Contextual Rules for Text Analysis

Dina Wonsever, Jean-Luc Minel

Second International Conference on Intelligent Text Processing and Computational Linguistics

publié dans
Lecture Notes in Intelligence Artificial

Contextual Rules for Text Analysis

Dina WONSEVER [1] Jean-Luc MINEL [2]

[1] Instituto de Computación, Facultad de Ingeniería, Universidad de la República
Herrera y Reissig 565 11300 MONTEVIDEO URUGUAY
Tel.: 598 2 711 42 44/103 Fax: 5982 711 04 69
wonsever@fing.edu.uy

[2] CAMS Équipe LALIC UMR 8557 - CNRS / EHESS / Université Paris Sorbonne
96, Bd Raspail 75006 PARIS FRANCE
Tel : 01 44 39 89 50 Télécopie: 01 44 39 89 51
minel@msh-paris.fr

Abstract. In this paper we describe a rule-based formalism for the analysis and labelling of texts segments. The rules are contextual rewriting rules with a restricted form of negation. They allow to underspecify text segments not considered relevant to a given task and to base decisions upon context. A parser for these rules is presented and consistence and completeness issues are discussed. Some results of an implementation of this parser with a set of rules oriented to the segmentation of texts in propositions are shown.

1 Introduction

In this paper we describe a rule-based formalism for the analysis and consequent labelling of texts segments. The rules are contextual: they allow identifying a portion of text that has to be labelled as a function of itself and, eventually, of a portion of text that precedes it (left context) and/or of a portion of text that follows it (right context). A portion of text satisfies the condition that allows the labelling by means of a rule if it includes certain elements (words, punctuation marks, portions of texts previously labelled) in a specific order. The rule determines what are the elements that have to be present and the order between them. These elements have not always to be contiguous. Intercalated between them there can be other portions of text for which the rule only states the maximum size and a list of elements that these portions of text must not contain.

In section 2 we describe the background that causes the introduction of this kind of rules. They were thought as a tool for specific goal-directed text processing tasks, mainly in Contextual Exploration systems. In section 3 we present in detail rules main features, the underlying text model, some examples and the results of the analysis obtained from a running system that fully implements them.

In section 4 a deductive system for an inactive right-corner chart parser for contextual rules is presented. A search strategy for ensuring consistency and completeness is proposed and all these concepts are briefly discussed.

Finally, some lines for future work are mentioned.

2 Background

Text analysis and information extraction from texts is a widespread need in our days. The increasing availability of very large quantities of textual information poses a challenge for new tools and capabilities from the field of natural language processing. Many of the requirements do not require a complete analysis of texts. They are goal-oriented: there is an objective to fulfil and textual information sources need to be analysed only in terms of this objective. While sometimes the process of text analysis needs to incorporate domain dependent knowledge in order to accomplish the desired task, it has also shown to be highly productive to analyse texts in a goal-oriented way on the basis of general domain-independent linguistic knowledge.

Information Extraction systems as seen in MUC Conferences [12] are an example of the first kind of goal-oriented systems. In this case the goal is the extraction of a particular kind of information from texts. MUC-3 focused on information about terrorist incidents; MUC-5 on information about enterprise joint ventures.

The Contextual Exploration Methodology developed by Lalic group [8], [9] is a linguistically motivated research line with relevant achievements that represents the second mentioned kind of goal-oriented text processing systems.

The rules system presented in this paper is a tool for the description and analysis of texts under the goal-oriented hypothesis. Only information relevant to a specific task needs to be described and decisions can be made based on relations between relevant items and contextual ones (i.e., items that are present in the co-text). In next section we describe in more detail the contextual exploration methodology, which greatly inspired our rules.

Contextual Exploration Methodology, proposed and developed by Jean-Pierre Desclés and his team, is based upon the observation that several textual processing tasks ([3], [4]), such as knowledge extraction or automatic summarising, may be solved by analysing exclusively units in texts, provided that their linguistic context is taken into account. For example, cognitive observations of professional summarisers have shown that they use textual, structural, thematic and lexical markers in their search strategies. Furthermore, various linguistic works on text analysis ([1], [6], [23], [24]) have shown the interest in identifying and locating linguistic markers and their combinations in order to lend meaning to textual units. Textual processing calls for identifying and studying the semantics of textual categories involved in texts that are independent of the text domain (medical, economical, technical, etc.). As a result, causality ([10], [14], [15]), definition [5], thematic announcement [4] have been studied under this perspective. All of these categories are not at the same level. Some of them like causality rely on a cognitive model [14], whereas others like thematic announcement are based on empirical observations of textual organization.

publié dans
Lecture Notes in Intelligence Artificial

3 Contextual Rules

Text is the input to contextual rules. In what follows we define a computational model of (written) text, under which operate our rules.

3.1 A Model of Text

The basic units of a text are its words (or lexical units) and punctuation marks occurrences. As usual, we use the term *token* for designing each of these occurrences of textual units. Tokens occur in a specific order in a given text. Taking account of these elements, a text may be directly modelled by a finite sequence over the set of all words and punctuation marks. We call this model *T0*.

