A Method for Automatically Constructing Case Frames for English

Daisuke Kawahara, Kiyotaka Uchimoto

National Institute of Information and Communications Technology
3-5 Hikaridai Seika-cho, Soraku-gun, Kyoto, 619-0289, Japan
{dk, uchimoto}@nict.go.jp

Abstract

Case frames are an important knowledge base for a variety of natural language processing (NLP) systems. For the practical use of these systems in the real world, wide-coverage case frames are required. In order to acquire such large-scale case frames, in this paper, we automatically compile case frames from a large corpus. The resultant case frames that are compiled from the English Gigaword corpus contain 9,300 verb entries. The case frames include most examples of normal usage, and are ready to be used in numerous NLP analyzers and applications.

1. Introduction

Selectional preferences are an important knowledge source for fundamental analyzers and other applications of natural language processing (NLP). To practically use selectional preferences in these NLP systems, they should have wide coverage. Therefore, it is necessary to automatically construct a wide-coverage knowledge base from large corpora. Knowledge acquisition from large corpora has attracted attention in recent years. In particular, there has been a lot of research on acquiring subcategorization frames (Brent, 1993; Ushioda et al., 1993; Manning, 1993; Ersan and Charniak, 1996; Briscoe and Carroll, 1997; Gahl, 1998; Carroll and Rooth, 1998; Lapata, 1999; Korhonen and Preiss, 2003). Subcategorization frames represent argument patterns of verbs and thus are purely syntactic patterns[1]. In practice, subcategorization frames are effective for improving parsing (Zeman, 2002). However, since subcategorization frames are not semantic but syntactic patterns, expressions that have the same subcategorization frame can have different meanings (e.g., metaphors). NLP applications such as machine translation[2] and paraphrasing (Ellsworth and Janin, 2007) based on frames require consistency in the meaning of each frame.

This paper aims at automatically building semantics-oriented frames, like FrameNet (Baker et al., 1998), from a large raw corpus. We call them "case frames". Case frames describe what kinds of case slots each verb has and what kinds of nouns can fill each case slot. For example, let us show a case frame for the verb "arrest":

arrest

sbj:{police, authority, ...} *obj*:{people, suspect, ...}
pp:on:{charge, suspicion, ...}

Frequencies are attached to each case frame, case slot, and word. They can be effectively used in applications of case frames. Note that we focus on the construction of case frames of English verbs.

[1]Originally, subcategorization frames do not provide selectional preferences, but it is possible to preserve words that constitute these frames as in (Korhonen et al., 2006). These words can be used as selectional preferences.

[2]http://bulba.sdsu.edu/framenetmt/

2. Related Work

Subcategorization frames are most related to case frames. They are a kind of case frame, and represent generalized argument patterns of verbs. For example, a subcategorization frame for the verb 'put' is "NP put NP PP", which means 'put' takes a noun phrase (NP) as its subject, and an NP and a prepositional phrase (PP) as its complements. Subcategorization frames were constructed by hand in the early stages of NLP (Boguraev et al., 1987; Grishman et al., 1994; The XTAG Research Group, 1998). These handmade lexicons are used as the gold standard when evaluating automatic construction approaches, which are stated below.

The first systems to automatically learn subcategorization frames from corpora emerged roughly a decade ago (Brent, 1993). These systems focused on only a small number of predefined subcategorization frames. Subsequent approaches targeted larger sets of predefined subcategorization frames and used larger corpora (Ushioda et al., 1993; Manning, 1993; Ersan and Charniak, 1996; Gahl, 1998; Carroll and Rooth, 1998; Lapata, 1999). Another system automatically detected a set of subcategorization frames and constructed a lexicon from them (Briscoe and Carroll, 1997). To extract relevant subcategorization frames for each verb, many of these approaches made use of hypothesis testing. However, it was reported to have poor performance especially for low-frequency subcategorization frames (Briscoe and Carroll, 1997; Manning and Schütze, 1999). Furthermore, verb sense ambiguity, which was not distinguished by these systems, was also a cause of poor performance. Recently, Korhonen et al. proposed a sophisticated method of integrating improved hypothesis testing and word sense disambiguation (Korhonen, 2002; Korhonen and Preiss, 2003).

There has been some work on automatic construction of case frames for Japanese.

