Acquisition of Linguistic Patterns for Knowledge-Based Information Extraction

Jun-Tae Kim and Dan I. Moldovan, *Member, IEEE*

***Abstract*—This paper presents an automatic acquisition of *linguistic patterns* that can be used for knowledge-based information extraction from texts. In knowledge-based approach to information extraction, linguistic patterns play a central role in the recognition and classification of input texts. Although the knowledge-based approach has been proved effective for information extraction on limited domains, there are difficulties in construction of a large number of domain-specific linguistic patterns. Manual creation of patterns is time consuming and error prone, even for a small application domain. To solve the scalability and the portability problem, an *automatic acquisition* of patterns must be provided. In this paper, we present the PALKA (Parallel Automatic Linguistic Knowledge Acquisition) system that acquires linguistic patterns from a set of domain-specific training texts and their desired outputs. A specialized representation of patterns called *FP-structures* has been defined. Patterns are constructed in the form of FP-structures from training texts, and the acquired patterns are tuned further through the generalization of semantic constraints. Inductive learning mechanism is applied in the generalization step. The PALKA system has been used to generate patterns for our information extraction system developed for the fourth Message Understanding Conference (MUC-4). The MUC-4 was an ARPA-sponsored competitive evaluation of text analysis systems. Experimental results with a set of news articles from MUC-4 are discussed.**

***Index Terms*—Knowledge-based natural language processing, information extraction, linguistic knowledge acquisition, inductive learning.**

I. Introduction

KNOWLEDGE-BASED natural language processing techniques have been successfully applied to information extraction from texts on a limited domain [25]. In knowledge-based information extraction, various forms of domain-specific *linguistic patterns* have been widely used for interpretation of input texts. A pattern like `[(perpetrator) attack (target) with (instrument)]` can be matched to the sentence "*Urban guerrillas attacked the administration office with explosives*" to extract information related to a terrorist event. By using domain-specific patterns, one can achieve fast and efficient text processing by directly mapping input sentence to its meaning without full syntactic analysis and without applying conversion rules from syntactic structure to semantic interpretation [11], [9], [17], [23].

Although the knowledge-based approach has been proved effective for information extraction on limited domains, there are difficulties in construction of a large number of domain-specific linguistic patterns. Manual creation of patterns is time consuming and error prone, even for a small application domain. As the size of a domain increases, creating knowledge base of patterns becomes an engineering bottleneck. Porting an existing system to a new domain takes a considerable amount of time because of the knowledge base construction. To solve the scalability and the portability problem, *automatic acquisition* of patterns must be provided.

In this paper, the PALKA (Parallel Automatic Linguistic Knowledge Acquisition) system that automatically acquires linguistic patterns from a training corpus is presented. The major goal of this system is to facilitate the construction of a large knowledge base of linguistic patterns. PALKA uses a set of sample inputs and outputs on a specific domain to construct linguistic patterns. A pattern is represented as a specialized form called an *FP-structure*. An FP-structure is a pair of a meaning frame and a phrasal pattern. The acquisition is performed as a feedback to the parser. Whenever a trial parsing of relevant sentence fails due to lack of an appropriate pattern, the acquisition system constructs a new pattern. The acquired patterns are further tuned through a series of generalizations of semantic constraints. Inductive learning mechanism [21], [20] is applied to the generalization steps.

The underlying assumption of our approach to the pattern representation and acquisition method is that, in a limited domain, a relatively small number of expressions is frequently used to describe certain information. Therefore, even though we represent a full phrase as a pattern, the number of patterns for the extraction of certain information would be finite. Furthermore, most of patterns for a certain application domain can be covered from a relatively small set of training corpus from that domain. Some of the experimental results show this effect.

The background of this research is the information extraction task of the fourth Message Understanding Conference (MUC-4). The MUC-4 was an ARPA-sponsored competitive performance evaluation of text analysis systems [25]. The PALKA system has been used to generate patterns for our information extraction system developed for the MUC-4. Since PALKA system was initially developed for the MUC-4 application, we use example texts and templates from the MUC-4 corpus to describe the system. However, PALKA can be easily adopted to other domains if appropriate knowledge sources are provided. In the next sections, the MUC-4 evaluation and the information extraction task are briefly described. The acquisition of patterns for information extraction is described in Section III, IV, and V, and the experimental results with the MUC-4 corpus are discussed in Section VI. Several related works are discussed in Section VII.

J.-T. Kim is with the Department of Computer Engineering, Dongguk University, Chung-Gu, Pil-dong, 3-16, Seoul, Korea 100-715.

D.I. Moldovan is with the Department of Computer Science and Engineering, Southern Methodist University, Dallas, TX 75275-0122; e-mail:moldovan@seas.smu.edu.

IEEECS Log Number K95062.

II. Information Extraction: MUC Tasks

MUCs are ARPA-sponsored series of conferences that concern the evaluation of natural language processing systems. The overall objective of the evaluations is to advance our understanding of the current text analysis techniques, as applied to the performance of an information extraction task. A typical task is to extract relevant information from on-line texts, and fill slots of pre-defined templates.

In the fourth Message Understanding Conference (MUC-4), 17 text analysis systems from industry and universities were involved in the evaluation. In MUC-4, the inputs were on-line news articles on terrorist incidents in Latin America, and the outputs were a set of templates representing a partially formatted database. An example text and corresponding template are shown in Fig.1. One should note that although information extraction requires understanding of texts in part, it is different from producing an in-depth representation of the content of complete text. The formats of inputs and outputs are predefined for an application domain. In developing systems, each participating sites were given a training corpus of 1,500 texts along with associated answer keys (instantiated templates), and the template filling rules.

TST1-MUC3-0073

SANTIAGO, 29 DEC 89 (EFE) -- [TEXT] POLICE TODAY REPORTED THAT DURING THE LAST FEW HOURS TERRORISTS STAGED THREE BOMB ATTACKS AGAINST U.S. PROPERTIES.

. . . . .

AT 0115 THIS MORNING (0415 GMT) INCENDIARY BOMBS WERE HURLED AT A MORMON TEMPLE AT NUNOA DISTRICT, IN SANTIAGO. THE BOMBS CAUSED MINOR DAMAGE. AT THE TIME OF THE ATTACK THE BUILDING WAS EMPTY, ACCORDING TO THE SOURCES.

THE ATTACKERS PAINTED THE WALLS AND LEFT PAMPHLETS WITH ULTRA-LEFTIST MESSAGES OF THE LAUTARO YOUTH FRONT. THE MANUEL RODRIGUEZ PATRIOTIC FRONT (FPMR), WHICH THE PINOCHET REGIME CONSIDERS TO BE THE COMMUNIST PARTY'S ARMED BRANCH, ANNOUNCED FOLLOWING THE U.S. INVASION OF PANAMA THAT IT WOULD ATTACK "U.S. INTERESTS IN CHILE."

(a) The sample text

0. MESSAGE: ID	TST1-MUC3-0073
1. MESSAGE: TEMPLATE	1
2. INCIDENT: DATE	29 DEC 89
3. INCIDENT: LOCATION	CHILE: SANTIAGO (CITY): NUNOA (DISTRICT)
4. INCIDENT: TYPE	BOMBING
5. INCIDENT: STAGE OF EXECUTION	ACCOMPLISHED
6. INCIDENT: INSTRUMENT ID	INCENDIARY BOMBS
7. INCIDENT: INSTRUMENT TYPE	BOMB
8. PERP: INCIDENT CATEGORY	TERRORIST ACT
9. PERP: INDIVIDUAL ID	-
10. PERP: ORGANIZATION ID	"MANUEL RODRIGUEZ PATRIOTIC FRONT"
11. PERP: ORGANIZATION CONFIDENCE	POSSIBLE
12. PHYS TGT: ID	"MORMON TEMPLE"
13. PHYS TGT: TYPE	OTHER: "MORMON TEMPLE"
14. PHYS TGT: NUMBER	1
15. PHYS TGT: FOREIGN NATION	UNITED STATES: "MORMON TEMPLE"
16. PHYS TGT: EFFECT OF INCIDENT	SOME DAMAGE: "MORMON TEMPLE"
17. PHYS TGT: TOTAL NUMBER	1
18. HUM TGT: NAME	-

. . . . .

(b) The sample template

Fig. 1. The example text and template.

