From: FLAIRS-01 Proceedings. Copyright © 2001, AAAI (www.aaai.org). All rights reserved.

Automatic Discovery of Linguistic Patterns for Information Extraction

Sanda M. Harabagiu, Mihai Surdeanu and Paul Morărescu
Department of Computer Science and Engineering
Southern Methodist University
Dallas, TX 75275-0122
{sanda,mihai,paulm}@seas.smu.edu

Abstract

Information Extraction (IE) systems typically rely on extraction patterns encoding domain-specific knowledge. When matched against natural language texts, these patterns recognize with high accuracy information relevant to the extraction task. Adapting an IE system to a new extraction scenario entails devising a new collection of extraction patterns - a time-consuming and expensive process. To overcome this obstacle, we have implemented in CICERO, our IE system, a pattern acquisition mechanism that combines lexico-semantic knowledge available from WordNet with syntactic information collected from training corpora. The open-domain nature of the knowledge encoded in WordNet grants portability of our approach across multiple extraction domains.

Introduction

The purpose in Information Extraction (IE) systems is to extract information relevant to pre-specified events and entities of interest from natural language texts. The identification of the relevant text snippets is typically based on a collection of extraction patterns that encode typical relations as well as a semantic lexicon characteristic for the domain of interest. Whenever a new extraction task is defined, a new set of extraction patterns need to be available. Since the linguistic information encoded in these patterns requires a non-trivial level of expertise, some of the best performing current IE systems (e.g. (Appelt et al.1993)) rely on human-crafted linguistic patterns for extraction.

Since portability and performance are recognized as the two major obstacles in widespread use of the IE technology, methods of automatic acquisition of IE patterns have been of interest since the the early evaluations of the Message Understanding Conferences (MUCs). Most systems generated patterns automatically if special resources were available: (a) texts annotated with domains-specific tags, e.g. AUTOSLOG, CRYSTAL, RAPIER, SRV or WHISK or; (b) or manually defined semantic frames, e.g. PALKA, LIEP.

Recently, three new different approaches have been developed. First, AUTOSLOG-TS (Riloff and Jones 1999) learns extraction patterns and their corresponding lexical dictionaries from unlabeled texts, by using some patterns seeds and bootstrapping new patterns and new lexical entries from unseen examples. The second approach, described in (Yangarber et al.2000) uses an iterative relaxation process by starting with several relevant patterns that split a training corpus into relevant and non-relevant documents. All clauses in the relevant documents are assumed candidate patterns, but only those that can be generalized into extraction patterns that have high distribution in the relevant texts and low distribution in the non-relevant documents are added to the resulting collection of patterns. The third approach described in (Bagga et al.1997) uses WordNet (Miller 1995) to learn general extraction patterns. Initial patterns are entered by the user, who also accounts for the semantic sense of each word. Generalizations of these patterns are made possible by subsumption operators working along WordNet noun hierarchies.

In this paper we present a new method of acquiring extraction patterns by combining the lexico-semantic knowledge available from WordNet with syntactic information derived from training texts. Our acquisition procedure, implemented in CICERO, our IE system, does not require semantic disambiguation of words and moreover, generates patterns in a specialized language, called RULESPEC, capable of translating a regular expression into a finite state machine, using an optimization compiler. A similar specification language, called FASTSPEC is used in the FASTUS system, to enable the definition of hand-crafted patterns. Expressing extraction patterns as rules that are translated into finite-state automata is an ideal way of encoding domain knowledge in IE systems, since finite-state transducers are the dominant technology in the IE field. Finite-state automata, frequently cascaded, are capable of recognizing linguistic constructs ranging from low-level syntactic expressions (e.g. noun groups and verb groups) to higher-level, domain relevant clausal patterns.

In our method, the extensive semantic net encoded

Copyright © 2001, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

in WordNet (*www.cogsci.princeton.edu/~wn*) is mined for concepts and lexico-semantic relations relevant to a domain. The resulting concepts and their interrelations are validated by domain- relevant corpora, enabling the discovery of their syntactic contexts. Our novel method generates linguistic patterns for a domain as ***production rules*** induced when using the principle of maximal coverage of collocating domain concepts.

The rest of the paper is organized as follows. Section 2 presents RuleSpec, our pattern specification language. Section 3 details the representation of domain semantic spaces and the knowledge mining techniques used to generate the extraction patterns. Section 4 discusses the experimental results. Section 5 summarizes the conclusions.