T0 : $T = w_1 \ldots w_n$, where n is the *length* of text T and w_i , $1 \leq i \leq n$, are its tokens

But usually there is more information in texts, mainly structural organization (titles, paragraphs, sections, etc.) and presentation features that distinguish some portions of texts of other ones. If we want to consider this information it is necessary to extend the basic model of text.

Also, different processes may add information to text. For instance, a tagger adds a morpho-syntactic category to each word. It is desirable to be able to represent all this information in a homogeneous form.

These considerations lead us to an extension *T1* to our initial model *T0*. We consider that all structural, presentational, syntactic, semantic, etc. information that is initially present in a text *T* or that may be added to it is essentially a qualification over a subsequence of text: i.e., a contiguous segment of text. Consequently, we propose a model where the basic component is a segment of text and its label. We represent a segment of text with a pair of indices denoting its first and last positions, whereas positions denote inter-token places.

A text *T* is then modelled (under *T1*) by a finite set *G*, defined as follows:

Let T be a text with n tokens and additional qualifying elements. Each token is a word or punctuation mark from a set Σ (the lexicon). Let N be the set of all additional labels that can be used to qualify segments in T and $V = N \cup \Sigma$ the total set of labels. We assume that $N \cap \Sigma = \Phi$ and that V is a finite set.

We define:

1. A *span* in T is a pair *(i,j)* of integers, $0 \leq i \leq j \leq n$. The span (i,j) *contains* the span (k,l) if $i \leq k$ and $l \leq j$. A *partition* of size *n* of span (i,j) is a set of n distinct spans $(k_1, k_2), (k_2, k_3), \ldots, (k_n, k_{n+1})$ with $i=k_1$, $j=k_{n+1}$
2. An *extent* in T is a pair *[s, L]*, where *s* is a span in T and *L* an element of V.
3. *G* is a set of extents in T such that:
    a. for each token *w* in T in position i, $[(i-1,i), w] \in \Gamma$
    b. for each additional element with label *L* that qualifies the segment from position i to position j in T, $[(i-1, j), L] \in \Gamma$
    c. There are no more elements in Γ.

We will refer to a text by *T* or by *G*, depending on the model used. In first case we can only refer to its tokens. In both cases the length (n) of a text denotes its number of tokens.

Γ contains an extent for each token in T, and additional extents for the other pieces of information. It is a task for a linguist to correctly identify the different concepts underlying labels. Notice that different sets of extents can be obtained from the same text, depending on which is the information we choose to represent. In section 3.2.2, an example of Γ for a simple text can be found. A similar way for defining texts was proposed by Clarke et al. in their paper *An Algebra for Structured Text Search* [7].

3.2 Description of Contextual Rules

A contextual rule is an expression whose purpose is to identify and label portions of text. A portion of text gets labelled if it satisfies a condition that is a function of the same portion and, eventually, of a portion of text that precedes it (left context) and/or of a portion of text that follows it (right context). A portion of text satisfies the condition that allows the labelling if it includes certain elements (words, punctuation marks, portions of texts previously labelled) in a specific order. The rule determines what are the elements that have to be present and the order between them. These elements have not always to be contiguous. Intercalated between them there can be other portions of text for which the rule only states the maximum size and a list of elements that these portions of text must not contain.

In what follows, we define all these notions in a precise way.

3.2.1 Syntax of Contextual Rules.

Let V be a finite set of labels, a contextual rule over V is an expression of the form:

Label → *LeftContext \ Body / RightContext; Sets Specification*

where:

- Label ∈ V
- RightContext, Body and LeftContext are strings with two types of elements : labels belonging to V and Exclusion Zones. RightContext is the string that follows the '/', LeftContext the string that precedes the '\' and Body the string that is between '\' and '/'.
- RightContext and LeftContext may be empty, Body cannot be empty.
- The string *RightContext.Body.LeftContext* ('.' stands for string concatenation) will be referred as the *condition* of the rule. In this string labels must surround any exclusion zone.
- An exclusion zone is an expression with form **(ExcludedSet, Size)*, where *ExcludedSet* is the name of a set of labels and *Size* a natural number
- SetsSpecification is the definition (by enumeration) of the sets mentioned in the exclusion zones of the rule. A set in *SetsSpecification* may be empty. Nevertheless, both labels surrounding the exclusion zone are considered to belong to it. If there are no Exclusion Zones, the part *SetsSpecification* is not present.