The evaluations of each system were performed based on two principal measures: *Recall* and *Precision*. Recall is the number of correct answers produced by the system divided by the number of total possible correct answers. It measures how comprehensive the system is. Precision is the number of correct answers by the system divided by the number of all answers provided by the system. It measures the system accuracy. For example, if there are 100 possible answers and the system provided 60 answers and 30 of them were correct, then the recall is 30% and the precision is 50%. In MUC-4 evaluation, the average scores of 17 participants were 35% recall and 33% precision.

The information extraction task on a limited domain is quite different from a general natural language understanding task. For natural language understanding, one must be able to process full complexity of language, and produce an output representation that presents all the meanings (including implicit meanings) of the input sentence. A classical approach is analyzing sentences in two steps. First, a syntactic analysis module parses the input into a syntactic parse structure by using a syntactic grammar. Then a syntax to semantics mapping module converts the syntactic structure to a meaning representation by using a set of semantic interpretation rules. To produce a complete meaning representation, other processes like reference resolution are necessary.

It is well recognized that for information extraction in a limited domain, a full syntactic analysis or a comprehensive semantic interpretation are not necessary. There is only a small number of event categories, such as `BOMBING` and `KILLING`, to which input texts can be mapped. Also, the types of information that need to be extracted are pre-defined. Therefore, even for a relevant sentence, only several terms that carry relevant information need to be interpreted. Because of these characteristics, many successful systems in MUC-4 used various forms of *linguistic patterns*, and performed pattern matching for the interpretation of input texts. For example, the underlined sentence in Fig. 1 can be matched to a pattern like `[(instrument) BE HURL AT(target)]` to produce a `BOMBING` incident type with `(instrument: INCENDIARY BOMBS)` and `(target: MORMON TEMPLE)`.

This knowledge-based approach has been proved effective for information extraction, but it needs a large number of domain-specific linguistic patterns. The main issue of this paper is how to provide scalability and portability by automating the construction of these patterns.

III. Automatic Pattern Acquisition

The goal of the PALKA system is to build automatically a knowledge base of domain-specific linguistic patterns. This section presents the representation of patterns, the basic approach to pattern acquisition, and the knowledge sources used by PALKA.

A. Representation of Patterns: FP-Structure

Information extraction task on a limited domain is different from a general natural language understanding task. The characteristics of information extraction can be summarized as follows:

- A small number of event categories leading to many-to-one mappings.
- The types of information are predefined.
- Information can be found anywhere in the sentence (not only in the subject or the object of the sentence, but also in the prepositional phrases or in the modifier).

For example, all the following sentences contain the same information—the *category of event* is *bombing*, the *instrument* is *dynamite*, and the *target* is *administration office*.

1) The dynamite *exploded* inside the administration office.
2) The dynamite *destroyed* the windows of the administration office.
3) The dynamite *was hurled* to the administration office.
4) The dynamite explosion *caused* serious damage to the administration office.
5) The administration office *was damaged* by the dynamite explosion.
6) An attack with a dynamite in front of the administration office *has left* one person injured.

Mapping above sentences to *explode-event*, *hurl-event*, or *leave-event* has no meaning when our intent is to classify them as a *bombing-event*. In sentence 2, the *target* of the event is not the *window* (the object of the sentence). In sentence 4, the *instrument* is not the *explosion* (the subject of the sentence). An efficient representation should map various expressions to one of desired categories, such as *bombing*, and detect the information carrying words or phrases from anywhere in the sentence.

Based on these observations, we represent a linguistic pattern as a pair of a meaning frame defining the types of information, and a phrasal pattern describing the syntactic ordering. This representation is called the *FP-structure* (Frame-Phrasal pattern structure) [13]. The knowledge base is organized as a network of FP-structures and a concept hierarchy. Fig. 2 shows an example of an FP-structure. A meaning frame is represented by a root, a set of slots, and semantic constraints on fillers. A phrasal pattern is an ordered combination of lexical entries or semantic categories (concepts in the concept hierarchy). To combine a phrasal pattern and a meaning frame, each slot of the frame is linked to the corresponding element in the phrasal pattern.

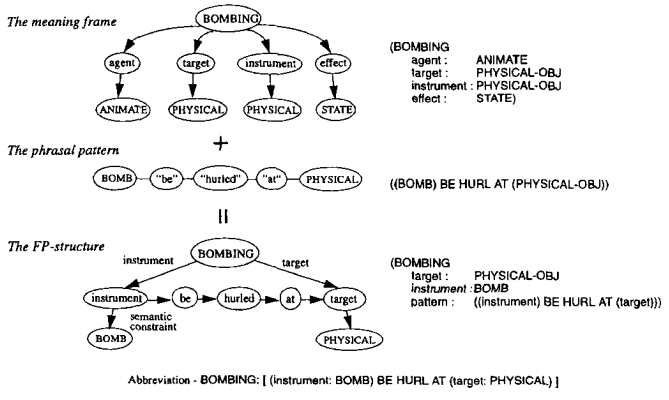


Fig. 2. The frame-phrasal pattern representation.

The FP-structures are used by the parser of the information extraction system to recognize input texts. By matching a phrase in the input text to the elements in a phrasal pattern, an FP-structure is activated, and by using the activated meaning frame, relevant information is extracted. Input words activate the phrasal pattern elements either directly or through the *isa* hierarchy of concepts. More details of the information extraction procedure of our system can be found in [23].

Similar representations can be found in phrasal lexicons [1] [34], pattern-concept pairs [32], phrasal patterns [9], concept nodes [17], and lexicosemantic patterns [12]. The FP-structure representation is different from these representations in several ways such as:

1) a meaning is represented by using pre-defined information types,
2) a full phrase is specified as a pattern to be matched, and
3) a semantic constraint is specified by using a concept or a combination of concepts in the domain concept hierarchy.

As one can see in the previous examples, the meaning of a phrase, or the category of an event, cannot be simply recognized by a main verb. There can be various domain dependent expressions for the `BOMBING` event, and such patterns can only be acquired by looking at actual texts on the domain. Representing full phrase as a pattern is feasible only when an application domain is specific and limited. The finiteness of expression on a limited domain is discussed in section VI.

B. Basic Approach to Acquisition

In our approach, the acquisition process is performed as a *feedback* to the parser. Fig. 3 shows the conceptual mechanism of pattern acquisition. Depending on the current status of the knowledge base, the parser may produce one of the following results: 1) correct interpretation, 2) no interpretation, or 3) incorrect interpretation. Examples of each case and corresponding actions by the acquisition system are as follows:

- *Case 1*: Correct interpretation (desired output = parser output).
  1. Pattern: *Appropriate*
  2. Action: None
- *Case 2*: No interpretation (desired output $\neq \phi$, parser output $= \phi$).
  1. Pattern: `BOMBING: [(instrument: DYNAMITE) EXPLODE]` (or no pattern)
  2. Sentence: "`A POWERFUL BOMB EXPLODED IN FRONT OF THE BUILDING`"
  3. Interpretation: None
  4. Action: Create a new pattern and generalize it with existing ones
- *Case 3*: Incorrect interpretation (*desired output* $= \phi$, *parser output* $\neq \phi$).
  1. Pattern: `BOMBING: [(instrument:THING) EXPLODE]`
  2. Sentence: "`THE FOREIGN DEBT CRISIS EXPLODED IN ANDEAN COUNTRIES`"
  3. Interpretation: `BOMBING-EVENT, instrument:FOREIGN DEBT CRISIS`
  4. Action: Specialize the pattern

In case 2, the parser produces no interpretation since the semantic constraint `DYNAMITE` is too specific to be matched by the input sentence. In this case, a new pattern is created from the input sentence, and generalized with previous patterns. In case 3, the input sentence is misinterpreted as a `BOMBING` event since the semantic constraint is over-generalized as `THING`. In this case, the semantic constraint is specialized to

an appropriate level. Through the acquisition process described above, the system creates a consistent and semantically correct knowledge base. When the system creates a new pattern, or generalizes/specializes a pattern, it uses the following knowledge sources.

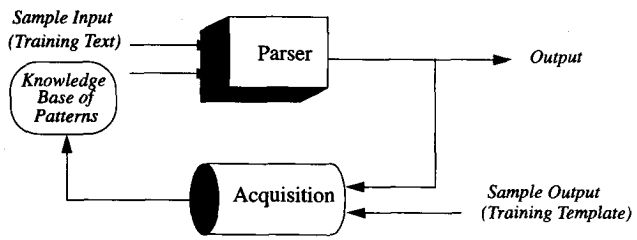


Fig. 3. Conceptual diagram of acquisition as a feedback to the parser.