The Pattern Specification System

To accommodate the representation of linguistic patterns, Cicero includes a ***pattern specification system***. The pattern specification system was designed for high efficiency of the extraction task as well for lightweight definition of extraction patterns. Patterns are economically and efficiently represented using a ***pattern specification language***, called RuleSpec, which is based on extended regular expressions. In our approach RuleSpec files are compiled into optimized finite state automata (FSA) using a ***pattern specification compiler*** (PSC). The compiler outputs C++ code which is later compiled to binary code for efficient execution. The FSA matching algorithm is implemented in the ***runtime system***, which is linked to the generated FSA code as a library.

Pattern specification language. Each pattern is represented as an extended regular expression associated with a set of actions to be executed on successful match. In order to improve the utilization efficiency, Cicero supports pattern macros, that allow the reutilization of common pattern information. The best example is the reutilization of Subject-Verb-Object (SVO) macros. For different actions, or different domains, an SVO pattern will have different subject/verb/object instantiations. Nevertheless, the syntactic structure is the same. In Cicero, the syntactic structure is defined by ***pattern macros***. For each specific action, macros are instantiated with specific information. Next we show the macro for one syntactic structures: active SVO. Similar structures are built for passive and relative structures.

```
EVENT_PHRASE ==>
{ #NG[@subj]:1 { COMPL }? }?
  #VG[in.active,in.type==TYPE_BASIC,(@head)]:2
  { #NG[@obj]:3 }?
  { #P[@prep1] #NG[@pobj1]:4 }?;
  out.item = in(2).item;
  out.svoPattern = @label;
  @semantics;;
```

The macros are represented through a macro grammar, a collection of regular expressions stripped of any semantic information. Macro variables (symbols starting with "@") are used to provide a generic representation for semantic information. When macros are instantiated, variables are replaced with semantic information, expressed either as boolean conditions (if variables are part of recognizers), or regular C/C++ code (if variables are part of actions). Next we illustrate the APPOINT pattern which instantiates the active and passive macros.

```
expand {ActiveSVO PassiveSVO} with {
  @label = "APPOINT"
  @subj = in.isCompany
  @head = $APPOINT_WORD
  @obj = in.isPerson
  @prep1 = "as" | "to"
  @pobj1 = in.isPosition
  @semantics =
   cout << "appoint found.<< endl}
```

The example above would match constructs such as: "[GMC] [recently appointed] [Mr. Baker] [as] [president of Buick]". The ActiveSVO macro instantiated by the APPOINT pattern matches the subject over "[GMC]", the verb over "[recently appointed]", the main object over "[Mr. Baker]" and the secondary object over [president of Buick]. A passive SVO macro instantiated by the APPOINT pattern can match constructs such as: "[Mr. Baker] [was appointed] [as] [president] [by] [IBM]". Upon successful matching, the actions (here identified by the variable "??semantics") are executed.

Pattern specification compiler. The compiler generates the corresponding FSA for each RuleSpec grammar. Due to the ambiguities of natural language grammars, the generated FSA is non-deterministic. To reduce search time, the FSA is organized such that the first match returned is the longest match.

Runtime system. The runtime system is implemented following the same efficiency principle. The match algorithm has two phases: (a) *search*, which performs a search on the FSA represented as a push-down automata, and (b) *execution*, which executes the actions associated with the matched patterns. Because actions are not executed during the search phase, action side-effects do not have to be backed-up on the search stack. This yields a fast and light match process.

Domain Knowledge for IE

Our method of acquiring linguistic patterns was devised as a three step process:
Step 1. First, we create a semantic space that models the domain via WordNet concepts and relevant connections between them. Building a semantic space for a domain of interest provides means for (a) finding all linguistic patterns that cover the relevant textual information in documents and moreover (b) enables the interpretation of the interrelations between different relevant textual expressions from the same document or even across documents (i.e. document threading).

A semantic space corresponding to a certain domain contributes to the resolution of some of the problems that still hinder the performance of current IE systems: [1] event recognition (also known as template merging),

[2] the inference of implicit arguments, and
[3] the interpretation of non-literal textual expressions of relevance to a given domain.

Step 2. In the second phase, we scan the phrasal parses of texts from the `MUC` corpora for collocating domain concepts that are connected in the domain semantic space. Production rules are induced using the principle of maximal coverage of collocating concepts. The phrasal parser implemented in CICERO generates correct syntactic links emerging from domain concepts, and thus enables the derivation of linguistic patterns.