There is one additional constraint (the reason for this constraint will be discussed in section 4.4): Label cannot belong to any exclusion zone in the rule Label. This condition is checked prior to rule execution (at compilation time).

publié dans
Lecture Notes in Intelligence Artificial

Below there is an example of a contextual rule:

CR : *relProp* → *\ relPron *(S,5) finVU *(S,10) / finVU; S={relPron,iniProp, finVU}*
where:

- V = {relProp, relPron, finVU, iniProp},
- LeftContext is empty, RightContext is the string *finVU* and Body the string *relPron*(S,5) finVU *(S,10).*
- CR has two Exclusion Zones: **(S,5)* and **(S,10)* with the same ExcludedSet (*S*). S is defined in the final part of the rule: *S={relPron,iniProp, finVU}*

The complete syntactic description of contextual rules in extended BNF notation is given in figure 1.

```
ContextualRule ::= label '→' Rhs (';' SetSpecs)?.
Rhs ::= '\' R '/' | R Oi R '/' | '\' R Od R | R Oi R Od R.
R ::= label ( ExZ label | label)*.
Oi ::= '\' (ExZ)? | (ExZ)? '\'.
Od ::= (ExZ)? '/' | '/' (ExZ)?.
SetSpecs ::= (SetSpec)+ .
SetSpec ::= identifier '=' '{' (label (',' label)*)? '}'.
ExZ ::= '*' '(' identifier ',' integer ')'.
```

Fig. 1. Syntax of contextual rules in EBNF

3.2.2 Meaning of Contextual Rules.

We give meaning to sets of contextual rules in terms of the results of its application to texts. Let *CR* be a contextual rule over set *V* of labels, *G* a text of length *n* over *V*, α and Condition strings over (V ∪ (ExclusionZones over V)) and (k,l) a span contained in (0,n) over *G*, where:
CR: *L → LeftContext \ Body / RightContext ; SetsSpec*,
Condition = LeftContext.Body.RightContext = C_1,…, C_s

We will define what does it mean that the span (k,l) satisfies the condition of the contextual rule CR and which are the elements that are added to the text. A span (k,l) may satisfy the *Condition* part of CR in more than one way, so we restrict our definition to a given partition of (k,l).

1. The span (k,l) *satisfies a for partition P* of size s of span (k,l), P = {t_1 , …, t_s}, if for all i between 1 and s the span t_i satisfies C_i, where:
   a. t_i *satisfies a label L* iff the extent [t_i , L] belongs to Γ.
   b. t_i = (k',l') *satisfies an exclusion zone *(S,n)* iff
      i. n ≥ k'-l' (the span is not greater than the size of the exclusion zone, measured in tokens)
      ii. ∀ L ∈ S, there does not exist a span (i,j) , j ≤ l', j ≥ k', such that (i,j) satisfies L
2. If span (k,l) satisfies the Condition part of contextual rule CR for partition P, with span (m,h) contained in (k,l) satisfying the first element in Body and span (r,s) contained in (k,l) satisfying its last element, a new extent e = [(m,s), L] is derived.

3. We represent by the function $f_{CR,(k,l),P}$ the application of rule CR to text Γ for span (k,l) under partition P. It is defined by:
   i. $f_{CR,(k,l),P}(\Gamma) = \Gamma \cup \{e\}$ if there exists an extent e such that it is the extent derived (2) from Γ by rule CR with span (k,l) and partition P
   ii. $f_{CR,(k,l),P}(\Gamma) = \Gamma$ otherwise

We say that *text G derives in one step the extent e = [(m,h), L] with rule CR* if there exists a span *(r,s)* contained in *(0,n)* and a partition *P* of *(r,s)* such that *e* is the extent derived according to rule 2 for span *(r,s)* and partition *P*.

In a real situation we do not have only one rule but a set of rules SCR that interact between them. We define a relation $\vdash_{SCR}$ between texts and extents. If $G \vdash_{SCR} e$, we will say that the *extent e is derivable from G by means of rules SCR*. Given the set SCR, the resulting set is not unique: it depends on the order of application of rules and the spans choice. We define the derivation relation by means of a sequence of elementary one step derivations.

Let *SCR* be the set $\{CR_1, \ldots, CR_n\}$. We say that $G \vdash_{SCR} e$ if there exist a finite sequence $R = CRp1, \ldots CRph$ of rules in SCR, a finite sequence $S = sp1, \ldots, sph$ of spans contained in (0,n) and a finite sequence of partitions $P = Pp1, \ldots Pph$, where each Pi is a partition of span si , such that: $G_{p1} = f_{CRp1,sp1,Pp1}(G)$, …, $G_{ph} = f_{CRph,sph,Pph}(G_{p(h-1)})$ and $e \in G_{ph}$

Additional restrictions on rules or on a preferred order of application have to made in order to obtain independence of one step derivations order in the derivation of *all* results from Γ and SCR. Notice that, in the derivations of all possible results, the contexts in rules do not cause ambiguity problems (as occurs in the framework of finite state calculus with the replace operator defined by Kaplan and Kay [16]). But the negation that is implicit in Exclusion Zones makes the set of all results sensible to the order of elementary derivation steps. Related to this problem, one possibility that is being studied is the definition of a property of *stratification* for sets of contextual rules, similar to the property with same name defined by Lloyd [19] for normal logic programs.