C. The Knowledge Sources

FP-structures cannot be acquired solely from texts, because semantic knowledge is involved. Possible knowledge sources are human, contextual knowledge, tagged texts, or filled templates. Using human knowledge introduces a fully interactive system which is out of our interests. Using contextual knowledge is unrealistic since it is not always available and is difficult to provide. In our system training *texts* and corresponding database *templates* (as illustrated in Fig. 1) are used as the two major knowledge sources. A text is a set of news articles on a specific domain. The domain currently used is concerned with terrorist events in Latin America. PALKA uses the text to acquire phrasal patterns. A template is an output representation of a sample text which is generated by hand. It contains all the relevant information extracted from the text. Currently, 1,400 news articles and their corresponding templates are available on line for a system training purpose. PALKA uses the templates to map a phrasal pattern to a corresponding frame. When templates are not available, PALKA acquires the mapping information through user interaction. Other knowledge sources used by PALKA are:

- The *concept hierarchy* contains a general classification of objects, events, states, and domain-specific concepts. It is used to specify a semantic constraint of each element in an FP-structure. It is also used for the generalization of patterns.
- The dictionary maps an input word to one or more concepts in the concept hierarchy. For example, "dynamite" is mapped to the concept BOMB, and "house" is mapped to the concept BUILDING, which is linked to the concept PHYSICAL-OBJECT through isa relations.
- The *frame definition* represents the type of information for a specific domain. The frame definition of BOMBING is described in the next section.

IV. Acquisition Procedure in PALKA

PALKA is an automatic pattern acquisition tool. It acquires linguistic patterns for a given meaning frame. Phrasal patterns (syntactic information) are acquired from texts, and mappings to the frame (semantic information) are acquired from templates. Fig. 4 shows the functional structure of the PALKA system. For a given frame definition, the PALKA system selects candidate sentences which may have relevancy, and converts them into simple clauses. After trial parsing, the user determines the correctness of the parsing output. If there is no output for a relevant sentence, a new FP-structure is created through *FP-mapping*, *FP-structure construction*, and *generalization*. If the output is incorrect, the matched pattern in the knowledge base is modified through *specialization*. In this section, the acquisition procedure is described in detail with examples.

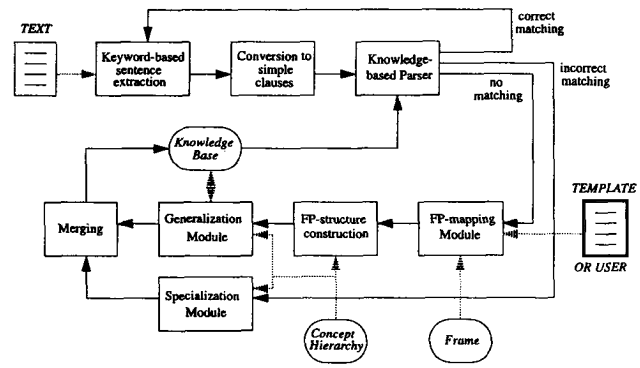


Fig. 4. The functional structure of PALKA.

A. Frame Definition and Sentence Extraction

The acquisition of FP-structures is performed for one frame at a time. For example, the system first acquires all the patterns for the BOMBING event frame, and then for the KILLING frame, and so on. In what follows, the acquisition procedure in PALKA is described by using the BOMBING frame example. The BOMBING frame is initially defined as shown in Fig. 5a.

The first slot isa points to a more general frame in the knowledge base to which this frame is connected. In the second slot keyword, several keywords are specified. Relevant sentences are extracted from sample texts by using these keywords. The keywords are selected by user from training corpus. Important nouns are selected easily from templates, and verbs are selected by looking at several relevant sentences. Only the root forms of words are specified in the keywords, and the syntactic variations are handled by the system. Specifying more keywords results in faster acquisition because more sentences would be extracted and tested. However, even insufficient set of keywords does not cause an incomplete acquisition. It only affects the acquisition rates. For example, even if Molotov-cocktail is not included in the keywords, the sentence that contains a Molotov-cocktail can be extracted because of other keywords like explode. Furthermore, even if no sentence with Molotov-cocktail is extracted, generalizations of patterns will cover the word, since the word is also mapped to general concepts such as BOMB or EXPLOSIVE in the concept hierarchy. If a pattern is constructed with a semantic constraint BOMB, the word Molotov-cocktail can be matched.

```
(BOMBING
    isa:          (TERRORIST-ACTION)
    keyword:      (bomb explosive dynamite grenade explode throw hurl explosion . . . )
    agent:        (ANIMATE)              ; optional
    target:       (PHYSICAL-OBJECT)      ; optional
    instrument:   (PHYSICAL-OBJECT)      ; optional
    effect:       (STATE))               ; optional
```

(a) The frame definition of BOMBING event

```
THE PERUVIAN POLICE HAVE REPORTED THAT A GRENADE WAS HURLED
AT A HOUSE IN THE SANBORJA NEIGHBORHOOD, WHERE U.S. MARINE RESIDE.
```

(b) The sentence extracted from a text

```
[THE PERUVIAN POLICE]noun-group [HAVE REPORTED]verb-group [THAT]rel-pronoun
[A GRENADE]noun-group [WAS HURLED]verb-group [AT]preposition
[A HOUSE]noun-group [IN]preposition [THE SANBORJA NEIGHBORHOOD]noun-group
[,]punctuation [WHERE]rel-pronoun [U.S. MARINE]noun-group [RESIDE]verb-group [.]punctuation
```

(c) The result of grouping words

```
1. [PERUVIAN POLICE] [HAVE REPORTED] [IT]
2. [GRENADE] [WAS HURLED] [AT] [HOUSE] [IN] [SANBORJA NEIGHBORHOOD]
3. [U.S. MARINE] [RESIDE] [IN] [SANBORJA NEIGHBORHOOD]
```

(d) The result of decomposition

```
[(instrument: GRENADE)] [WAS HURLED] [AT] [(target: HOUSE)]
[IN] [SANBORJA NEIGHBORHOOD]
```

(e) The result of FP mapping

```
(BOMBING
    target:       (BUILDING)
    instrument:   (GRENADE)
    pattern:      ((instrument) BE HURL AT (target)))
```

(f) The result of FP-structure construction

Fig. 5. The result of each step for FP-structure acquisition

The other four slots—`agent`, `target`, `effect`, and `instrument`—indicate the types of information used in this domain. For each slot, an initial semantic constraint is specified. These constraints do not represent the optimal ones. It is used for initial selection of candidates for each slot, only when output templates are not available. If output templates are available as a part of training corpus, these initial constraints are not necessary. The constraints for each slot are determined during acquisition through generalization. When a pattern is initially constructed from the first example sentence, the most specific concept from that sentence is chosen as a constraint. Later, as more example sentences are processed, the constraint is generalized until it reaches to the optimal level.

By using the keyword "`grenade`," the sentence in Fig. 5b is extracted from the text, as a candidate to a relevant sentence.

B. Conversion to Simple Clauses

The original text consists of complex sentences which contain relative clauses, nominal clauses, conjunctive clauses, etc. Since FP-structures are constructed for single clauses, it is necessary to convert a complex sentence to a set of simple clauses. A simple phrasal parser converts the extracted sentence into simple clauses through the following steps.

1) *Step 1. Grouping words:* The phrasal parser groups words based on each word's syntactic category and ordering rules for noun-groups and verb-groups. Basic syntactic disambiguation of word category is performed at this stage. The result of grouping words for the example sentence is shown in Fig. 5c.
2) *Step 2. Simplification and decomposition:* After grouping is performed, the phrasal parser first simplifies the sentence by eliminating several unnecessary elements such as determiners, adverbs, quotations, brackets, and so on. Then it converts the simplified sentence into several simple clauses by using conversion rules. The conversion rules include separation of relative clauses, nominal clauses, and conjunctive clauses. The three simple clauses shown in Fig. 5d are the results of the phrasal parsing.

Based on the keywords specified in the `BOMBING` frame, only the second clause is selected for further processing. To describe the *FP mapping module* and the *FP-structure construction*, we assume that the output of the parser of clause 2 is NIL (i.e., no matching).

C. FP Mapping Module

At this point, the definition of the `BOMBING` frame is available (the *meaning frame*), and the simple clause pattern is extracted (the *phrasal pattern*). To construct an FP-structure from these, links between the frame slots and the phrasal pattern elements should be established. There are two different modes of operation according to the availability of templates.