Step3. Finally, the patterns are classified against the WordNet hierarchies and only the most general linguistic domain patterns are retained. The results matched all the linguistic patterns hand-crafted previously for `CICERO` and produced several new patterns.

Similar to other knowledge-based tasks, this method of automatically acquiring domain linguistic patterns has had a large start-up effort. This included the need for an unsupervised method of encoding morphological links in WordNet as well as heuristics for reformatting the information from conceptual definitions. However, the high performance of this linguistic pattern-acquisition method indicates that it is a valuable tool for building ontologies of domain patterns, and extremely useful for indexing digital libraries as well.

Experiments

In our experiments we have found that if we start with a predefined set of linguistic rules, expressed as subject-verb-object (SVO) patterns, WordNet-based observations help enhance the set with additional patterns. Thus we noticed that novel connections between domain concepts do not result directly from available WordNet relations, but they are rather combinations of WordNet relations mixed with :

- (i) *lexico-semantic relations* implicit in the conceptual definitions (known as *glosses* in WordNet);
- (ii) *morphologically cued relations*;
- (iii) *concept-identity relations* established between a synset and all its usages in the gloss of other synsets; and
- (iv) *collocational relations*, connecting multi-word synset elements with the synsets of each word[1](e.g. synset {*take office*} has collocating relations to synsets {*fill, take*} and {*office, position, post, berth, spot, place, situation*}).

Our general conclusion after these experiments was that although WordNet displays a magnitude of linguistic information, acquiring domain knowledge for IE involves a complex search and the derivation of several additional forms of connections among concepts. For example, a new pattern for the `MUC-6` domain was found due to the connection between the trigger words *take* and *helm* (as a form of position of leadership). The representation of this new pattern is:

[*Subject*=Person][*Trigger-phrase*=`take the helm`]
[*Preposition*={at|of}] [*Preposition-object*=Organization}

This pattern extends the general SVO structure of the linguistic patterns implemented in IE systems, allowing more complex triggers. The acquisition of this pattern is derived by the WordNet relations between synsets {*assume, take on, take over*} and {*position, office, place, post, slot*}. Synset {*take office*} can be reached via:
(a) *concept-identity* relations, since the concepts *assume* and *office* are both used in the gloss of *take office*.
(b) a *collocational* relation, generated by the same sense of *office* in the synsets {*position, office, place, post, slot*} and {*take office*}.
Moreover, synset {*take office*} is used to define synset {*accede to, enter upon*}, a hyponym of {*succeed, come after, follow*}. Therefore we infer that a succession event can be expressed also by any collocation of the verb *take* (with the semantic sense from *take office*) and
(1) any element from the synset {*position, office, place, post, slot*};
(2) any of its hypernyms; or
(3) any synset defined[2] in the hierarchy of {*position, office, place, post, slot*}.
A synset that pertains to case (c) is {*helm*}, defined as *(position of leadership)*. Therefore, *[take the helm]* is induced as a novel trigger phrase.

Learning new patterns involves not only deriving new trigger words, but also their satellites (e.g. *Subject, Object*, or *Prepositional Object*). Collecting all the collocations of trigger words from a corpus is not sufficient for establishing meaningful connections in a domain. Thus, we need to validate the connections of the trigger concepts in a semantic space that models the domain of interest. For example, in finding the satellites of trigger-phrase *take the helm*, we searched for connections to *management-position* or *manager*. WordNet provides a connection between synsets {*helm*} and {*manager, director, managing director*}. The gloss of {*helm*} defines it as a *position* having the attribute of *leadership*. In turn concept *leadership* is a nominalization of the verb *to lead*. Another nominalization of the verb *lead* is *leader*, the subject of the action. Because synset {*leader*} is a hypernym of *manager*, a semantic connection between *helm* and *manager* is found. This indicates possible pattern matchings of *taking the helm* and any position of management.

We conclude that at the core of these experiments is the construction of domain semantic spaces, encoding several additional forms of relations, detailed in the following Section .

Step 1: Building semantic spaces for IE

Domain knowledge is acquired in the form of a semantic space formalized as a triplet

< concepts, connections, contexts >

[1] These relations implement the assumptions of compositional semantics.