In what follows, we illustrate some of the previous concepts by means of an example. Consider the following text:

The man that I have seen recently is your father.

with morpho-syntactic information added by a tagger at a previous stage of analysis.
It is modelled by the following set of extents:

Γ = {[(0,1),The], [(1,2), man], [(2,3), that], [(3,4), I], [(4,5), have], [(5,6), seen], [(6,7), recently], [(7,8),is], [(8,9),your], [(9,10),father], [(10,11),.], [(0,1),det], [(1,2), noun], [(2,3), relPron], [(3,4), persPron], [(4,5),finAux], [(5,6), ppart], [(6,7),adverb], [(7,8),finVerb], [(8,9),poss], [(9,10),noun], [(10,11),punct], [(4,6), finVU], [(7,8), finVU] }

CR : *relProp ® \ relPron *(S,5) finVU *(S,10) / finVU; S={relPron,iniProp, finVU}*

If the contextual rule CR is applied to Γ we see that:

(i) span (2,3) satisfies relPron
(ii) span (3,4) satisfies the first exclusion zone: *(S,5)
(iii) span (4,6) satisfies finVU

(iv) span (6,7) satisfies the second exclusion zone: *(S,10)
(v) span (7,8) satisfies finVU

By consequence, the condition part of rule CR is satisfied by span (2,8). As a result of the application of rule CR to Γ we obtain Γ' = Γ ∪ {[(2,7), relProp]}. It is easy to see that there is only one span contained in (0,11) that satisfies the rule, the span (2,8). There is only one partition under which this span satisfies the rule.

Remarks

1- This example is just for explanatory purposes, and we do not claim that rule RC is an adequate description of any class of relative propositions in English. However, it is worth noticing that a similar rule has been included in a system for the segmentation of French texts in propositions (figure 2 shows the output of this system for a news text) and the results are quite satisfactory.
2- Morpho-syntactic categories names intend to be auto-explanatory.
3- Rule RC cover a variety of cases. For instance, the relative propositions embedded in the sentences:
The man that I have seen recently in the park is your father.
The man that John and Mary saw five minutes ago is your father.
would be recognized by this rule.
4- Finally, although the information used in the previous example is mainly morpho-syntactic, this is not a restriction in the system. Labels may denote semantic, pragmatic, textual organization information and rules can be stated on the basis of these categories.

Antarctique -médecin -USA WELLINGTON -
[prop p8/ Le médecin américain de la station de recherche Amundsen -Scott , au pôle sud [relProp prl3/ , qui se traite elle-même contre un cancer du sein depuis le mois de juillet , /relProp prl3] va pouvoir être évacué par un avion militaire américain [relProp prl2/ qui est parvenu à atterrir samedi sur la base [relProp prl1/ où règne une température de proche de 50 degrés celsius /relProp prl1] /relProp prl2] /prop p8] (Agence France Presse, 16/10/1999)

Fig. 2. Segmentation in propositions of a French text.

Recursive structures arise naturally in natural language analysis. In particular, recursion is essential in the segmentations of texts in propositions. In figure 2, we show a French text (news from *France-Presse* agency) with the propositions retrieved by our system. *P8*, *prl1*, *prl2* and *prl3* are the names of the rules that have been used. We present and describe briefly rules *prl1* and *prl2*:

*prl1 : relProp → \ relPron *(S,20) finVU *(S,20) / sent; S={vu, iniProp}*
*prl2 : relProp → \ relPron *(S,20) finVU *(S,20) relProp *(S,10) / sent; S={vu, iniProp}*

The two rules (*prl1* and *prl2*) label relative propositions The first one states roughly that a text segment is a relative proposition (*relProp*) if it starts with a relative pronoun (*relPron*), it contains a finite verbal unit (*finUV*), it ends at the end of the sentence (*sent*) and a condition about two zones is satisfied. These zones are the text segment between the relative pronoun and the finite verbal unit and the text segment between the verbal unit and the end of the sentence. Either of these zones is

admissible as part of the relative proposition if it does not contain a text segment labelled as a verbal unit (*vu*) or as an initial part of a proposition (*iniProp*), and its size is less or equal than 20 (this is an empirical value, adjusted by testing). The second rule is similar to the first one, with the difference that it allows a kind of embedding of relative propositions. The set of rules for propositions is going to be used on a corpus of accident reports; the identification of clauses in such texts is a mandatory stage for a system [2], which aims at displaying a sequence of images by interpreting these texts.

4 A Logic for Text Analysis with Contextual Rules

4.1 Previous Work

Grammar systems have been viewed as deductive systems. This point of view is widely used in categorial grammars [19], [21]. Under this approach, inference rules mimic grammar rules and the parsing of a string is transformed into a deduction from the lexical categories of the input string.

Rewriting context free rules can also be seen as inference rules and parsing with these rules as a deductive process. Given a grammar, $G = (V, T, S, P)$, where V is the set of variables, T the set of terminals, S the start symbol and P the set of production rules [13] there is deduction of a string $w_1...w_n$ if the relation $S \vdash w_1...w_n$ holds, where '$\vdash$' is the reflexive and transitive closure of the derivation relation '$\Rightarrow$' of production rules.

