Construction of an English Dependency Corpus incorporating Compound Function Words

Akihiko Kato, Hiroyuki Shindo, Yuji Matsumoto
Nara Institute Science of Technology
8916-5 Takayama, Ikoma, Nara 630-0192, Japan
kato.akihiko.ju6@is.naist.jp, shindo@is.naist.jp, matsu@is.naist.jp

Abstract

The recognition of multiword expressions (MWEs) in a sentence is important for such linguistic analyses as syntactic and semantic parsing, because it is known that combining an MWE into a single token improves accuracy for various NLP tasks, such as dependency parsing and constituency parsing. However, MWEs are not annotated in Penn Treebank. Furthermore, when converting word-based dependency to MWE-aware dependency directly, one could combine nodes in an MWE into a single node. Nevertheless, this method often leads to the following problem: A node derived from an MWE could have multiple heads and the whole dependency structure including MWE might be cyclic. Therefore we converted a phrase structure to a dependency structure after establishing an MWE as a single subtree. This approach can avoid an occurrence of multiple heads and/or cycles. In this way, we constructed an English dependency corpus taking into account compound function words, which are one type of MWEs that serve as functional expressions. In addition, we report experimental results of dependency parsing using a constructed corpus.

Keywords: MultiWord Expressions, Dependency Parsing

1. Introduction

Multiword expressions (MWEs) consist of groups of tokens, which should be treated as a single syntactic or semantic unit. MWEs are also known as "idiosyncratic interpretations that cross word boundaries" (Sag et al., 2002). The recognition of MWEs in a sentence is important for such linguistic analyses as syntactic and semantic parsing, because it is known that combining an MWE into a single token improves accuracy for various NLP tasks, such as dependency parsing (Nivre and Nilsson, 2004), and constituency parsing (Arun and Keller, 2005).

For syntactic parsing that takes in MWEs, it is preferable that the information of an MWE (e.g., part of speech and span of tokens) be integrated into a corpus, such as a phrase or dependency treebank, because an MWE should be a syntactic unit. Actually, an MWE is grouped under "subtree" in French Treebank (Abeillé et al., 2003), which is often used in research focusing on both MWE recognition and syntactic parsing (Green et al., 2011; Candito and Constant, 2014). However, MWEs are not annotated in Penn Treebank, the standard corpus of English syntactic parsing.

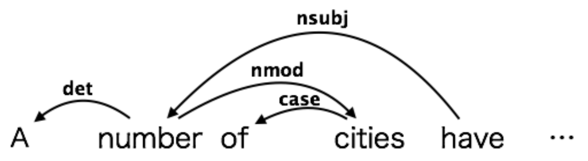
In dependency structure that takes MWEs into consideration (i.e., MWE-aware dependency), each MWE becomes a single node. Therefore, to convert word-based dependency to MWE-aware dependency directly, one could combine nodes in an MWE into a single node. Nevertheless, this method often leads to the following problem: A node derived from an MWE could have multiple heads and the whole dependency structure including MWE might be cyclic (Figure 1). This is mainly because Penn Treebank style annotation does not give the special treatment for MWEs. We discuss this problem further in Section 3.

In order to tackle the above problem, we propose a 3-step conversion method for making MWE-aware dependency structures. First, we establish an MWE as a single subtree in a phrase structure belonging to Penn Treebank (Figure 3a → Figure 3b). Second, we replace the subtree corresponding to the MWE by a preterminal with its leaf node as a child (Figure 4). The preterminal has the same part of speech as that of the MWE. Its child node is made by joining all components of the MWE with underscores. As a final step, we convert the phrase structure to Stanford Dependency (de Marneffe and Manning, 2008). In this way, we can avoid an occurrence of multiple heads and/or cycles in an MWE-aware dependency tree, because an MWE constitutes a single node in this dependency. We applied this conversion method to Ontonotes corpus (Pradhan et al., 2007) in order to construct a dependency corpus that takes into consideration MWEs. In this work, we focused on compound function words, which are one type of MWEs that serve as functional expressions. This is because compound function words have a variety of functionalities that may affect language analyses such as parsing and POS tagging. To reduce the cost of annotation in constituting each MWE as a subtree, we classified patterns of MWEs as seen in phrase structure in terms of ease of conversion. We manually annotated only instances including MWEs which were difficult to convert automatically.