Haruno and Utsuro et al. proposed a method to acquire Japanese case frames from relatively small corpora (Haruno, 1995; Utsuro et al., 1997). The case frames produced by their methods consist of semantic features of a thesaurus instead of words. On this point, our method is different from theirs. Their methods find appropriate generalization levels of case slots with a machine learning tech-

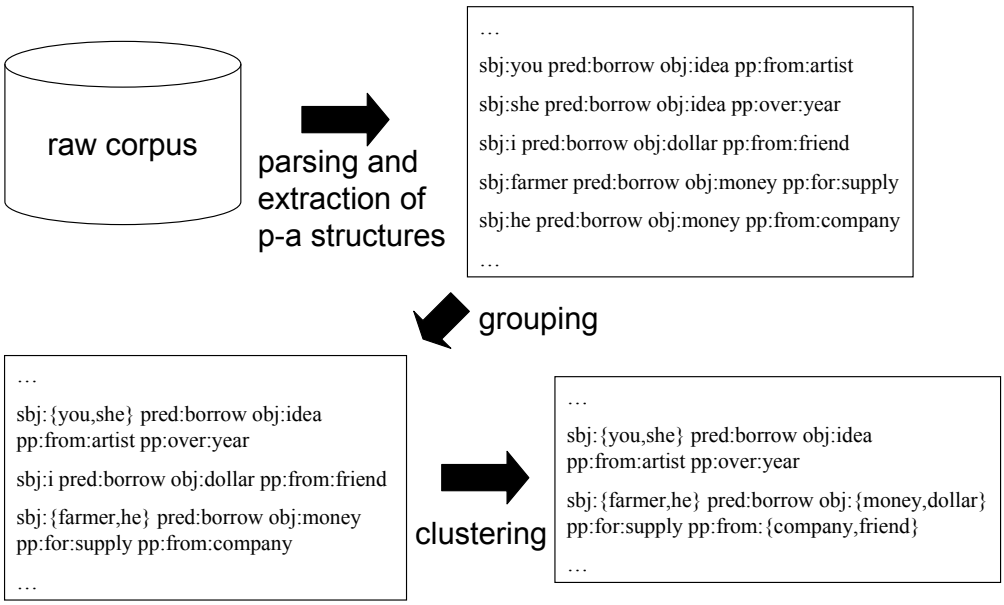

Figure 1: Overview of case frame construction.

nique or an information theoretic data compression technique. These generalization approaches are questionable from the viewpoint of the accuracy of the resulting case frame lexicons.

Kawahara and Kurohashi succeeded in constructing Japanese case frames from a huge Web corpus that consists of 500M Japanese sentences (Kawahara and Kurohashi, 2006). Since Japanese is head-final and has case-marking postpositions, it seems easier to build case frames for Japanese than for languages such as English.

3. The Specification of Case Frames

We define each verb as having case frames independently and each case frame as having several case slots. Here, we must decide the following two things:

- representation of case slots
- description of selectional preferences in case slots

For the representation of case slots, Fillmore and his followers have used deep cases. However, it is difficult to set up a definitive set of deep cases, and it is also difficult to determine to which case slot an argument is assigned. Instead of deep cases, we use surface cases for the representation of case slots: sbj (subject), obj (direct object), obj2 (indirect object), pp:in (prepositional phrase headed by "in"), pp:at and so on.

For the description of selectional preferences in case slots, two means can be considered: semantic classes and words themselves. If semantic classes are used, it is difficult to determine the appropriate generalization level of semantic classes, and it is also difficult to define the definitive set of semantic classes in order to describe the preferences without contradiction. We use words themselves as a means of describing selectional preferences. This is done by using a huge corpus. Also, we have an advantage in that our case frames can reflect real language usage by using word frequencies.

4. A Method for Automatically Constructing Case Frames

Our method for automatically constructing case frames has two key points. It can be used to

- extract predicate-argument structures from reliable automatic parses, and
- construct semantics-oriented case frames.

The overview of case frame construction is shown in Figure 1. In our approach, we regard predicate-argument structures as seeds. First, these structures are extracted from automatic parses of a large raw corpus. The problem is that the parses inevitably contain errors. To avoid the effect of the errors as much as possible, we use only reliable parses. Then, predicate-argument structures are aggregated and clustered to produce case frames. The clustering process is performed based mainly on the semantic similarity between constituent words. Thus, our case frames are semantics-oriented, but since case frames still have something to do with subcategorization patterns, we use also the similarity of argument patterns. Therefore, we use both the syntactic similarity of argument patterns and the semantic similarity between constituent words.

We construct case frames in the following way.

1. Tagging and parsing of a large raw corpus

   We apply tagging and parsing to a raw corpus. To easily extract predicate-argument structures, labeled dependency parsing is used. Dependency labels enable case slots to be identified.