Shieber, Schabes and Pereira, in their paper *Principles and Implementation of Deductive Parsing* [25] propose a more fine-grained approach to parsing as deduction, with an explicit representation of input string positions in the atomic formulas of the logic. Following this line, we have developed a logic for parsing (i.e., analysing texts) with contextual rules.

In a similar way than in the deductive systems proposed by the mentioned authors, in our system inference rules have the form:

$$\frac{A1 \quad ... \quad An}{B} \qquad \textit{<side conditions on Ai, B>}$$

The antecedents *A1...An* and the consequent *B* of an inference rule are formula schemata; they may contain syntactic metavariables to be instantiated by terms when the rule is used. A formula *B* can be *deduced* if it is obtained as the consequent of an inference rule after a finite number of applications of inference rules to axioms or other formulas deduced from the system. Axioms and inference rules must be provided in order to define a parsing logic. Definition of a goal is more related to control than to the deductive engine. In our case, the intention is to obtain all possible items that can be soundly deduced from axioms and inference rules and we do not define goals.

This view of parsing as deduction clarify details about the parsing process and sets a framework for the discussion of correctness and completeness of parsing algorithms and its relation with search strategies in the space of possible deductions.

Second International Conference on Intelligent Text Processing and Computational Linguistics

publié dans
Lecture Notes in Intelligence Artificial

4.2 Right-corner Chart-Parsing and Contextual Rules

We present a logic for parsing with contextual rules. The parsing method is bottom-up with left to right scan of the input string and right to left scan of the right hand side of grammar rules. It is an inactive chart parser, with rules triggered by the rightmost symbol of the right hand side, i.e., the right corner. The right-corner chart parsing method for context free grammars has the following interesting property (under a breadth first search strategy): when processing a rule, a category exists if and only if it has been added to the chart.

4.3 A Logic for Right-Corner Inactive Chart Parsing for Contextual Rules

Let *G* be a text of length *n* over total set of labels *V* with token alphabet *S*, *SCR* a set of contextual rules over alphabet *V*. Letters *i, j, i', j'* denote inter-tokens position in *G*, $0 \leq i, i', j, j' \leq n$. In all formulas it holds $i \leq i'$, $j' \leq j$, additional constraints are explicitely stated. We use greek letters $\alpha, \beta, \ldots$ to denote substrings of the right hand side of a contextual rule, capital letters *A, B, …* to denote elements of V.

The Atomic Formulas

We have two kinds of propositional atomic formula schemata.

[i, A, j] - Inactive chart element of category *A* covering positions *i* to *j* of the input text. The intended meaning is that $G \vdash_{SCR} [(i,j), A]$

[i, i', A → α • β, j', j] - Active chart element. $A \rightarrow \alpha.\beta$ is a rule in *SCR*. The dot signals the scanning position in the rule; the index *i* the scanning position in the input text and the index j the initial scanning position. The intended meaning is that there exists *k*, $i \leq k \leq j$ such that span *(k, j)* satisfies the string *b'*, where *b'* is *b* without any slash ('\', '/') that it might content.

The Axioms

There is an axiom for each element (extent) present in the input text.

$[i, C, j]$, $0 \leq i < j \leq n$, $C \in V$

Axioms take the form of inactive items.

The Inference Rules

We define inference rules for triggering a contextual rule according to its rightmost symbol (predict), for processing a contextual rule (complete) and for generating new inactive items (active to inactive).

Rule P1 - Predict

$$\frac{[i, B, j] \qquad A \rightarrow \alpha\, B}{[j, j, A \rightarrow \alpha\, B \bullet, j, j]}$$

publié dans
Lecture Notes in Intelligence Artificial

Rule P2 - Predict

$$\frac{[i, B, j]}{[j, j, A \rightarrow \alpha\ B\ /\ \bullet, j, j]}\ A \rightarrow \alpha\ B\ /$$

An inactive element labelled *B* triggers rules whose rightmost symbol is *B* (P1) or that end in the sequence *B /* (empty right context). In the first case *B* belongs to the right context. The complete rule is specialised according to the symbol that is being scanned (the symbol left to the dot).

Rule C1 - Complete

$$\frac{[i, i', A \rightarrow \alpha\ /\ \bullet \beta, j', j]}{[i, i', A \rightarrow \alpha \bullet / \beta, i, j]}$$

The forward slash signals the rightmost position of the rewriting zone, i.e., the last position of an element with label *A* if it is finally deduced.

Rule C2 - Complete

$$\frac{[i, i', A \rightarrow \alpha\ \backslash \bullet \beta,\ j', j]}{[i, i, A \rightarrow \alpha\ \bullet \backslash \beta, j', j]}$$

The backward slash signals the leftmost position of the rewriting zone, i.e., the first position of an element with label *A* if it is finally deduced.