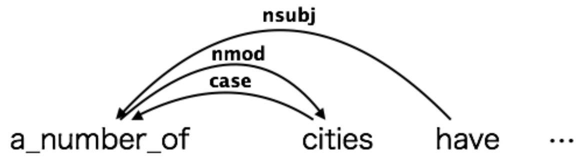
Furthermore, we evaluated the performance of the first order MST Parser (McDonald et al., 2005) on the constructed MWE-aware dependency corpus. We got an unlabeled attachment score (UAS) comparable to that obtained on the original Ontonotes corpus. Moreover, we qualitatively analyzed some test instances in which the MWE-aware dependency parser and the original dependency parser inferred the different outputs on the head of an MWE (Section 4).

2. Related Work

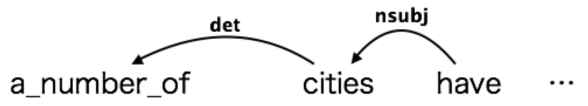
Shigeto et al. (2013) created a dictionary of English compound function words, and annotated those appearing in Penn Treebank. In their corpus, each MWE has its part of



(a) Word-based dependency structure (the syntactic head of "a number of cities" is "number")



(b) Squashed graph in which a cycle and multi-heads occur



(c) MWE-aware dependency structure (the syntactic head of "a number of cities" is "cities")

Figure 1: A cycle and multi-heads occur if we combine nodes in the MWE ("a number of") into a single node.

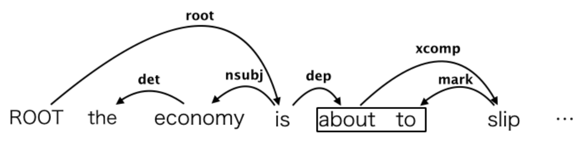
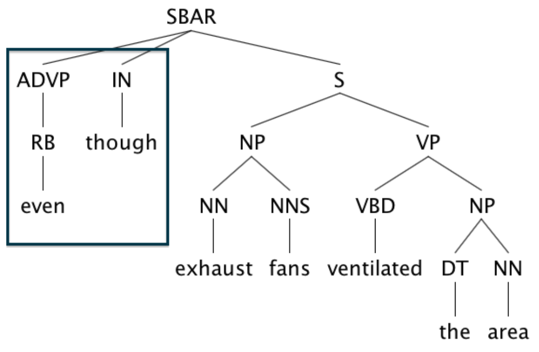
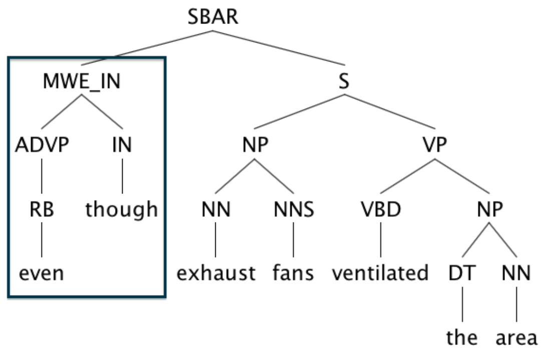


Figure 2: A cycle and multi-heads occur if we combine nodes in the MWE ("about to") into a single node.



(a) An LCA-tree before conversion. The square in this figure indicates the span of the MWE.



(b) The LCA-tree after conversion. Here, the MWE ("even though") becomes a single subtree.

Figure 3: Conversion of an LCA-tree in a "Simple" case

speech and span of tokens. But MWE-aware dependency is publicly unavailable.

As an example of an MWE-aware dependency corpus, there is Universal Dependency (Mcdonald et al., 2013). This project is developing a cross-linguistically consistent treebank annotation for many languages. Compound function words are annotated in a flat, head-initial structure, in which all words in an MWE modify the first word using a "mwe" label.

With regard to MWE-aware syntactic parsing, an MWE is recognized before or at the time of syntactic parsing (Green et al., 2011; Candito and Constant, 2014). Nivre and Nilsson (2004) conducted Swedish dependency parsing assuming perfect MWE recognition. They focused on multiword names (persons and places), numerical expressions and compound function words.