1) *Automatic mapping mode*: If templates are available, PALKA finds out mapping by using the information in a corresponding template[1]. Each slot of the frame definition corresponds to one or more slots in the template. For example, the `target` slot of the frame corresponds to the `PHYS TGT:` slot and `HUM TGT:` slot of the template. For each slot of the frame, the system searches through the template to pick up fillers for that slot. Then each element in the phrasal pattern is compared with the collected fillers. If an element is matched with a filler, then a link between the corresponding slot and the matched element is made.
2) *Interactive mapping mode:* In case templates are not available, PALKA first finds out candidates for each slot by using the initial semantic constraint specified for each slot in the frame definition, and then establishes the mapping through user interaction. For example, the general semantic constraint of the `target` slot is `PHYSICAL-OBJECT`, and so the candidate elements for `target` slot are "`GRENADE`" and "`HOUSE`," since their semantic categories are under the concept `PHYSICAL-OBJECT` in the concept hierarchy. The candidates for each slot are presented to the user, and user selects one for each slot.

The mapping shown in Fig. 5e is obtained after the FP mapping procedure. The `agent` and `effect` slots are not linked, since either 1) no element in the phrasal pattern is matched to the corresponding fillers, or 2) no candidates which satisfy the initial semantic constraint are found.

D. FP-Structure Construction

After all the links are established, PALKA constructs an FP-structure based on the mapping information. The basic strategy for constructing an FP-structure is to include the mapped elements and the main verb, and discard the unmapped elements. This is because, as mentioned earlier, we need to pick up rele-

1. Also, semantically tagged texts can be used to provide necessary mapping information.

vant information only. Unmapped elements contain no relevant information. However, an obligatory element like main verb, subject or object is included even if it is unmapped. This is to facilitate the matching process of parsing when the pattern is actually used for information extraction. Some basic rules for the FP-structure construction are:

Rule 1. All mapped elements are replaced by their semantic categories.

Rule 2 If a mapped element in a noun group is a head noun, the whole group is replaced by that element. If it is not, the remaining elements are included too.

Rule 3 All unmapped prepositional phrases, except the phrases containing keywords, are discarded.

Rule 4 An unmapped noun group is included if it is not a part of a prepositional phrase, after replaced by the semantic category of its head noun.

Rule 5. All verbs are replaced by their root forms, and all auxiliary verbs except the be-verb in passive form are discarded.

By applying rules 1 and 2, the noun groups "GRENADE" and "HOUSE" are replaced respectively by the concepts GRENADE and BUILDING. After applying rules 3 and 5, the prepositional phrase "IN SANBORJA NEIGHBORHOOD" is discarded, and the verb group "WERE HURLED" is replaced by "BE HURL."

The final form of the FP-structure acquired from the example sentence is shown in Fig. 5f. Note that the semantic constraints for the instrument and target slot are too specific now. These constraints are generalized to an appropriate level as more example sentences corresponding to this pattern are processed. Fig. 6 shows an example knowledge base generated by PALKA. The knowledge base consists of a set of FP-structures created by PALKA, the domain concept hierarchy, and the connections between them (the semantic constraints). The problem of how to establish correct connections for semantic constraints is presented in the next section as a generalization problem.

V. Generalization and Specialization

The goal of generalization is to determine an optimal level in a concept hierarchy for each element's semantic constraint in the phrasal pattern. Fig. 7 shows this problem. The semantic constraint of each element is given by a concept or a disjunction of concepts in the concept hierarchy. It determines the coverage of a phrasal pattern. If a constraint is given too specific, it may miss a sentence. If it is given too general, it may be matched incorrectly.

Since the semantic category of a newly created pattern is determined to be the *most specific* one, it should be generalized if possible. An acquired FP-structure is compared with existing ones for generalization. Whenever two FP-structures with similar phrasal patterns are generated, their semantic constraints are generalized. When an over-generalized pattern is found (incorrect matching), the corresponding semantic constraint is specialized.

In this section, two different approaches to generalization are presented—*single-step* and *incremental*. In both cases, inductive learning mechanism [21], [20] is applied to the modification of semantic constraints. For induction, the semantic constraint of a newly created pattern is used as a positive example, and the semantic constraint of an incorrectly matched pattern is used as a negative example.

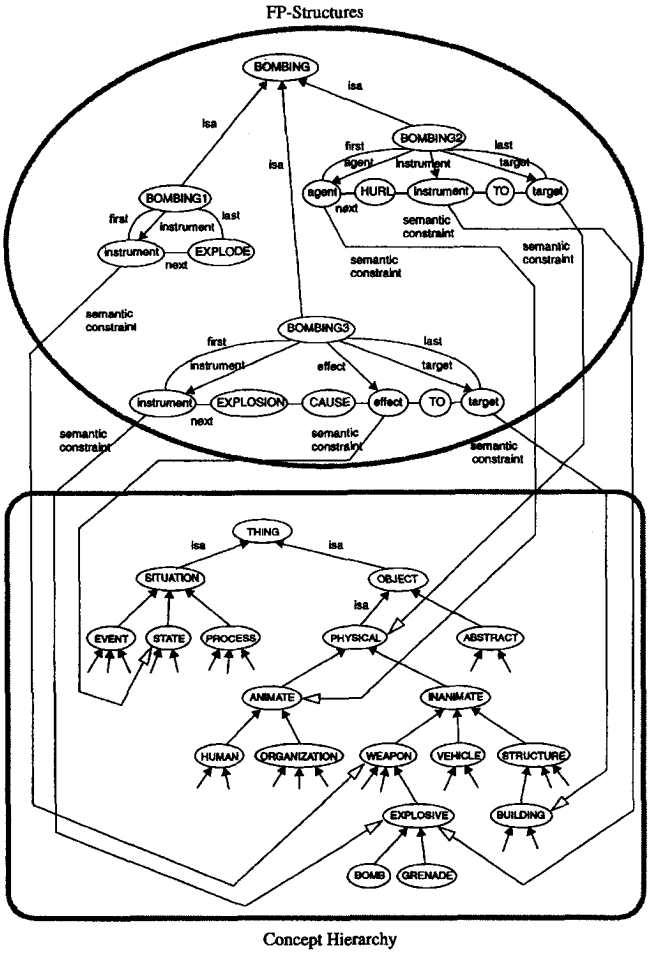


Fig. 6. An example of the knowledge base created by PALKA.

A. Single-Step Generalization

In the single-step approach, the algorithm keeps lists of example semantic constraints (concepts) for each element of an FP-structure during the acquisition process, and computes an appropriate semantic constraint at the end. When a positive example is encountered, a corresponding concept is added to the positive list (P). When a negative example is encountered, it is added to the negative list (N). The generalization is performed at the end of the acquisition process by using these lists.

Generalization is to detect the *most general concepts* among the *consistent semantic constraints*. The consistent semantic constraints are those which subsume the positive examples and do not subsume the negative examples. Let S be a set of concepts. Let $Sup(S)$ be a set of all subsumers of the concepts in S (the concepts in S and their ancestors), CS be a set of consistent semantic constraints, and $MGCS$ be a disjunction of most

general concepts among *CS*. Then, *Sup*(*P*) is the set of all hypotheses for consistent semantic constraints. Among them, *Sup*(*N*) must be eliminated since they produce incorrect interpretations. Therefore, the set *Sup*(*P*) − *Sup*(*N*) represents the set of consistent semantic constraints *CS*. The most general concepts are selected from *CS*, and combined with disjunctions to form a final semantic constraint. The most general concepts in a set are the concepts which do not have their subsumers in the set except themselves.

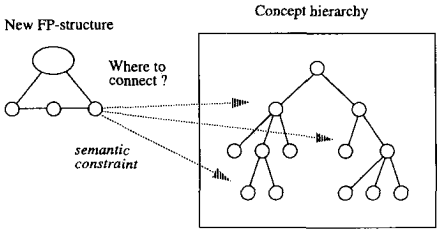


Figure 7. Determination of semantic constraints.