[2] (i.e. having the genus of the gloss)

The set of **concepts** is represented by a collection of WordNet synsets found relevant to a given domain. The **connections**, spanning the semantic space concepts , model several forms of relationships:
[1] *Thematic connections* between concepts in a certain *context*. Thematic relations are derived from (a) lexico-semantic relations encountered in the gloss definitions; (b) morphological relations; and (c) interpretation of WordNet paths, glosses and morphological relations.
[2] *Subsumption connections*, generated either from original WordNet *IS-A* relations or from the interpretation of gloss geni.
[3] *Contextual connections* spanning the context objects and describing the possible relationships between them. We distinguish four types of contextual connections: ***entail*** and ***antonym*** connections, encoded in WordNet and ***compose*** and ***similar*** connections. We assume that a contextual object ***entails*** another one if all propositions true in the former will remain true in the latter. A context is ***antonymous*** to another if any of the propositions that hold in its space will not be true in the latter, and vice versa. Assuming that a proposition P_1 holds in context C_1 and a proposition P_2 holds in a context C_2, if there is a context C_0 in which both P_1 and P_2 hold, we say that there are ***compose*** connections from C_0 to both C_1 and C_2. Finally, when all propositions holding in a context C_1 hold also in C_2 (and vice versa), we establish a ***similar*** connection between C_1 and C_2.
Contexts are semantic objects encompassing : (a) concepts, (b) thematic and subsumption connections and (c) conditions that enable their inter-connections. Contexts model various ways of expressing information regarding events of interest in a given domain, and are a means of capturing the relationship between these events. For example, in the MUC-6 domain, the event of appointing a person to a new managerial position can be expressed by stating that the respective person has been promoted, or by announcing the person's new position, or by stating that the person became or is the executive in that position, or by stating that the person stepped into the new position. Since promoting (or becoming, stepping in or being) cannot always be viewed as a form of appointing, we consider ***entailment*** (or implication) connections between these events (modeled by different contexts).

When the domain of interest is defined as a sequence of relevant keywords or collocations text, with a structured sequence of words, (e.g. a complex nominal =***management succession*** in the case of MUC-6 or ***aircraft crashes*** for the dry-run of MUC-7)[3], the semantic space is acquired by the following algorithm:

Algorithm 1 (Builds semantic spaces)
Input: Sequence of words: $\{word_k\}$
Output: *<concepts, connections, contexts>*[4]

```
For every keyword defining the domain
 1. Collect all the morphological derivations
    of the keyword.
 2. Collect all synsets containing either a
    keyword or a collocation containing it.
 3. For each keyword k group all synsets
    related to it in $Syn(k)=\{S_k^1,..,S_k^j\}$.
 4. Given any pair of synsets $S_k^i$ and $S_m^j$
    corresponding to different keywords, we measure
    the semantic density of common noun and verb
    words in their hierarchies and glosses.
 5. Select the top half of the most densely connected
    pairs of synsets.
 6. Expand each selected pair along WordNet relations.
 7. Goto 3 unless all there is at least one synset
    each $Syn(k)$ connectionfor each keyword.
 8.1.   Discard all unconnected synsets.
 8.2.   Derive themes and subsumptions
 9. Specialize every context in the domain by
    9.1.   taking all hyponyms having
      subjects, objects or prepositional relations to
      common concepts
    9.2.   retrieving concepts that
      have the common concepts in their glosses
 10. Generalize all classes of events or
    entities in every context.
 11. Derive contextual connections.
```

Step 2: Text Mining

The methodology of creating a semantic space needs to be validated by a corpus-based empirical test. Table 1 lists the number of domain concepts, contextual objects, subsumption and thematic connections as well as the number of contextual connections obtained for the MUC-6 domain (e.g. ***management succession***).

Three problems arise when using the domain contextual objects to devise linguistic patterns:

1. As WordNet synsets are not encoded for a specific domain, many of the synsets gathered in the contextual objects contain entries that are not used in the respective domain.
2. Thematic relations were induced only from the conceptual glosses. Text describing events from a given domain generally display far more thematic relations than those from the definitions of concepts. These relations should be incorporated in the linguistic rules.
3. The degree of generality of concepts from every context has to be done in harmony with the generality of the concepts employed in real world texts for the domain of interest.

These problems are resolved as a by-product of a corpus-based procedure that acquires the domain linguistic patterns.

Algorithm 2 (Finds domain linguistic patterns)
Input: Contexts from the semantic space, Text corpora.
Output: Linguistic rules.