3. Construction of an MWE-aware Dependency Corpus

We built an MWE-aware dependency corpus on the basis of a corpus by Shigeto et al. (2013). In their corpus, an entire MWE is assigned a single part of speech. In that case, it is preferable that an MWE be a single node in an MWE-aware dependency structure. On the other hand, each word is treated as a node in conventional dependency structure.

If we try to directly convert word-based dependency to MWE-aware dependency, we need to combine nodes in an MWE into a single node. However, this naive approach often leads to the following problem: A node derived from an MWE could have multiple heads and the whole dependency structure including MWE might be cyclic. In Figure 1 for instance, "a_number_of" has multiple heads ("cities" and "have"), and a cycle occurs because of the following edges: "a_number_of" → "cities" and "cities" → "a_number_of" (Figure 1b). Therefore we need to remove one of the edges in order to get a dependency tree. Another problem that arises is the following: The syntactic head of "a number of cities" is "cities" in an MWE-aware dependency structure (Figure 1c), because "a_number_of" is a determiner. Nevertheless, we cannot get the correct dependent of "have" (i.e., "cities") by the above direct method. Similarly, if we combine words of the MWE into a single node in Figure 2, "about to" has multiple heads ("is" and "slip"), and a cycle occurs between "about to" and "slip".

In order to solve these problems, we adopted the following approach: First we establish an MWE in a phrase structure tree as a single subtree. After that, we convert phrase structure to dependency structure. With this method, we can avoid cycles and/or multiple heads.

Concretely, we built a corpus of dependency structures considering compound function words according to the following method:

(1) Find an MWE in a phrase structure tree of Ontonotes

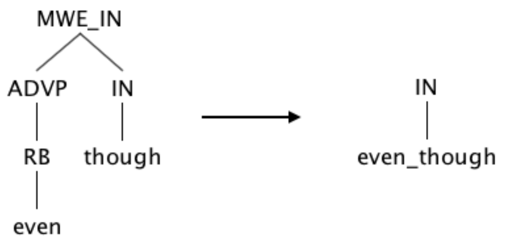


Figure 4: Replacement of a subtree in which MWE ("even though") is grouped

and establish it as a single subtree (Figure 3a → Figure 3b)[1]

(2) Replace the above subtree by a preterminal with its leaf node as a child. The preterminal has the same part of speech as that of the MWE. Its child node is made by joining all components of the MWE with underscores, as in Figure 4.

(3) Convert phrase structure to Stanford Dependency (de Marneffe and Manning, 2008)[2]

In Step (1), we convert an MWE into a single subtree in the phrase structure tree. For example, "even though" in Figure 3a is annotated as an MWE in Shigeto et al. (2013). We convert it as in Figure 3b. If we can convert the span of an MWE into a single subtree without influencing the structures of other subtrees, we call this instance "Simple". Otherwise the instance is "Complex". When grouping MWE, we focus on the LCA-tree, which is the subtree rooted in the Least Common Ancestor (LCA) of the components of the MWE. In Figure 3a, the tree rooted in the LCA of "even" and "though" (LCA here being SBAR) is the LCA-tree.

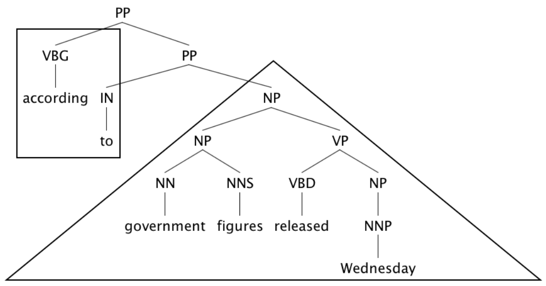
The method we described above relates to Finkel and Manning (2009). For joint parsing and named entity recognition, they classified named entities which do not correspond to a phrase in the constituency tree to the following two categories. A named entity belonging to the first category is contiguous multiple children of some nonterminal. This category corresponds to the above "Simple" case. On the other hand, A span of each named entity belonging to the second category crosses brackets in the parse tree. It corresponds to the above "Complex" case.