Fig. 8 shows an example of single-step generalization. P_1, P_2, and P_3 are positive examples, and N_1 is a negative example. In this example, the *Sup*(*P*) and the *Sup*(*N*) are:

$$Sup(P) = Sup(\{P_1, P_2, P_3\}) = \{P_1, P_2, P_3, A, B, C\}$$

$$Sup(N) = Sup(\{N_1\}) = \{N_1, B, C\}$$

Therefore, the consistent semantic constraint set is:

$$CS = Sup(P) - Sup(N) = \{P_1, P_2, P_3, A\}$$

Since A and P_3 are the most general concepts, the semantic constraint is determined as:

$$Result\ semantic\ constraint = MGCS = (A \vee P_3).$$

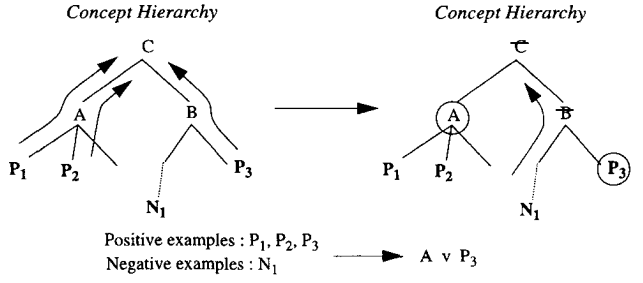


Fig. 8. Single-step generalization.

Fig. 9 shows an example of the determination of the semantic constraint for the pattern `BOMBING:[(instrument:X) EXPLODE]`. In this example, all the positive examples are under the concept `EXPLOSIVE`, and the negative examples are under `SITUATION`, `ANIMATE`, and `VEHICLE`. Therefore, the semantic constraint of the `instrument` slot is determined as the concept `WEAPON` after computing *Sup*(*P*) − *Sup*(*N*).

B. Incremental Generalization

When the size of a training set is large, changing semantic constraints during acquisition may speed up the acquisition process, since the number of parsing failure decreases. In the incremental approach, the algorithm modifies semantic constraints as it sees a new example sentence. Generalization and specialization are performed immediately when a new positive or negative example is encountered during the acquisition process. When a positive example is encountered, the semantic constraint is generalized, and when a negative example is encountered, the semantic constraint is specialized.

Let S be a set representing current semantic constraint, and *Inf*(*S*) be a set of all subsumees of the concepts in S (the concepts in S and their descendants). When a new positive example P_i is found, a new set of consistent semantic constraint *CS* is determined by computing *Sup*($S \cup \{P_i\}$) − *Sup*(*N*). The generalization of corresponding semantic constraint is performed by replacing current set S to a disjunction of the most general concepts, *MGCS*, among *CS*. When a new negative example N_i is found, a new set of consistent semantic constraint *CS* is determined by computing *Inf*(*S*) − *Sup*($N \cup \{N_i\}$). The specialization is performed by replacing current set S to a disjunction of the most general concepts, *MGCS*, among *CS*.

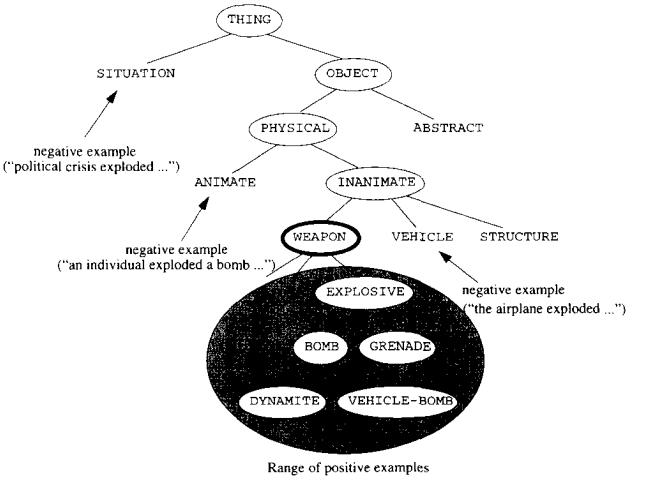


Fig. 9. Example of single-step generalization for the pattern `[(instrument:X) explode]`.

Fig. 10 shows examples of incremental generalization. In Fig. 10a, the current semantic constraint is A_1, and the semantic constraint of a new pattern (positive example) is P_i. In this example, the *Sup*($S \cup \{P_i\}$) and the *Sup*(*N*) are:

$$Sup(S \cup \{P_i\}) = Sup(\{A_1, P_i\}) = \{A_1, A_3, P_i, A, C\}$$

$$Sup(N) = \{N_1, B, C\}$$

Therefore, the new consistent constraint set is:

$$CS = Sup(S \cup \{P_i\}) - Sup(N) = \{A_1, A_3, P_i, A\}$$

Since A is the most general concept, the new semantic constraint is determined as:

Modified semantic constraint = $MGCS = A$.

In Fig. 10b, the current semantic constraint is A, and the semantic constraint of incorrect interpretation (negative example) is N_i. In this example, the $Inf(S)$ and $Sup(N \cup \{N_i\})$ are:

$$Inf(S) = \{A, A_1, A_2, A_3, N_i\}$$

$$Sup(N \cup \{N_i\}) = Sup(\{N_1, N_i\}) = \{N_1, N_i, A_3, A, B, C\}$$

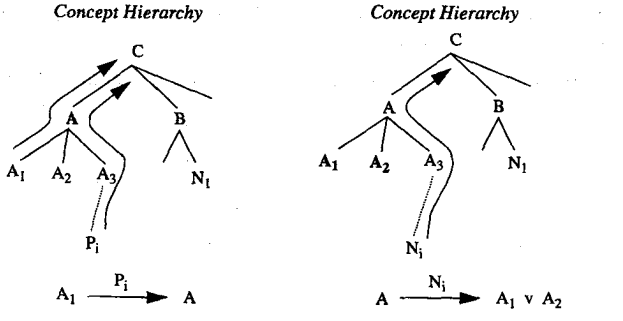
Therefore, the new consistent constraint set is:

$$CS = Inf(S) - Sup(N \cup \{N_i\}) = \{A_1, A_2\}$$

Since both A_1 and A_2 are the most general concepts, the semantic constraint is determined as:

$$Modified\ semantic\ constraint = MGCS = (A_1 \vee A_2).$$

In case there were no negative examples (no specializations occur), a semantic constraint is generalized to the highest level in the concept hierarchy according to above procedure. There are two possible ways to prevent this. First, one can select the *most specific* concept among those which subsume *all* the positive examples. Second, one can put limitations to the maximum generalizable level. For example, putting limitations to 2 or 3 levels below the top concept produces much better results even though they are still over-generalized in many cases.



(a) generalization using a positive example (b) specialization using a negative example

Fig. 10. Incremental generalization.

In our example in Fig. 5, the semantic constraint of the `instrument` slot is generalized from GRENADE to EXPLOSIVE, and the `target` slot is generalized from BUILDING to PHYSICAL-OBJECT by using the first method.

VI. Experimental Results

The PALKA prototype is implemented using C on a SUN workstation. A preliminary experiment has been performed with 500 MUC-4 texts and corresponding output templates to acquire FP-structures for the BOMBING frame and the KILLING frame. Each text contains approximately 14 sentences on average. The time spent for the acquisition was about five hours including user interaction and manual post-processing such as minor corrections on the phrasal pattern form. Actually the user interaction is very simple, because in most cases the user only needs to decide whether a phrase extracted by the system is relevant to BOMBING (or to KILLING) or not. Although only two frames and 500 texts were used in this experiment, the time to create knowledge base of linguistic patterns is significantly reduced compared with manual creation.

Fig. 11 shows the total number of sentences extracted for each frame from 500 texts, the number of new FP-structures acquired, the number of generalizations and specializations performed, the number of final FP-structures, and the average number of FP-structures created per one sentence. The result shows that 30% of the processed sentences produced an FP-structure for the BOMBING frame, but only 6% of the sentences produced an FP-structure for the KILLING frame. This shows that relatively smaller number of different expressions are used to describe the KILLING event in this domain. As an example, the pattern "`[(target:X) BE KILL]`" was found 97 times during the acquisition process. Several examples of collected sentences and the acquired FP-structures from them are shown in Fig. 12. Note that the two patterns `BOMBING: [(TARGET: HUMAN) BE KILL BY (EXPLOSION) OF (INSTRUMENT: BOMB)]` and `KILLING: [(TARGET: HUMAN) BE KILL]` can be activated simultaneously. In such cases, currently we put an inhibition relation from the first to the second pattern, so that whenever bombing-related information exists, the input is classified as a BOMBING event.

frame	sentences extracted	patterns acquired	generalizations	specializations	FP-structures created	average creation
BOMBING	220	89	22	5	67	30.5 %
KILLING	601	108	71	12	37	6.2 %

Fig. 11. Result of the acquisition from 500 MUC4 texts.

