Meaning Extraction from a Peircean Perspective

J.J. Sarbo, J.I. Farkas, F. Grootjen,
P. v. Bommel and T. v.d. Weide
University of Nijmegen, Toernooiveld 1, 6525 ED
Nijmegen, The Netherlands, e-mail: janos@cs.kun.nl

Abstract

Meaning extraction from text documents is a form of information management. The approach suggested in this paper is based on Peirce's semiotic which, by virtue of its deeper foundation, provides us with an adequate modelling of the information content of language. We exemplify the potential of the Peircean approach by extracting the meaning of a sample English text.

Keywords language analysis, C.S. Peirce, semiotic, meaning extraction.

1 Introduction

We consider the problem of determining the structure of the information content of a text document. Such a structure will be referred to as its meaning. We will present a partial solution to this problem which is based on syntactic analysis, and we will argue that the meaning of a text (in the above sense) can be derived from the syntactic structure of its constituent sentences. Preliminary research using conceptual lattices is presented in [Sarbo and Farkas, 1995], [Sarbo, 1997] and [Sarbo, 1999]. Related research on text summarisation can be found, amongst others, in [Jones, 1993].

Meaning extraction requires an adequate modelling of language. We argue that the traditional approaches are not satisfactory in this respect, as it is witnessed by the limited success of such models in natural language processing. Traditional language modelling takes as its starting point that hierarchical structure is somehow given. However, this assumption is sometimes too rigid and cannot fit the high flexibility of language use.

The approach proposed in this paper is based on Peirce's semiotic ([Peirce, 1931]) which provides us with a deeper foundation of language. In this approach, which is monostratal, hierarchical structure arises via the interaction of language symbols as a result of linguistic semiosis ([Farkas and Sarbo, 1999], [Debrock et al., 1999]).

Because interactions are events, language will be considered a set of symbol-events, a process. We apply the process view of language to syntactic (and morphologic) structure ([Aarts and Aarts, 1982]), and by using English as an example we illustrate the potential of the Peircean framework in the parsing of sentences.

Because a text consists of sentences, and sentences are symbols, the above model can also be applied to texts. It will be argued that the meaning of a text arises as a result of the interaction of sentences, and precisely by adopting the same approach and representation as in sentence parsing.

The structure of this paper is as follows. In the first part we introduce Peirce's semiotic, and outline a Peircean approach to language and its adaptation to English. In the second part, we illustrate the potential of the Peircean model by extracting the meaning of a sample text.

2 Peirce's semiotic

Peirce's semiotic is strongly related to his categories. In his doctrine of categories Peirce states that all phenomena present three aspects which, though irreducible to one another, have a different degree of dependency. The aspect of firstness is the aspect by virtue of which each phenomenon has an absolutely novel *quality*, unrelated to anything whatever. The aspect of secondness is the aspect by virtue of which each phenomenon involves an *interaction*. The aspect of thirdness is the aspect by virtue of which each phenomenon involves some *habit* (meaning). Though secondness cannot be reduced to firstness, it presupposes firstness, and, similarly, though thirdness cannot be reduced to either firstness or secondness, it presupposes both firstness (through secondness) and secondness. This dependency of the categories is formalised by the ordering firstness<secondness<thirdness, where "<" is a total order on categories.

Peirce's early papers suggested a convergence of his theory of the three categories and his presentation of the various semiotic triads. The most important of these triads is the triad of sign, object, and interpretant, which is a kind of *ontological* triad telling us what there is in the world. Based on the ontological triad, Peirce defined three triads of sign: the triad of icon, index, and symbol, dealing with how signs refer to their objects; the qualisign, sinsign and legisign triad, referring to the sign itself, prior to any relational possibilities and actualities; and the triad of rheme, dicent sign and argument, characterising the formal rules that associate signs and objects (cf. fig. 1). The above order relation on categories can be applied to Peirce's signs, according to their category exhibited. For example, icon<index<symbol is an expression of degeneracy with respect to the realisation of the sign's object and interpretant; qualisign<icon<rheme is an expression of degeneracy with respect to the sign's ontological type.

		Ontological type		
		1 Material	2 Relational	3 Formal
Phenome-	1 Quality	Qualisign	Icon	Rheme
nological	2 Indexicality	Sinsign	Index	Dicent Sign
type	3 Mediation	Legisign	Symbol	Argument

Fig. 1: *Peirce's classification of signs*

3 Language and ontological perspective

Language appears only as a form of interaction, whether we speak it or write it. In terms of Peirce's categories, interactions, or events, represent the category of secondness. An event involves the fact *that* something happens, but says nothing whatever about *what* happens. The latter aspect is the aspect of thirdness. *What* happens in an event requires that the event be embedded in a context of events which are related to each other. Such web of related events is what is called a *process*.

Language consists of symbols. Because signs are generated from signs, and in turn generate other signs, every sign must be related to an event. From this it follows that language symbols are sign-events which, by virtue of their intepretants, are embedded within a process. Language is a process involving symbol-events which are themselves generated according to rules which, in Peircean terms, are habits evolving from interaction with other symbol-events. Language processes involve both syntactic and semantic rules or habit.