3.1. Simple Case

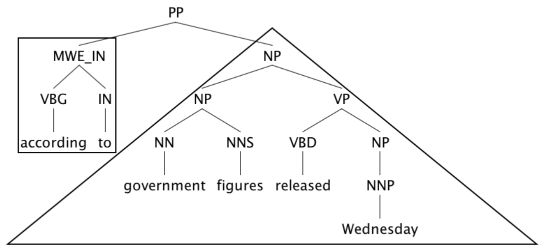
In the "Simple" case, we insert a new internal node under the LCA (Figure 3a → Figure 3b). This internal node covers precisely the span of the MWE.

3.2. Complex Case

In the "Complex" case, if we convert the MWE into a subtree (e.g. Figure 5a → Figure 5b), the structures of other subtrees need to be changed. For example, it is reasonable



(a) An LCA-tree before conversion. The span of the right subtree (from "to" to "Wednesday") partially overlaps the span of the MWE ("according to"). The triangle in this figure indicates the subtree (T_{post}) which covers precisely the span excluding the MWE.



(b) The LCA-tree after conversion. Here, The MWE ("according to") becomes a single subtree. The triangle in the figure indicates the subtree (T_{post}).

Figure 5: Conversion of an LCA-tree in a "Complex-Normal" case

to remove the right child of LCA (PP) in Figure 5a at the time of the conversion as in Figure 5b.

Further, we classified "Complex" into "Complex-Normal" and "Complex-Abnormal" based on the extent to which the LCA-tree is changed by the conversion.

3.2.1. Complex-Normal Case

In this case, we can convert an MWE into a single subtree without influencing subtrees which cover words outside the MWE. In Figure 5a for instance, the span of the right subtree (from "to" to "Wednesday") partially overlaps the span of the MWE ("according to"), but there is an internal node which covers the entire span to the right of the MWE (see the grandchild of the LCA (NP)).

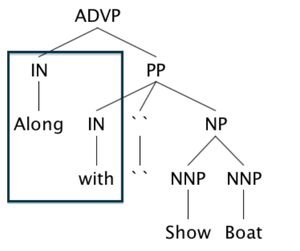
We define the subtree covering the span to the left of the MWE as T_{pre}; the subtree covering the MWE as T_{mwe}; and the subtree covering the span to the right of the MWE as T_{post}. In Figure 5a, the subtree surrounded by a triangle is T_{post}.

We converted the tree so that the LCA has T_{pre}, T_{mwe}, T_{post} as children (Figure 5a → Figure 5b).

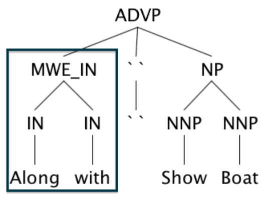
When both T_{pre} and T_{post} were present, we manually determined which of the following choices is preferable: LCA having T_{pre}, T_{mwe}, T_{post} as flat children, or LCA having a new internal node (covering T_{mwe} and T_{pre} or T_{post}) as a child.

[1]We utilized MWE information (position in sentence and part of speech) provided by Shigeto et al. (2013).

[2]We designated "-conllx -basic -makeCopulaHead -keepPunct" as an option for the conversion command.



(a) An LCA-tree before conversion. There is no internal node which covers precisely the entire span to the right of the MWE ("along with").



(b) The LCA-tree after conversion. Here, the MWE ("along with") becomes a single subtree.

Figure 6: Conversion of an LCA-tree in a "Complex-Abnormal" case

Case	No. of instances	Types of MWEs
Simple	5129	427
Complex-Normal	1742	117
Complex-Abnormal	57	27

Table 1: Gross corpus statistics of an MWE-aware Dependency Corpus.

3.2.2. Complex-Abnormal Case

In this case, we cannot avoid influencing subtrees outside the MWE when converting the MWE into a single subtree (Figure 6a → Figure 6b). The span of the right subtree of LCA (from "with" to "Boat") partially overlaps the span of the MWE ("along with"), and there is no internal node which covers precisely a span to the left or right of the MWE. Thus we manually determined how to combine a subtree covering the MWE with subtrees covering spans excluding the MWE.