A basic assumption of our approach to pattern acquisition is that only a finite number of expressions is frequently used to describe a specific event in a limited domain. In other words, the patterns acquired from a relatively small number of sample texts can cover a much larger number of texts from the same domain. The growth of the knowledge base eventually becomes saturated. Figs. 13 and 14 show the changes of acquisition rates for the BOMBING and the KILLING frames while processing 500 texts. Since the acquisition rate varies depending on the order of sentences processed, 100 experiments were performed with random re-ordering of sentences and the results were averaged. Also, to observe the effect of generalization to the acquisition rate, the incremental algorithm was used in these experiments.

In Fig. 13, the acquisition rate decreases without being saturated yet. This is because 1) there were only 200 related sentence examples found, and 2) as mentioned earlier, a relatively large number of expressions are used to describe the BOMBING event. More example sentences are needed to reach to the saturation. In Fig. 14, the acquisition rate for the KILLING frame is almost saturated when 200 sentence examples are processed (only 2/3 of total processing is shown.) In both cases, it is clear

THE BOMB, MADE UP OF DYNAMITE AND A FUSE, EXPLODED JUST BEFORE DAWN IN THE HONDUTEL OFFICE IN SAN PEDRO SULA, 190 KM NORTH OF THIS CAPITAL.
BOMBING: [(INSTRUMENT: BOMB) EXPLODE IN (TARGET: BUILDING)]
POLICE SOURCES HAVE REPORTED THAT THE EXPLOSION CAUSED SERIOUS DAMAGE TO THE SALVADORAN EMBASSY BUILDING IN THE ELEGANT PROVIDENCIA NEIGHORHOOD IN EASTERN SANTIAGO.
BOMBING: [(EXPLOSION) CAUSE (EFFECT: DAMAGE) TO (TARGET: BUILDING)]
GUERRILLAS ATTACKED MERINO'S HOME IN SAN SALVADOR 5 DAYS AGO WITH EXPLOSIVES.
BOMBING: [(AGENT: HUMAN) ATTACK (TARGET: BUILDING) WITH (INSTRUMENT: EXPLOSIVE)]
TERRORISTS THREW A BOMB AT CIVIL DEFENSE MEMBERS IN NEJAPA , NORTH OF SAN SALVADOR, WHILE HARASSING THE PARAMILITARY GROUP'S OUTPOST.
BOMBING: [(AGENT: HUMAN) THROW (INSTRUMENT: BOMB) AT (TARGET: PHYSICAL)]
TWENTY-SIX PEOPLE, FOUR OF WHOM ARE POLICEMEN, WERE INJURED IN A BOMB EXPLOSION THAT OCCURRED BEFORE DAWN ON 30 OCTOBER.
BOMBING: [(TARGET: HUMAN) BE INJURE IN (INSTRUMENT: BOMB) (EXPLOSION)]
THE OFFICIAL REPORT POINTS OUT THAT THE POLICEMEN WERE KILLED BY THE EXPLOSION OF DYNAMITE THAT HAD BEEN PLACED IN THE VEHICLE TRANSPORTING THEM.
BOMBING: [(TARGET: HUMAN) BE KILL BY (EXPLOSION) OF (INSTRUMENT: BOMB)]
TEN TERRORISTS HURLED DYNAMITE STICKS AT U.S. EMBASSY FACILITIES IN THE MIRAFLORES DISTRICT, CAUSING SERIOUS DAMAGE BUT FORTUNATELY NO CASUALTIES.
BOMBING: [(AGENT: HUMAN) HURL (INSTRUMENT: BOMB) AT (TARGET: STRUCTURE)]
TWENTY EIGHT POLICEMEN HAVE BEEN KILLED IN THIS CITY OVER THE LAST TWO WEEKS.
KILLING: [(TARGET: HUMAN) BE KILL]
NEVERTHELESS, ACCORDING TO MILITARY REPORTS ISSUED EIGHT HOURS AFTER THE CLASHES IN THAT TOWN, THE ARMED FORCES KILLED 11 GUERRILLA MEMBERS AND SUSTAINED SIX CASUALTIES.
KILLING: [(AGENT: ANIMATE) KILL (TARGET: HUMAN)]
FOUR ALLEGED TERRORISTS, ALL WEARING SKI MASKS, MURDERED MAYOR ALFREDO CHAMORRO IN THE SMALL TOWN OF PALLANCHACRA.
KILLING: [(AGENT: HUMAN) MURDER (TARGET: HUMAN)]
YESTERDAY SHINING PATH TERRORISTS ARRIVED IN THE VILLAGE OF CHINCHIPE AND SHOT 16 PEASANTS WHO WERE MEMBERS OF THE PEASANT PATROLS.
KILLING: [(AGENT: HUMAN) SHOOT (TARGET: HUMAN)]

Fig. 12. Example sentences and corresponding patterns acquired.

that the acquisition rate strictly decreases, which means that the size of the knowledge base approaches the saturation point. Also, in both experiments, the effect of generalization on the acquisition rate is clearly shown. With generalization, the acquisition rate decreases more rapidly because some of the patterns are not actually created but generalized with others.

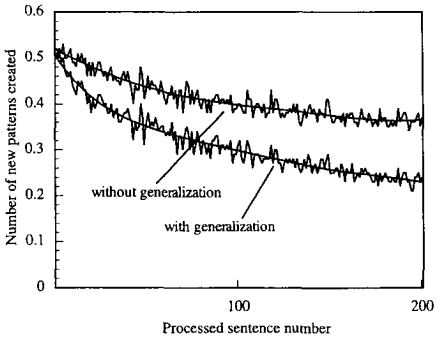


Figure 13. Average number of patterns created for BOMBING frame.

Figs. 15 and 16 show the improvements of parsing performances as more sentences are used to acquire patterns for each frame. For these experiments, various sizes of training sentence sets were used to generate patterns, and 200 sentences corresponding to each frame were randomly selected from MUC-4 texts as a test set. In Fig. 15, the recognition accuracy reached 60% using the patterns acquired from 200 training sentences. It shows that much more sample sentences are required to cover the BOMBING frame.

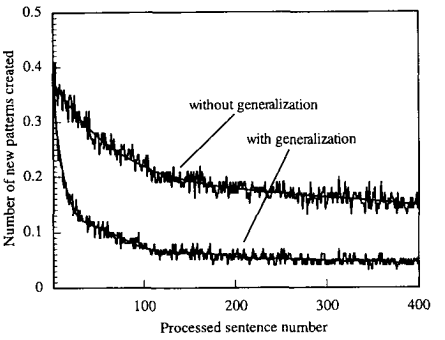


Figure 14. Average number of patterns created for KILLING frame.

In Fig. 16, the recognition accuracy for the KILLING frame reached 90% using the patterns from the same number of training sentences.

To observe the effect of generalization on the parsing performance, excluding the effect of the growth of the knowledge base, experiments with a single pattern were performed. For these experiments, 100 sentences which can be matched to the pattern "[(instrument:X) EXPLODE]" were collected from MUC-4 corpus, and tagged as positive or negative example. For example, as shown in Fig. 9, "*three dynamites exploded*" is a positive example, and "*the airplane exploded*" is a negative example for the "[(instrument:X) EXPLODE]" pattern. From the sentence set, n (training set size) sentences were randomly selected, and single-step generalization was performed on the semantic constraint X of the instrument slot. The generalization result was then used to parse the original 100 sentences to compute the recognition accuracy. Since the generalization result may vary according to the selection of the training sentences, 100 experiments were performed for each training set size, and the resulting recognition accuracies were averaged.

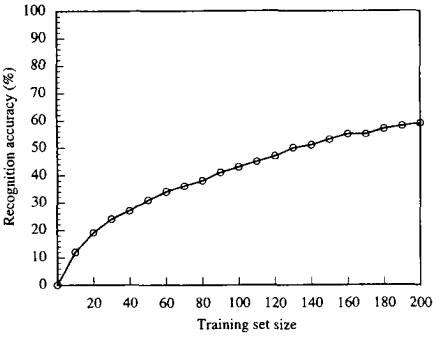


Fig. 15. Recognition accuracy vs. training set size for BOMBING frame.

The results in Figs. 17 and 18 show that the recognition accuracy is monotonically increasing when the generalization is performed with a larger training set. However, it is affected by:

1) how many negative examples are in the training sentence set, and
2) how many concepts are necessary to represent the optimal semantic constraint.

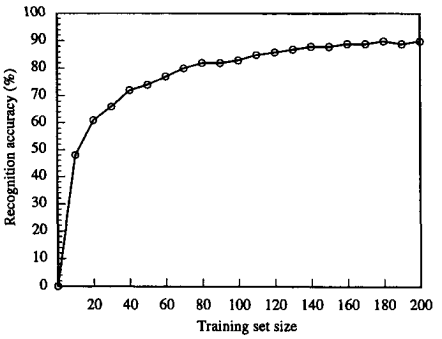


Fig. 16. Recognition accuracy vs. training set size for KILLING frame.