Rule C3 - Complete

$$\frac{[i, i', A \rightarrow \alpha\ B \bullet \beta,\ j', j] \quad [k, B, i]}{[k, i', A \rightarrow \alpha\ \bullet B\ \beta, j', j]}$$

Simple left completion. The scanning index *i* is modified to the start of the existing element the rule is seeking for.

Complete - C4 - Exclusion Zone

We represent exclusion zones in a compiled form, making explicit the previous label in the rule. This label acts as a cut for the exclusion zone. An exclusion zone in the rule has the form: *ex(S,C,N)*, where *S* is the name of the set of excluded labels, *C* the previous label in the rule and *N* the maximum size of the zone.
There are two rules for processing an exclusion zone.

Rule C4a - Complete

$$\frac{[i, i', A \rightarrow \alpha\ ex(S,C,N) \bullet \beta,\ j', j] \quad [k, C, i]}{[k, i', A \rightarrow \alpha\ \bullet ex(S,C,N)\ \beta, j', j]}$$

publié dans
Lecture Notes in Intelligence Artificial

Rule C4b - Complete

$$\frac{[i, i', A \rightarrow \alpha\ ex(S,C,N) \bullet \beta,\ j', j] \quad \neg[k, C, i] \quad \neg[k', C_i, i]}{[i1, i', A \rightarrow \alpha\ ex(S,C,N1) \bullet \beta, j', j]} \quad N>0,\ N1=N-1,\ i>1,\ i1=i-1,\ C_i \in S$$

In this rule, S is the set of excluded labels, *C* the cut label and *N* the maximum size of the zone. Under the absence of the cut label at current text position (index *i*), and the absence of each of the labels belonging to the excluded set, parsing proceeds with the scanning of the zone with maximum size *N1*, from string position *i1*, to the left. The side conditions ensure that we have not arrived to the maximum size of the zone (*N>0*) and that we have not arrived to the beginning of the input string (*i>1*).

Rule A1 - Active to Inactive

$$\frac{[i,\ i', A \rightarrow\ \bullet \alpha, j', j]}{[i', A, j']}$$

This rule creates inactive items from active but completed items (the dot is at the beginning of the right hand side). The new inactive item is labelled with the label *A* at the left-hand side and it spans positions *i'* to *j'* from input string.

Negative Elements
Negative information can not be deduced from axioms with inference rules having the form $A_1, ..., A_n \rightarrow B$, i.e., Horn clauses. As in the logic programming paradigm, negation should be understood as the failure in proving the corresponding positive information. That is, infer $\not\vdash A$ under the failure in proving *A*. This is a condition about deductions in the system, and it is tested in our system by the side conditions of rules. It simply amounts to testing the non existence of inactive elements having the form $[k, A, i]$, $0 \leq k < i$.

Rule N1 - negative elements

$$\frac{\neg\exists\ [i, A, j],\ 0 \leq i < j}{[i, A, j]}$$

To illustrate the behaviour of these inference rules we will present a deduction of a relative proposition with some rules and an input sentence.

R1: finVU → \ finVerb /
*R2: finVU → \ finAux *(S, 2) partVerb /; S={}*
*R3 : relProp → \ relPron *(S,5) finUV *(S,10) / finUV; S={relPron,iniProp}*

These rules are not realistic although they give some flavour of the intended purpose of our system. *finUV* stands for finite verbal unit, *finVerb* for verb in finite form, *finAux* for auxiliar in finite form, *partVerb* for past participle and *relPron* for relative pronoun. The input sentence : *The man that I have seen yesterday is your*

father. gives rise to the axioms concerning the grammatical categories of lexical units shown in figure 3. We do not represent the axioms corresponding to the input tokens.

A1	[0, det, 1]	The
A2	[1, noun, 2]	man
A3	[2, relPron, 3]	that
A4	[3, perPron, 4]	I
A5	[4, finAux, 5]	have
A6	[5, partVerb, 6]	seen
A7	[6, adverb, 7]	yesterday
A8	[7, finVerb, 8]	is
A9	[8, det, 9]	your
A10	[9, noun, 10]	father
A11	[10, punct, 11]	.

Fig. 3. Axioms for the input sentence *The man that I have seen yesterday is your father.*

In what follows, we present a step-by-step deduction of the relative proposition *that I have seen yesterday* in the example sentence.