[3] The methodology can be extended easily when the domain is defined by a list of keywords or by several free text paragraphs, as was the case for TREC-6 or TREC-7

[4] A semantic space.

Nr. words	Nr. concepts	Nr. contextual objects	Nr. subsumption connections	Nr. thematic connections	Nr. contextual connections
245	81	20	45	104	32

Table 1: Cardinality of the semantic space built for MUC-6

```
1. For every Contextual object from the semantic space of
   the domain
2.  For every $\mathcal{V}$ verb or Act-nominalization
3.    Scan all texts and gather the phrasal context
      where $\mathcal{V}$ or any concept subsumed by it occur
4.    If (there is a phrasal context
      where a new thematic role for $\mathcal{V}$ exists)
5.      If (all the other roles of $\mathcal{V}$ are encountered
        in that phrasal context as well)
6.        Create a new contextual object for $\mathcal{V}$.
7.      If (the filler of the new role subsumes
        any of the existing fillers)
8.        Add the new prepositional-role for that filler.
9. For every Contextual object from the semantic space
      of the domain
10.   Find the most general filler.
11.   Find the synset elements that were retrieved from
        phrasal contexts.
12.   Create a linguistic rule and mark its label
        with RULE-label.
13.   Mark the verbal concepts encountered in text
        with the RULE-label-$\mathcal{V}$ attribute.
14.   Mark the thematic filler words encountered in
        text with the RULE-label<theme> attribute.
15.   Translate the themes in FASTSPEC.
```

Step 3: Generalization of patterns

Given any pattern mined from WordNet and validated in the test texts, a semantic class generalization of its slots against WordNet hierarchies is performed. Table 2 illustrates the results of this procedure applied to the MUC-6 domain. The MUC-6 corpus was preprocessed with the CICERO phrasal parser. We have devised only one novel context (and consequently a new rule) since promote, a subsumer of appoint was found to have a supplementary theme provided by the previous position of the promoted executive. Moreover, we have automatically produced all the linguistic patterns that were manually crafted for CICERO and came up with several novel linguistic rules, corresponding to the *Step-down* and *Take-the-helm* contexts. In addition, by combining the knowledge from WordNet with the experience of building CICERO we have devised a methodology that creates rapidly and easily linguistic patterns for any new domain. The existence of the semantic space (and its contextual connections) provides with the relational semantics between the events extracted from texts, and makes possible event correference (or merging) and threading across documents. We contemplate the usage of the semantic space for information retrieval from the Internet and to the task of summarization.

Nr. rules	Nr. thematic connection	Nr. words encountered in texts
21	209	193

Table 2: Attributes of linguistic rules derived for MUC-6

Conclusions

This paper describes an implementation of a system that acquires domain linguistics rules for IE with the use of the WordNet lexico-semantic database. Two different algorithms that participate in the acquisition are presented. The first algorithm generates a semantic space for each domain. This semantic space is an important resource that can be used for other aspects of the IE process as well. For example, the event merging, i.e. the process of recognizing the same event in a text, could greately benefit from such a resource. The second algorithm collects syntactic information characterizing collocating concepts from the semantic space, recognized in texts. After generalizing the semantic class of lexical fillers the extraction patterns are produced in the RULESPEC format.

References

Douglas E. Appelt, Jerry R. Hobbs, John Bear, David Israel, Megumi Kameyama, and Mabry Tyson. The SRI MUC-5 JV-FASTUS Information Extraction System. In *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, 1993.

Amit Bagga, Joyce Yue Chai and Alan Biermann. The Role of WordNet in The Creation of a Trainable Message Undestanding System. In *Proceedings of the 14th Conference on Artificial Intelligence (AAAI/IAAI-97)*, 941–948.

Mary Elaine Califf and Raymond J. Mooney. Relational Learning of Pattern-Match Rules for Information Extraction. In *Proceedings of the ACL Workshop on Natural Language Learning*, pages 9–15, 1997.

Geroge A. Miller. WordNet - A Lexical Database for English. In *Communcations of the ACM*, Vol 38, No 11:39–41,1995.

Ellen Riloff and Rosie Jones. Learning dcitionaries for Information Extraction by multi-level bootstrapping. In *Proceedings of the 16th National Conference on Artificial Intelligence (AAAI-99)*.

Roman Yangarber, Ralph Grishman, Pasi Tapanainen, Silja Huttunen. Unsupervised discovery of scenario-level patterns for information extraction. In *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP-2000)*, pages 282–289.