Fig. 17 shows the effect of generalization on the recognition accuracy for various ratios of negative examples in the sentence set. The result shows that if there exists a larger number of negative examples, the recognition accuracy improves more rapidly as the training set size increases. It also implies that it is better to select more negative examples as a training set.

Fig. 18 shows the effect of generalization on the recognition accuracy for the patterns with various semantic constraint group lengths. The semantic constraints of different patterns may have different lengths of description. For example, three group semantic constraint means that the optimal semantic constraint is represented by the disjunction of three concepts as $(A_1 \vee A_2 \vee A_3)$. In this experiment three patterns with different semantic constraint group lengths were used. The result shows that more complex descriptions of semantic constraints need more examples to produce a correct generalization.

In all cases, the average parsing recognition accuracy reached 90% with generalization using far less than 20% of entire set of texts.

VII. Related Work

The use of knowledge structures for interpretation of texts in a similar way to knowledge-based information extraction can be found in script-based approaches [29], [18], [16], direct memory access parsing [27], [15], and use of phrasal lexicon [1], [32], [34]. One common problem of these approaches is that the parser needs a large amount of domain-specific knowledge structures.

Previous work on lexical acquisition focused on the acquisition of the meaning of unknown words [7], [4], [10], the acquisition of collocative information using statistics [5], [30], and the construction of semantic dictionary [3], [24], [12], [28].

The goal of the acquisition of word meaning is to improve a parser by providing a flexibility or self-extending capability, and is different from the goal addressed in this paper. The statistical approach is efficient and easily implemented, but the collocative knowledge acquired by a statistical method usually does not provide semantics. The construction of semantic dictionary is most closely related to the goal and the approach of PALKA system described in this paper.

With a program called RINA, Zernik studied the acquisition of new word patterns and idioms in the form of *phrasal lexicon* [33], [34]. In his approach, the phrasal pattern is acquired from the user's utterance, and the meaning is learned though the context. Our approach is different from Zernik's in a sense that the context is not given to the system, and the meaning is provided explicitly as a form of database templates. Zernik's work provides a good modeling of language learning, but for practical application it is unrealistic since all the detailed context can not be provided in general.

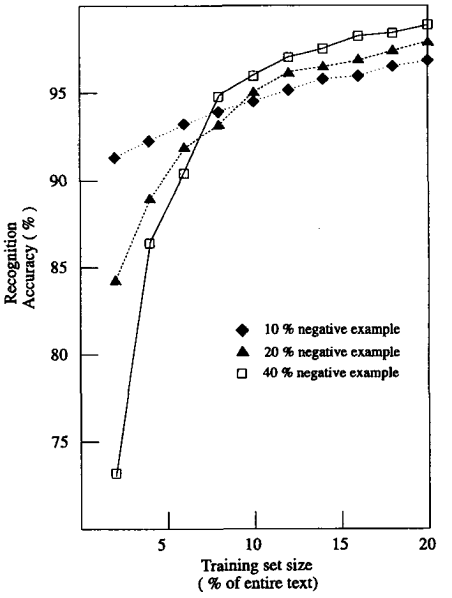


Fig. 17. Effect of generalization on the parsing performance with various percentage of negative examples: The parser processed sentences related to [(instrument:X) EXPLODE] pattern.

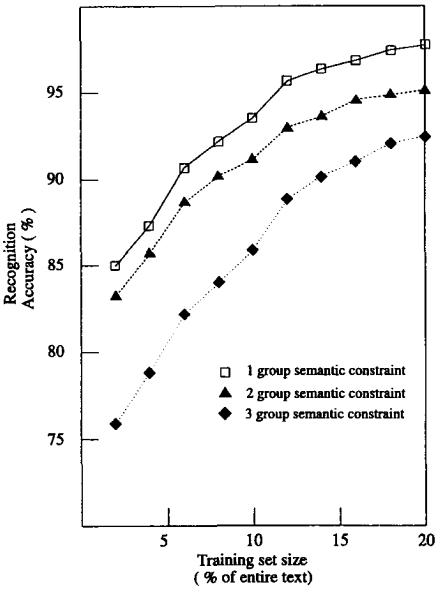


Figure 18. Effect of generalization on the parsing performance for the patterns with various semantic constraint group lengths.

A more practical approach to the acquisition of semantics can be found in the work of Velardi et al. [31], [24]. They presented a methodology for the extensive acquisition of a

case based semantic dictionary. Their system, PETRARCA, analyzes a large sample of sentences including a given word, and produces semantic dictionary entries for that word. Both approaches on acquisition in PALKA and PETRARCA are based on the analysis of large corpus. The differences are that PETRARCA acquires patterns for each word, but PALKA acquires patterns for each meaning frame. For a domain-specific application, acquiring patterns for target frames are more efficient since it can match phrases in the input text directly to the target meaning frame. Also, PETRARCA uses syntax to semantics mapping rules to associate each syntactic pattern to possible semantic interpretation, but PALKA uses the training corpus templates to map phrasal pattern to its semantic frame.

Jacobs and Rau [11] and Jacobs [12] have applied a statistical method to automatic pattern acquisition for knowledge based news categorization. The acquired pattern is called a *lexicosemantic pattern*, which is used for a knowledge based system NLDB that automatically assigns categories to news stories. An example of the pattern is: "*C1...announced...acquisition of...C2*," and the rule says "if in this pattern, *C1* and *C2* are company names, then it represents a *Takeover*, and the agent is *C1* and the target is *C2*." He used a training set of 11,500 news stories with human assigned categories for each story. In analyzing the training set, individual terms are weighted by a statistical means, and the heavily weighted terms are used as the building block of the patterns. The statistical method helps to find words and phrases that are good indicators of each category. His work shows a good example of how the statistical method can be successfully applied for the linguistic pattern acquisition.

One of the most closely related works is that of Riloff and Lehnert [28]. They developed a system called AutoSlog which constructs a domain-specific semantic dictionary for an information extraction system for the MUC-4 domain. Their information extraction system is based on a set of domain-specific dictionary entries called *concept nodes*. Concept nodes are triggered by specific lexical items relevant to the domain, and when activated, it acts as a case frame that picks up relevant information from the sentence. What a concept node tells is something like "if a word *attacked* is found and the class of its direct object is a *phys-target*, then fill the *target* slot with the direct object." The AutoSlog system also uses training texts and corresponding templates to automatically construct the concept nodes.

The AutoSlog system demonstrates its feasibility by significantly reducing the time to construct a domain-specific semantic dictionary. It has similar features with the PALKA system, and it uses the same knowledge sources. However, there are differences in representation and basic approach. A concept node is triggered by one lexical item and then rules are applied, but an FP-structure is recognized when a full phrase is matched—FP structures are more restricted forms of patterns. As a result, more patterns are necessarily compared to concept node representation to achieve certain level of recall. However, when sufficient amount of patterns are acquired, the precision with FP-structures would be much higher. One of the key issue in the AutoSlog approach is how to determine triggering words. The quality of heuristics used to find triggering words is critical to the overall performance of the acquisition system. In PALKA case, different keyword set affects only the rate of acquisition. Finally, AutoSlog checks templates first and then looks at the corresponding text to extract a sentence that contains a specific word. PALKA reads texts first, and refer templates to get mapping information. The template-first approach can be more efficient if there is only one related sentence, but text-first approach can be more productive when multiple sentences are related to certain information. For a given set of training corpus, text-first approach can produce much more relevant patterns.

The underlying assumption on the acquisition of linguistic pattern for a full phrase is that, in a limited domain, a relatively small number of expressions are frequently used to describe certain information. Similar idea can also be found in Lehman's work on adaptive parser [16]. The idea was that people are usually bounded in their use of language, and so a parser can be adaptive to certain user, avoiding definition of all meaningful forms of expression a priori. In our approach, we assume limited expressions in a limited domain, and the experimental results shows the feasibility of the approach.

VIII. Conclusions

In this paper, an automatic pattern acquisition system PALKA was presented. The system acquires linguistic patterns that can be used for knowledge-based information extraction from texts. A linguistic pattern is represented as a pair of a meaning frame and a phrasal pattern, which is called FP-structure. Phrasal patterns (syntactic information) are acquired from training texts, and mappings to the frame (semantic information) are acquired from corrsponding templates. The acquired patterns are generalized inductively. An experiment using PALKA with 500 MUC-4 domain texts demonstrates the feasibility of our acquisition approach.The time to construct a knowledge base of linguistic patterns is significantly reduced, and the acquisition rates saturated with relatively small set of training corpus.