1 [5, partVerb, 6] - Axiom A6 (*seen*)
2 [6, 6, finUV → \ auxFin *(S,2) partVerb /•, 6, 6] - Predict P2 from 1 and R2
3 [6, 6, finUV → \ auxFin *(S,2) partVerb • /, 6, 6] - Complete C1 from 2
4 [5, 6, finUV → \ auxFin *(S,2) • partVerb /, 6, 6] - Complete C3 from 3 and 1
5 [4, 6, finUV → \ • auxFin *(S,2) partVerb /, 6, 6] - Complete C4a from 4 and A5
6 [4, 4, finUV → • \auxFin *(S,2) partVerb /, 6, 6] - Complete C2 from 5
7 [4, finUV, 6] - (*have seen*) Active to inactive from 6
8 - 12 [7, finUV, 8] (*is*) from A8, Predict P2, Complete C1, C3 and C2, A-I
13 [8, 8, relProp → \ relPron *(S,5) finUV *(S,10) / finUV •, 8, 8] Predict P1 from 12
14 [7, 8, relProp → \ relPron *(S,5) finUV *(S,10) / • finUV, 8, 8] C3 from 13 and 12
15 [7, 8, relProp → \ relPron *(S,5) finUV *(S,10) • / finUV, 7, 8] C1 from 14
16 [6, 8, relProp → \ relPron *(S,5) finUV *(S,9) • / finUV, 7, 8] C4b from 15 and N1
17 [4, 8, relProp → \ relPron *(S,5) • finUV *(S,9) / finUV, 7, 8] C4a from 16 and 7
18 [3, 8, relProp → \ relPron *(S,4) • finUV *(S,9) / finUV, 7, 8] C4b from 17 and N1
19 [2, 8, relProp → \ • relPron *(S,5) finUV *(S,9) / finUV, 7, 8] C3 from 18 and A3
20 [2, 2, relProp → • \ relPron *(S,5) finUV *(S,9) / finUV, 7, 8] C2 from 19
21 [2, relProp, 7] (*that I have seen yesterday*) A1 from 20

All possible deductions from axioms with inference rules using grammar rules are computed. The results of successful applications of grammar rules are found under the form of inactive elements. Some deduction steps have been condensed (8 -12) and the use of inference rule N1 for negation is not explicitly shown.

4.4 Management of Exclusion Sets

The bottom-up right-corner chart parsing method was selected as best suited to manage possible inconsistencies that might arise from the use of negation in the exclusion zones of contextual rules.
Consider the following rule:

*A ® \ B *(S,n) E / ; S ={D}*

In this rule, the right corner symbol is *E*, so the rule is triggered under recognition of *E*. Recognition guided by the rule proceeds in a right to left way. Next category to be recognised is **(S,n)*, that is, a text span of length not greater than n (possibly empty) without any occurrence of a label *D*. If all labels that may apply to the positions in text before *E* last position have been recognised, it is possible to decide for the non-occurrence of a label *D*. There remains, however, a problem related with the co-occurrence of negation of grammar symbols and contexts in the rules. Consider the following rules:

*1. A ® B \ *(S,n) /C ; S={D}*
2. D ® B \ T / C X

from configuration ... *B T C X ... (extents [(i,j),B], [(j,k),T], [(k,l),C], [(l,m),X] belong to G)* we can deduce, when processing symbol *C* (right corner for rule 1)

rule 1 ---→ ... *B A C X* ... , under the assumption there is not an extent with label D between *B* and *C*

and from rule 2, when processing its right corner *X*

rule 2 ----→ ... *B D C X* ... and conditions for deducing *A* from rule 1 are no longer valid !

In order to avoid this kind of inconsistency we restrict negation to categories whose deduction does not include rules with no empty context (see the constraints defined in section 3.2.1). In this way, under an adequate search strategy, all the information needed to conclude the non-existence of an extent in a text span is available when needed. In fact, if the chart does not contain an element for the corresponding label, it is sure to conclude its negation.

4.5 Search Strategies, Soundness, Completeness.

The parsing logic defined in the previous section does not commit to any search strategy of deductions. Inactive items deduced by any possible search strategy are *correct*, as it holds that if there is a deduction for the inactive item [i, A, j], then

$\Gamma \Rightarrow_{SCR}$ [(i,j), A]. But not all strategies produce *consistent* results, in the sense given in previous section.. This happens because of the nature of the *negation as failure* [19] associated with the semantics of forbidden elements in exclusion zones. As labels that appear in exclusion zones cannot be deduced from rules with not empty context there is a search strategy that guarantees that negative items are present when needed. It corresponds to the following order for trying inference rules: *complete* (including *negation* when it is needed), *active to inactive* and *predict*. If there is various alternatives for applying *predict*, the inactive item with the least right index is chosen. In this way all the deductions (i.e. inactive items) necessary to correctly compute an exclusion zone are present when needed. Notice that as *complete* is the inference rule with higher priority at any stage of parsing there can be at most one application of *complete*, including *inactive to active* which is merely a form of *complete*.

As each new item that is inferred triggers the deduction of its possible consequences, the previously mentioned search strategy is also complete with respect to a relation of *consistent* derivation between texts and extents. The deduction process is not goal-directed and it naturally proceeds until no more rules can be applied. At the end of the process, all inferred elements could be found under the form of inactive items.

A parser for contextual rules has been implemented, following the inference rules and the proposed search strategy.

5 Conclusions, Future Work

We have presented a new rule-based framework for analysing natural language texts. Its main features are that they provide context conditions for rule application and that they allow underspecification of text segments by means of a restricted form of negation. A bottom-up right-corner parser for these rules has been developed, relying on a definition of parsing as deduction.