   To obtain as reliable (accurate) parses as possible, we use relatively short sentences in a corpus. In practice, we extract sentences with 20 words or fewer from the English Gigaword as stated in the next section.

2. Extraction of predicate-argument structures from parses

   We extract predicate-argument structures from automatic parses of sentences that contain only one main verb in the active voice. Only headwords are preserved, and they are lowercased and lemmatized. For example, let us consider the following sentence:

   You borrowed an idea from another artist.

   From the parse of this sentence, the following predicate-argument structure is extracted.

   sbj:you pred:borrow obj:idea pp:from:artist

   We use this this notation to express a predicate-argument structure, in which "sbj", "obj" and "pp:from" are case slots, and "pred" means a predicate.

3. Grouping of predicate-argument structures by selecting the most dominant argument

   Predicate-argument structures are first collected for each verb. Then, they are grouped according to the most dominant argument in their predicate-argument structure. For instance, let us consider the following four predicate-argument structures.

   (a) sbj:you pred:borrow obj:idea pp:from:artist

   (b) sbj:she pred:borrow obj:idea pp:over:year

   (c) sbj:farmer pred:borrow obj:money pp:for:supply

   (d) sbj:he pred:borrow obj:money pp:from:company

   In this case, "obj" arguments are regarded as the most dominant ones, and thus (a) and (b) are classified into the group "obj:idea", and (c) and (d) are classified into the group "obj:money":

   (a') sbj:{you, she} pred:borrow obj:idea pp:from:artist pp:over:year

   (b') sbj:{farmer, he} pred:borrow obj:money pp:for:supply pp:from:company

   It is difficult to decide the order of dominant arguments, but in this paper we heuristically decided it as follows:

   (1) The "obj" argument is regarded as the most dominant.

   (2) If "obj" does not exist, "sbj" is selected.

   (3) Furthermore, if both of them are not found, "pp:*" is selected.

   (4) Any predicate-argument structure that does not contain any of the above is discarded.

   We call the result of this phase "initial case frames".

4. Clustering of the initial case frames

   We apply clustering to the initial case frames to merge very similar case frames. We use the similarity measure (Kawahara and Kurohashi, 2006), which is based on the similarity of case slot patterns and the similarity of constituent words. The similarity between words is calculated using the perl module, WordNet::Similarity[3].

Table 1: Case frame examples (numeral following each example denotes its frequency).

burn:1	
sbj	they:262, it:113, protester:99, ...
obj	flag:247, effigy:81, house:67, ...
pp:in	<num>:29 ramallah:14 brisbane:11, ...
pp:for	week:15, hour:6, month:5, ...
⋮	⋮
burn:2	
sbj	candle:26, lamp:5
pp:on	motor-scooter:7, altar:3, platform:1,
pp:for	day:2, steinhaeuser:1
⋮	⋮

5. Experiments

To build case frames for English verbs, we used the English Gigaword Second Edition (LDC2005T12) as a source corpus. This corpus contains approximately 100M sentences sourced from five international English-newswire services. To obtain reliable parses, we used sentences that consisted of 20 words or fewer. With this filtering, we obtained approximately 47M sentences.

For tagging the corpus, we used Tsuruoka's tagger[4](Tsuruoka and Tsujii, 2005). For a labeled dependency parser, we used the MSTParser[5](McDonald et al., 2006) , which achieved top results in the CoNLL 2006 (CoNLL-X) shared task of multilingual dependency parsing. We trained a parsing model on sections 2-21 of the WSJ portion of the Penn Treebank. This model achieves a labeled dependency accuracy of 89.9% and a complete sentence rate of 36.3% for section 23 of the WSJ [6]. For the sentences that consist of 20 words or fewer in this section, the dependency accuracy improved up to 91.5% and the complete rate becomes 56.4%.

As a result, we constructed case frames consisting of approximately 9,300 verbs. The average number of case frames for a verb was 5.5. Table 1 lists some examples of these case frames.

We evaluated the resultant case frames. We selected 20 verbs randomly and evaluated them by hand. The case frame evaluation is performed according to the following three criteria:

- Verb usage is disambiguated by dominant arguments. That is to say, there are no different meanings or case slot patterns in a case frame.

[3]http://www.d.umn.edu/~tpederse/similarity.html.

[4]http://www-tsujii.is.s.u-tokyo.ac.jp/~tsuruoka/postagger/

[5]http://sourceforge.net/projects/mstparser/

[6]The test sentences are also preprocessed by the Tsuruoka's tagger.