One of the limitations of our approach is that we acquire patterns for verb-oriented clauses only. To provide more detailed information for each slot of the frame, patterns for noun phrases should also be acquired. In addition, in the current implementation the relations between different patterns are not investigated. Establishing such relations is desirable for both the effeciency of representation and the flexibility of interpretation. Also, when a large number of frames are involved in one application, there is the possibilty of many conflcts between multiple FP-structures.Currently, inhibition relations between conflict structures are created manually as mentioned in the last section. The detection and resolution of conflicts by the system would increase the usability. Finally, the use of other knowledge sources such as an on-line dictionary is being considered for a future work. This is important when

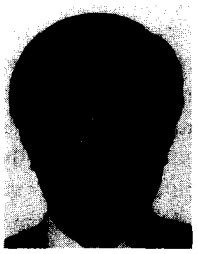
the output template of the training text is not available. Using an on-line dictionary or a tagged corpus can be feasible solution. These possibilites are being considered for further improvements of the system.

ACKNOWLEDGMENT

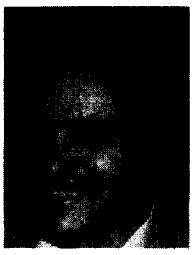
This research has been funded by National Science Foundation Grant No. MIP 9009109.

REFERENCES

[1] J.D. Becker, "The phrasal lexicon," *Bolt Beranek and Newman Inc.* Report no. 3081, 1975.
[2] R.C. Berwick, *The Acquisition of Syntactic Knowledge*, MIT Press, 1985.
[3] R.J. Brachman and J.G. Schmolze, "An overview of the KL-ONE knowledge representation system," *Cognitive Science*, vol. 9, 1985.
[4] J.G. Carbonell, "Towards a self-extending parser," *Proc. 17th Meeting Assoc. for Computational Linguistics*, 1979.
[5] K. Church and P. Hanks, "Word association norms, mutualinformation, and lexicography," *Proc. 28th Meeting Assoc. for Computational Linguistics*, 1990.
[6] G. DeJong, "Prediction and substantiation: A new approach to natural language processing," *Cognitive Science*, vol. 3, pp. 251-273, 1979.
[7] R.H. Granger, "FOUL-UP: A program that figures out meanings of words from context," *Proc. 5th Int'l Joint Conf. Artificial Intelligence*, 1977.
[8] A. Hauptmann, "From syntax to meaning in natural language processing," *Proc. 10th Nat'l Conf. Artificial Intelligence*, 1991.
[9] J.R. Hobbs, D. Appelt, M. Tyson, J. Bear, and D. Israel, "FASTUS: System summary," *Proc. Fourth Message Understanding Conf.*, 1992.
[10] P. Jacobs and U. Zernik, "Acquiring lexical knowledge from text: A case study," *Proc. Seventh Nat'l Conf. Artificial Intelligence*, 1988.
[11] P. Jacobs and L. Rau, "Scisor: Extracting information from on-line news," *Comm. ACM*, vol. 33, no. 11, 1990.
[12] P. Jacobs, "Using statistical methods to improve knowledge-based news categorization," *IEEE Expert*, Apr., 1993.
[13] J.-T. Kim and D. Moldovan, "Acquisition of semantic patterns for information extraction from corpora," *Proc. Ninth Conf. AI applications*, 1993.
[14] J.-T. Kim, *Semantic Knowledge Acquisition for Information Extraction from Texts on a Parallel Marker-Passing Computer*, PhD dissertation, Univ. of Southern California, Dept. of EE-Systems, 1993.
[15] H. Kitano, "ΦDM-Dialog: An experimental speech-to-speech dialog translation system," *Computer*, June, 1991.
[16] J.F. Lehman, *Adaptive Parsing: Self-Extending Natural Language Interface*, Kluwer Academic Publisher, 1992.
[17] W.G. Lehnert, C. Cardie, D. Fisher, J. McCarthy, E. Riloff, and S. Soderland, "Description of the CIRCUS system used for MUC-4," *Proc. Fourth Message Understanding Conf.*, 1992.
[18] W.G. Lehnert, "The role of scripts in understanding," D. Metzing, ed., *Frame Conceptions and Text Understanding*. Berlin: De Gruyter, pp. 79-95, 1980.
[19] *Strategies for Natural Language Processing*, W.G. Lehnert and M.H. Ringle, eds., Lawrence Erlbaum Associates, 1982.
[20] R. Michalski, "A theory and methodology of inductive learning," *Artificial Intelligence*, vol. 20, 1983.
[21] T. Mitchell, "Generalization as search," *Artificial Intelligence*, vol. 18, 1982.
[22] D, Moldovan, W. Lee, C. Lin, and M. Chung, "SNAP: Parallel processing applied to AI," *Computer*, June, 1992.
[23] D. Moldovan, S. Cha, M. Chung, K. Hendrickson, J. Kim, and S. Kowalski, "Description of the SNAP system used for MUC-4," *Proc. Fourth Message Understanding Conf.*, 1992.
[24] M.T. Pazienza and P. Velardi, "Methods for extracting knowledge from corpora," *Proc. Fifth Ann. Workshop Conceptual Structures*, 1990.
[25] *Proc. Fourth Message Understanding Conf.*, Morgan Kaufmann, 1992.
[26] J. Pustejovsky, "The generative lexicon," *Computational Linguistics*, vol. 17, no. 4, pp. 409-441, 1991.
[27] C.K. Riesbeck and C.E. Martin, "Direct memory access parsing," *Report 354*, Dept. of Computer Science, Yale Univ., 1985.
[28] E. Riloff and W. Lehnert, "Automated dictionary construction for information extraction from text," *Proc. Ninth Conf. AI Applications*, 1993.
[29] R. Shank and R. Abelson, *Scripts, Plans, Goals, and Understanding*, Lawrence Erlbaum Associates, N.J., 1977.
[30] F. Smadja, "Macrocoding the lexicon with co-occurrence knowledge," *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon*, Lawrence Erlbaum Associates, N.J., 1991.
[31] P. Velardi, M.T. Pazienza, and S. Magrini, "Acquisition of semantic patterns from a natural corpus of texts," *ACM SIGART Newsletter*, no. 108, Apr. 1989.
[32] R. Wilenski, Y. Arens, and D. Chin, "Talking to Unix in English: An overview of UC," *Comm. ACM*, vol. 27, no. 6, 1984.
[33] U. Zernik and M.G. Dyer, "The self-extending phrasal lexicon," *Computational Linguistics*, vol. 13, nos. 3-4, 1987.
[34] U. Zernik, *Strategies in Language Acquisitions: Learning Phrases from Examples in Context*, PhD Dissertation, Computer Science Dept., UCLA, 1987.
[35] *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon*, U. Zernik, ed., Lawrence Erlbaum Associates, N.J., 1991.



Jun-Tae Kim received the BS degree from the Department. of Control and Instrumentation, Seoul National University in 1986, and the MS and PhD degrees in electrical engineering and computer engineering from the University of Southern California in 1990 and 1993, respectively. He was a postdoctoral research associate in the Department of Computer Science and Engineering at Southern Methodist University in 1993 and 1994. Currently, Dr. Kim is an assistant professor in the Department of Computer Engineering, Dongguk University in Seoul, Korea. His research interests include artificial intelligence, information retrieval, natural language processing, and parallel and distributed processing.



Dan I. Moldovan (S'76-M'78) received his MS and PhD degrees in electrical engineering and computer science from Columbia University, New York, in 1974 and 1978, respectively. He was a member of the technical staff at Bell Laboratories from 1976 to 1979, an assistant professor of electrical engineering at Colorado State University from 1979 to 1981, and assistant, and later associate professor of computer engineering at the University of Southern California from 1981 to 1993. Currently he is a professor and chairman of the Department of Computer Science and Engineering at Southern Methodist University and serves as director of the Parallel Computers Research Laboratory. He took a one-year sabbatical leave from 1987 through 1988 to work at the National Science Foundation in Washington, D.C., as program director for experimental systems in the Division of Microelectronics and Information Processing Systems.

Dr. Moldovan's primary research interests are in the fields of parallel processing and artificial intelligence, in which he has published more than 100 papers. He is the author of a textbook *Parallel Processing: From Applications to Systems*, published by Morgan Kaufmann in 1993.