3.1 Syntactic signs

From a syntactic point of view, symbol-events have a specific function, regardless of their semantic function. The syntactic value of the language symbols making up the unit of meaning may be seen in the function of the value which they have in forming such a unit.

By virtue of their secondness, events are marked by a binary relation. Therefore, linguistic symbol-events must be also binary. This is why, strictly speaking, one lexical item by itself has no meaning. The syntactic value of the symbol-events will therefore depend upon the *sort* of relation that obtains between two language symbols.

If one of the symbols has by itself no information content and therefore is a mere quality (a phoneme or a visible character), it will need another symbol to actualise its 'potential' content. Such nexus of two symbols, one of which is self-sufficient, but the other has mere potential content, may be called a *proto-symbol* (P) which corresponds to the category of firstness. An example of this is the symbol-nexus of free morpheme and affix.

Similarly, when the nexus is constituted by an asymmetrical relation between one language symbol which derives its full content from its association with another language symbol which is in principle self-sufficient, it may be called a *deutero-symbol* (D) which corresponds to the category of secondness. An example of this is the symbol-nexus of adjective and noun, or the one of determiner and noun.

Finally, when the nexus consists of two language symbols which are self-sufficient but together generate the interpretant of the unit formed by the string, e.g. a sentence, it will be called a *trito-symbol* (T) which, by its aspect of thirdness, mediates between the language symbols constituting a unit of meaning, or a thought, in the Fregean tradition. An example of this is the symbol-nexus between verb and subject.

To complete the picture, it is necessary to say a word about the *triadic relation* characterising each of these signs, because without such relation, they would not be signs, let alone syntactic signs. But precisely what makes them *syntactic* signs is the very fact that they stand for specific *rules* or habits. Thus, the object of syntactic signs is the rule for which they stand. Their interpretant on the other hand is the generation of the selection of the next symbol-event. The interpretant of the entire string of language symbols is, from a syntactic point of view, the establishment of the correctness of the string, regardless of its semantic content.

3.2 Levels and classes of syntactic signs

In as much as linguistic symbols are also syntactic symbols, proto-, deutero-, and trito-symbols constitute a Peircean triad of linguistic symbols. By virtue of their category exhibited, these signs define the ordering $P<D<T$ which, in turn, defines the *levels* of syntactic signs. In the remaining we will denote by X a level of syntactic signs, and by X' the level subsequent to X. A sign (or symbol-event) of some level $X \in \{P,D,T\}$ will be called an X-level sign (or symbol-event).

Language implements syntactic signs basically by lexical items and their relations. These are called *syntactic structures* or, equivalently, *language units*, depending on whether we want to emphasise their structural or linguistic properties. In the mapping of syntactic signs to syntactic structures (*syntactic mapping*), the notion of argument and functor, an abstraction from the combinatorial properties of lexical items, plays a crucial role. This combinatorial property can be characterised as relational or argumental need. A lexical item has *relational need* if it can be a functor, and *argumental need* if it can be an argument in some relation.

By analysing the structure of the three types of syntactic sign, we can recognise an argument and a functor symbol in each of them. In the case of trito symbols, the functor is that symbol which has the most relational need in the determination of the interpretant. We tacitly assume that such a distinction can always be made.

We denote the constituents of an X-level symbol-event, the argument and the functor symbol, and the syntactic symbol itself as X_1, X_2 and X_3. By virtue of the category and dependency which different signs respectively exhibit, syntactic signs may be said to define the ordering $X_1<X_2<X_3$ which, in turn, defines the *classes* of

level X. The total order on levels and classes can be extended, by flattening, to a total order on syntactic signs.

The syntactic sign emerging from a symbol interaction is called its *descendant*. A syntactic sign that has no combinatorial need is a *completed* or well-formed sign. Two symbols are said *incompatible* if they cannot establish a relation syntactically, and *compatible*, otherwise. A completed sign is incompatible with any symbol. A sign of class X_i (i=1,2,3) of some level X is denoted an X_i sign.

4 The emerging syntactic sign

4.1 Symbol interactions

From a receiver's point of view, input symbols have merely potential content according to the receiver's (parser's) hypothesis. The set of such hypotheses is called the parser's dictionary. Input symbols appear one after the other, interact, and syntactic signs emerge by *symbol relation*. This might be called the 'automatic' type of sign generation. Because the descendant sign contains, besides the meaning of its constituents, the additional meaning of the relation itself, the signs generated by symbol relation are monotonously increasing. From the monotonicity of the ordering of syntactic signs it follows that a lower level combinatorial need must have priority over a higher level one.

But there are also cases of a degenerate symbol interaction. One of them is the interaction between incompatible symbols. In such a case, one of the interacting symbols, which is a sign generated in one symbol-event, is coerced to an argument or a functor, but *not* both, in another symbol-event. This type of sign generation is called *symbol coercion*.