Some extensions are being presently studied. In first place, an extension of atomic categories to structured terms in order to increase the rules expressive power. On the other side, efficiency and ambiguity issues are under consideration. It would be interesting to have a way to decide between competing parses for the same text segment. If this decision can be taken locally, there will be an increase in efficiency, as fewer items would be available for subsequent rules processing.

Acknowledgments. The authors wish to acknowledge the contribution of Professor Jean-Pierre Desclés in offering valuable suggestions and stimulating discussions during the course of this work.

This work has been developed with the support of an *ECOS* grant (French-Uruguayan cooperation) and a *Csic* grant (*Universidad de la República*, Uruguay).

References

1. Adam J.M.: *Éléments de linguistique textuelle*, Mardaga. Liège, (1990)
2. Battistelli D.: Passer du texte à une séquence d'images, analyse spatio-temporelle de textes, modélisation et réalisation informatique (système SPAT). PhD, Université Paris-Sorbonne (2000)

Second International Conference on Intelligent Text Processing and Computational Linguistics

publié dans
Lecture Notes in Intelligence Artificial

3. Ben Hazez S., Minel J-L., Designing Tasks of Identification of Complex Patterns Used for Text Filtering. RIAO'2000, Paris, (2000), pp. 1558- 1567
4. Berri, J., Cartier E., Desclés J-P, Jackiewicz A., Minel J.L., 1996, Filtrage Automatique de textes. *Natural Language Processing and Industrial Applications*, pp 28-35, Moncton, N-B, Canada.
5. Cartier E.: LA DÉFINITION : ses formes d'expression, son contenu et sa valeur dans les texte. Work in progress , Université de Paris Sorbonne. Paris (1997)
6. Charolles M.:Les plans d'organisation textuelle , période, chaînes, portées et séquences. *Pratiques*, n 57, Metz (1988)
7. Clarke Ch., Cormack G. V., Burkowski F. J., An algebra for structured text search and a framework for its implementation. The Computer Journal, 38(1), pp. 43-56, 1995
8. Desclés J-P, Cartier E., Jackiewicz A. , J-L. Minel.: Textual Processing and Contextual Exploration Method. *CONTEXT 97*, Universidade Federal do Rio de Janeiro, Brésil (1997) pp. 189-197.
9. Desclés, J-P.: *Systèmes d'exploration contextuelle. Co-texte et calcul du sens.* (ed Claude Guimier). Presses Universitaires de Caen, (1997), pp. 215-232.
10. Garcia D.: Analyse automatique des textes pour l'organisation causale des actions Réalisation du système informatique COATIS. PhD. Université Paris-Sorbonne, Paris (1998)
11. Gazdar G., Mellish C.: Natural Language Processing in Prolog. Addison-Wesley (1989)
12. Hobbs J., Appelt P D.,Bear J., Israel D., Kameyama M., Sticckel M., Tyson M.: FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Texts in Finite-State Language Processing. eds. Roche E. Schabes Y. MIT Press (1997)
13. Hopcroft J.,Ullman J.: Introduction to Automata Theory. Languages and Computation. Addison-Wesley, USA (1979)
14. Jackiewicz A.: La notion de cause pour le filtrage de phrases importantes d'un texte. *Natural Language Processing and Industrial Applications*, Moncton, N-B, Canada (1996) pp. 136-141
15. Jackiewicz A.: L'expression de la causalité dans les textes. Contribution au filtrage sémantique par une méthode informatique d'exploration contextuelle. PhD, Université Paris-Sorbonne (1998)
16. Kaplan R., Kay M.: Regular Models and Phonological Rule Systems. Computational Linguistics. Vol 20, n°.3 (1994)
17. Kartunnen L.:The replace operator. in eds. Roche E. Schabes Y. MIT Press (1997)
18. Lambek J:, The Mathematics of Sentence Structure. American Mathematical Monthly (1965)
19. Lloyd J.: Foundations of Logic Programming. Springer-Verlag (1987)
20. Minel J-L., Desclés J-P., Cartier E., Crispino G., Ben Hazez S., et Jackiewicz A: Résumé automatique par filtrage sémantique d'informations dans des textes. Présentation de la plate-forme FilText. *TSI* (2000).
21. Moortgat M. Categorial Investigations: Logic and Linguistic Aspects of the Lambek Calculus. Ph.D Thesis, University of Amsterdam, Amsterdam (1988)
22. Rastier F., Cavazza M., Abeille A: *Sémantique pour l'analyse*. Masson, Paris (1994)
23. Roulet E.: *L'articulation du discours en français contemporain*. Bern, Peter Lang (1985)
24. Roulet E.: Complétude interactive et connecteurs reformulatifs. Cahiers de linguistique française, n°8, (1987)
25. Shieber S., Schabes Y., Pereira F.: Principles and Implementation of Deductive Parsing. TR-94-08, Mitsubishi Research Laboratories, Cambridge Research Center (1994)