- Case frames must have obligatory case slots, such as "obj" case slots for transitive verbs.
- Case slots, except a dominant one, may contain an ineligible example. If this happens for a dominant case slot, such a case frame is deemed to not satisfy the first condition and is determined to be incorrect.

We obtained an accuracy of 88.4%. Major errors were caused by the incorrect selection of dominant arguments. Furthermore, clustering caused some errors because we did not handle WordNet synsets. In future, we will investigate these problems in order to improve the accuracy.

6. Conclusion

This paper described a method of constructing a wide-coverage case frames for English. The acquired case frames contain most examples of normal usage and are ready to be applied to numerous NLP applications. In future, we will handle more sentences extracted from the Web. Furthermore, we will consider a more sophisticated way of determining the dominant argument of each predicate-argument structure.

7. References

Collin Baker, Charles J. Fillmore, and John Lowe. 1998. The Berkeley FrameNet Project. In *Proceedings of COLING-ACL1998*, pages 86–90.

Bran Boguraev, Ted Briscoe, John Carroll, David Carter, and Claire Grover. 1987. The derivation of a grammatically-indexed lexicon from the Longman Dictionary of Contemporary English. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, pages 193–200.

Michael Brent. 1993. From grammar to lexicon: Unsupervised learning of lexical syntax. *Computational Linguistics*, 19(2):243–262.

Ted Briscoe and John Carroll. 1997. Automatic extraction of subcategorization from corpora. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, pages 356–363.

Glenn Carroll and Mats Rooth. 1998. Valence induction with a head-lexicalized PCFG. In *Proceedings of the 3rd Conference on Empirical Methods in Natural Language Processing*, pages 36–45.

Michael Ellsworth and Adam Janin. 2007. Mutaphrase: Paraphrasing with framenet. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 143–150.

Murat Ersan and Eugene Charniak, 1996. *A Statistical Syntactic Disambiguation Program and What It Learns*, pages 146–157. Springer.

Susanne Gahl. 1998. Automatic extraction of subcorpora based on subcategorization frames from a part-of-speech tagged corpus. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 428–432.

Ralph Grishman, Catherine Macleod, and Adam Meyers. 1994. COMLEX Syntax: Building a computational lexicon. In *Proceedings of COLING1994*, pages 268–272.

Masahiko Haruno. 1995. A case frame learning method for Japanese polysemous verbs. In *Proceedings of the AAAI Spring Symposium: Representation and Acquisition of Lexical Knowledge: Polysemy, Ambiguity, and Generativity*, pages 45–50.

Daisuke Kawahara and Sadao Kurohashi. 2006. Case frame compilation from the web using high-performance computing. In *Proceedings of LREC2006*.

Anna Korhonen and Judita Preiss. 2003. Improving subcategorization acquisition using word sense disambiguation. In *Proceedings of ACL2003*, pages 48–55.

Anna Korhonen, Yuval Krymolowski, and Ted Briscoe. 2006. A large subcategorization lexicon for natural language processing applications. In *Proceedings of LREC2006*.

Anna Korhonen. 2002. Semantically motivated subcategorization acquisition. In *Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition*, pages 51–58.

Maria Lapata. 1999. Acquiring lexical generalizations from corpora: A case study for diathesis alternations. In *Proceedings of ACL1999*, pages 397–404.

Christopher Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts.

Christopher Manning. 1993. Automatic acquisition of a large subcategorization dictionary from corpora. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 235–242.

Ryan McDonald, Kevin Lerman, and Fernando Pereira. 2006. Multilingual dependency analysis with a two-stage discriminative parser. In *Proceedings of CoNLL-X*, pages 216–220.

The XTAG Research Group, editor. 1998. *A Lexicalized Tree Adjoining Grammar for English*.

Yoshimasa Tsuruoka and Jun'ichi Tsujii. 2005. Bidirectional inference with the easiest-first strategy for tagging sequence data. In *Proceedings of HLT-EMNLP2005*, pages 467–474.

Akira Ushioda, David Evans, Ted Gibson, and Alex Waibel. 1993. The automatic acquisition of frequencies of verb subcategorization frames from tagged corpora. In *Proceedings of the Workshop on Acquisition of Lexical Knowledge from Text*, pages 95–106.

Takehito Utsuro, Takashi Miyata, and Yuji Matsumoto. 1997. Maximum entropy model learning of subcategorization preference. In *Proceedings of the 5th Workshop on Very Large Corpora*, pages 246–260.

Daniel Zeman. 2002. Can subcategorisation help a statistical dependency parser? In *Proceedings of COLING2002*, pages 1156–1162.