Regarding "Simple" case and "Complex-normal" case, we decided a symbol of an LCA as follows[3]:

$$\arg\max_{X_{LCA}}(P(A \rightarrow B_1..X_{LCA}..B_m) \times P(X_{LCA} \rightarrow C_1..C_n))$$

We show the numbers of instances and types of MWEs for each case in our corpus (Section 00-24 of Wall Street Journal in Ontonotes) in Table 1 [4].

4. MWE-aware Dependency Parsing

In this section, we report dependency parsing based on the corpus we built according to the method described in Section 3.

4.1. Experimental Setting

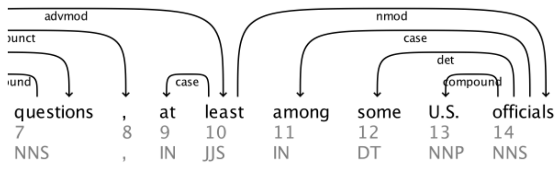
We trained and tested an original dependency parser and an MWE-aware dependency parser independently. The evaluation measure we adopted was UAS (unlabeled attachment score). We used the first-order MST Parser (McDonald et al., 2005) and the standard split for Ontonotes (Wall Street Journal): Sections 02-21 for training, 23 for testing. We also used gold part-of-speech tags both in training and testing.

[3]These probabilities are calculated from Ontonotes.

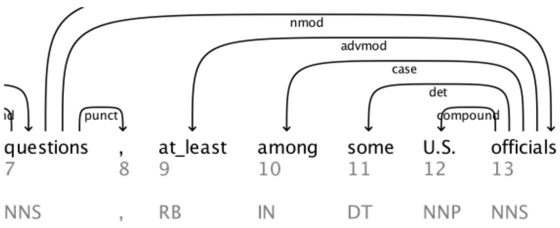
[4]In Complex-Normal case, 53 instances had both T_{pre} and T_{post}.

	UAS (total)	UAS (for sentences including MWEs)
Original	89.99	87.77
MWE-aware	90.01	87.84

Table 2: Experimental results for the original and the MWE-aware dependency parsing



(a) Dependency structure inferred by the original dependency parser.



(b) Dependency structure inferred by the MWE-aware dependency parser.

Figure 7: An instance for which the original dependency parser inferred an incorrect output and the MWE-aware dependency parser inferred a correct output.

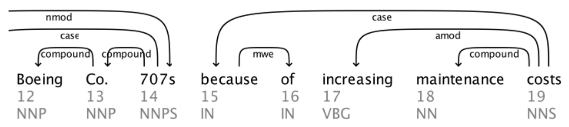
4.2. Experimental Results

We show the experimental results in Table 2. Because the original and the MWE-aware dependency structures are different, we cannot directly compare the results of these dependency parsers, however, we got comparable UAS results for the two parsers on both whole sentences (in a test set of 1640 sentences) and sentences including MWEs (266 sentences).

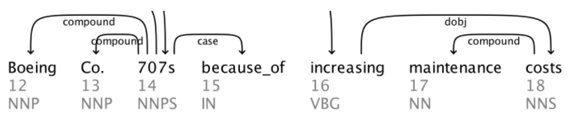
In the following, we describe the qualitative analysis. First, we show an instance for which the original dependency parser inferred an incorrect output and the MWE-aware dependency parser returned a correct output.

In Figures 7a and 7b, the original dependency parser infers "least", which is a component of the MWE ("at least"), as the head of "officials" incorrectly. On the other hand, the MWE-aware dependency parser infers "questions" correctly[5].

[5]The full sentence is "Mrs. Hills' remarks did raise questions, at least among some U.S. officials, about what exactly her stance is on U.S. access to the Japanese semiconductor market."


nmod
case
compound
compound
mwe
case
amod
compound
Boeing Co. 707s because of increasing maintenance costs
12 13 14 15 16 17 18 19
NNP NNP NNPS IN IN VBG NN NNS

(a) Dependency structure inferred by the original dependency parser.


compound
compound
case
dobj
compound
Boeing Co. 707s because_of increasing maintenance costs
12 13 14 15 16 17 18
NNP NNP NNPS IN VBG NN NNS

(b) Dependency structure inferred by the MWE-aware dependency parser.

