞券が当たります！ http://t.co/hhZgXrNG3e http://t.co/nKcaLkfh9L
• α 6300 プロカメラマン インプレッション https://t.co/6G7LHx7uK7 https://t.co/Ro4ICXhJsH
• 「極限の目撃者」 好奇心を呼び覚ます、圧倒的な映像の力。確かな技術に裏打ちされたソニーのビデオカメラが生み出す、心揺さぶる映像の力であなたも「極限」を目撃してみませんか？ http://t.co/J9akne8mwl http://t.co/7djExrhnSe
• 【新商品のお知らせ】昨日発表の Handycam(R) 『FDR-AX1』は、4K の撮影に対応。民生用機器として最高画質である 4K/60p の高画質な映像を撮影し、楽しむことができます。 http://t.co/6meuIs0pSt http://t.co/ziFblJ43bK

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press

• 【新商品のお知らせ】現行機比約 1/4 に小型化した 4K ハンディカムを発売。大型で高感度な 1.0 型イメージセンサーを搭載し、暗所での高画質撮影も実現しました。http://t.co/euh50sqXKO http://t.co/PMRkNqqmr8
• こちらの写真は、α cafe の「水底の歌」さんの作品（α 77 で撮影）。雲ひとつない青空とエメラルドグリーンの池が、さわやかな気分にしてくれますね！ それでは、良い週末をお過ごしください。http://t.co/0Bq7X5S9Nz http://t.co/wSW35BqgUb
• まるで映像に入り込んだような没入感を体感できるヘッドマウントディスプレイ『HMZ-T2』と人気ゲームタイトル『グランツーリスモ5』の開発者によるスペシャルトークコンテンツを本日公開！ http://t.co/rNjjAwvHVa http://t.co/nb9zYJhSY5
• チャットモンチーと Koji Nakajima（ナカコー）が、ハイレゾ対応ワイヤレススピーカーを体験！ http://t.co/aAZBLuhNMm http://t.co/kfy8cyyQCc
• ヘッドホン MDR-1A で若く才能溢れるアーティストの夢や好奇心をサポート http://t.co/vmwnE4u8DG http://t.co/MbK6McG6OP
• 伝統と色気をマセラティ グラントゥーリズモ S に学んでみませんか? http://bit.ly/hRzdOs #htkt_sports
• 壁際に置くだけでリビングの壁に 147 インチの映像を投写する 4K 超短焦点プロジェクターを発売 http://t.co/8atCorKDlN http://t.co/QiZaXKrqwh
• 指紋でも手相でもない。あなたの手のひらにあるあなただけのもの http://bit.ly/aHdY3Z
• 泊りでの出張のときに、いつも悩むのが充電。PC と携帯電話はもちろん、移動中に楽しむ“ウォークマン”のものまで、バッグの中は AC アダプターだらけになってしまいます。みなさんは、どう解決していますか? http://ckpt.jp/aZO4Tv

Output:sony_jpn (total 168 patterns)

1:【副詞可能名詞】、【一般名詞】[17]
2:【括弧開記号】新 商品 の お知らせ 【一般名詞】[15]
3:【形容動詞語幹名詞】な [9]
4:【一般名詞】【サ変接続名詞】【一般名詞】【数名詞】[8]
5: ソニー の【一般名詞】【一般名詞】【サ変接続名詞】[7]
6:【副詞可能名詞】から [7]
7:【自立動詞】[7]
8: ソニー の【一般名詞】が [7]
9: ソニー の【サ変接続名詞】[7]
10:【サ変接続名詞】し [7]

https://doi.org/10.1017/S1351324918000268 Published online by Cambridge University Press