Figure 8: An instance for which the original dependency parser inferred a correct output, but the MWE-aware dependency parser inferred an incorrect output.

In this instance, by recognizing the MWE before parsing, the MWE-aware dependency parser avoids an incorrect inference originating from the part-of-speech tag of a component of the MWE. This is consistent with the error analysis by Nivre and Nilsson (2004) for Swedish dependency parsing involving compound function words.

Next, we show an instance for which the original dependency parser inferred a correct output, but the MWE-aware dependency parser inferred an incorrect output. In Figures 8a and 8b, the original dependency parser infers "costs" correctly as the head of "because". However, the MWE-aware dependency parser infers "707s" as the head of MWE ("because of")[6].

In fact, this MWE and almost all MWEs in the test set (286 out of 289) also appear in the training set. Therefore, to make a correct inference for the above instance, we need to explore MWE-specific features (the word form and the part-of-speech tag of each component of the MWE) and/or higher order features for the dependency parser rather than dealing with unseen MWEs.

5. Conclusion

We created an English dependency corpus incorporating compound function words and conducted dependency parsing using the constructed corpus. Our corpus is available at: https://github.com/naist-cl-parsing/mwe-aware-dependency. In the future, we plan to design features for MWE-aware dependency parsing, and to integrate MWE recognition and dependency parsing. We also plan to explore a linguistic analysis in which MWE-aware dependency is preferable to word-based dependency.

6. Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 15K16053, 26240035. We are grateful to members of Computational Linguistics Laboratory and David Sell at NAIST, and anonymous reviewers for valuable feedback.

[6]The full sentence is "Earlier the company announced it would sell its aging fleet of Boeing Co. 707s because of increasing maintenance costs."

7. Bibliographical References

Abeillé, A., Clément, L., and Toussenel, F., (2003). *Treebanks: Building and Using Parsed Corpora*, chapter Building a Treebank for French, pages 165–187. Springer Netherlands, Dordrecht.

Arun, A. and Keller, F. (2005). Lexicalization in crosslinguistic probabilistic parsing. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL '05*, (June):306–313.

Candito, M. and Constant, M. (2014). Strategies for Contiguous Multiword Expression Analysis and Dependency Parsing. *In Proceedings of Association for Computational Linguistics*, pages 743–753.

de Marneffe, M.-C. and Manning, C. D. (2008). The Stanford typed dependencies representation. *In Proceedings of the Coling workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8.

Finkel, J. R. and Manning, C. D. (2009). Joint parsing and named entity recognition. *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics on - NAACL '09*, (June):326–334.

Green, S., Marneffe, M.-C. D., Bauer, J., and Manning, C. D. (2011). Multiword Expression Identification with Tree Substitution Grammars : A Parsing tour de force with French. *In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 725–735.

McDonald, R., Pereira, F., Ribarov, K., and Hajič, J. (2005). Non-projective dependency parsing using spanning tree algorithms. *In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 523–530.

Mcdonald, R., Nivre, J., Quirmbach-brundage, Y., Goldberg, Y., Das, D., Ganchev, K., Hall, K., Petrov, S., Zhang, H., Täckström, O., Bedini, C., Castelló, N. B., and Lee, J. (2013). Universal Dependency Annotation for Multilingual Parsing. *In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 92–97.

Nivre, J. and Nilsson, J. (2004). Multiword units in syntactic parsing. *In Proceedings of the LREC Workshop on Methodologies and Evaluation of Multiword Units in Real-World Applications*, pages 39–46.

Pradhan, S. S., Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., and Weischedel, R. (2007). OntoNotes: A unified relational semantic representation. *International Conference on Semantic Computing*, pages 517–524.

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 3rd International Conference on Intelligent Text Processing and Computational Linguistics*, pages 1–15.

Shigeto, Y., Azuma, A., Hisamoto, S., Kondo, S., Kouse, T., Sakaguchi, K., Yoshimoto, A., Yung, F., and Matsumoto, Y. (2013). Construction of English MWE Dictionary and its Application to POS Tagging. *In Proceedings of the 9th Workshop on Multiword Expressions*, pages 139–144.