Syntactic signs are composite signs which meet certain criteria. Thus, if a syntactic sign consists of related signs, the signs involved must in principle be *contiguous* to one another. This requirement is based upon the triadic structure of a syntactic sign the object of which is always a rule expressive of the expectation that a certain type of language unit must be followed by another type of language unit. The contiguity property can be defined as a covering relation on the ordering of syntactic signs ([Davey and Priestley, 1990]).

The contiguity property is the driving force behind symbol coercion. Briefly, two symbols which are contiguous, but cannot establish a relation on some level X, must relate with each other on some level, higher than X. For this reason, one of the interacting symbols (the one appearing first) will be forced to enter a higher class of syntactic signs without symbol relation which is the essence of symbol coercion.

4.2 Towards an algorithm for syntactic signs

The properties of *symbol coercion* are formalised as follows ($X_i \rightarrow Y_j$ denotes that a sign of class i of level X may enter class j of level Y, and $X_i \rightarrow Y_j \vee Y_k$ is a shorthand for $X_i \rightarrow Y_j \vee X_i \rightarrow Y_k$):

(α_1) $X_1 \rightarrow X_3$; (α_2) $X_1 \rightarrow X_1^{'} \vee X_2^{'}$; ($\alpha_3$) $X_3 \rightarrow X_1^{'}$.

In sum, X_1 and X_3 signs can increase their meaning without a symbol relation, but an X_2 sign presupposes an X_1 sign and must relate with it. This meets our expectation that a relational need must be fulfilled always, though an argumental need can be optional.

Because language possesses a finite number of lexical items only, some syntactic signs must be generated *incrementally* via degenerate symbol relations. In such a relation the mediation aspect is incomplete and the descendant of the X-level symbol interaction may become an X_1 or X_2 sign on the same level. Accordingly, the rules of *symbol relation* are formalised as follows (the symbol relation of X_1 and X_2 is denoted as X_1-X_2):

(β_1) $X_1\text{-}X_2 \rightarrow X_1 \vee X_2$; ($\beta_2$) $X_1\text{-}X_2 \rightarrow X_3$; ($\beta_3$) $X_1\text{-}X_2 \rightarrow X_1^{'} \vee X_2^{'}$.

4.3 Cumulative signs

Because of the incremental nature of syntactic signs, there may be encountered simultaneously more than one sign of the same class. By virtue of its aspect of firstness, an X_1 class may contain a number of signs which are *unrelated*, but the collection of which is a sign. By virtue of its aspect of secondness, an X_2 class may contain a number of signs which are *unrelated*, but which share a *common referent*. By virtue of its aspect of thirdness, an X_3 class may contain a completed sign which must be a single sign.

The simultaneous occurrence of signs of a class corresponds with another case of a degenerate symbol interaction, called *symbol stacking*. In such a case, the symbols involved are accumulated on a stack. The need for a stack is a consequence of the contiguity property, by virtue of which, symbols which are not contiguous may not interact. Stacking is a hypothesis which, due to a next symbol interaction can be further developed. In such a case, the stack has to be split and a segment of it, which must be a tail segment by the contiguity property, is removed from the stack as a single sign, as part of a symbol relation or coercion operation.

4.4 Primary signs

We assume that the input symbols enter a lowest class (prm) as a sequence of primary signs, e.g. phonemes or characters. Prm, which has the aspect of firstness, is by definition a class of syntactic signs. The input primary signs, which have no combinatorial need, are collected in prm, as long as their sequence forms a morphological symbol which is a dictionary entry. When this happens, the symbol receives its combinatorial need from the dictionary, and enters the lowest level, in particular, P_1 if it has no P-level relational need; and P_2, otherwise:

(γ) prm$\rightarrow P_1 \vee P_2$.

4.5 Mediating evaluation

Syntactic signs are yielded by symbol interaction. But the decision as to *when* the mediation takes place depends upon the type of evaluation, which can be lazy or greedy. In general, we will assume lazy evaluation of relations, because it can be more economic in some cases. The lazy evaluation of syntactic sign-events affects the modelling of the terminator symbol (e.g. the point symbol) which, therefore, will be treated as an incompatible argument and a nullary functor on each level, thereby forcing the realisation of pending relational needs.

In sum, syntactic signs arise in language due to (1) the quality of contiguity, (2) symbol interaction, and (3) mediating evaluation which, respectively, have the aspect of the categories, firstness, secondness and thirdness. The emerging syntactic sign may become part of a cumulative sign, or change its aspect of correspondence with its object, or establish a relation with another sign. In sum, symbol interactions do emerge by (1) symbol stacking, (2) symbol coercion, and (3) symbol relation, which, as above, exhibit the aspects of Peirce's categories.

5 English syntactic signs

The Peircean model we developed so far applies to language in general. The subject of this section is its adaptation to a particular language, English. After introducing an important transformation we will exemplify our model in section 5.6.

5.1 Syntactic mapping

We illustrate the syntactic mapping of language by using English as an example. In this mapping we capitalise on the semiotic properties of syntactic signs and the syntactic and conceptual distinctions that may be expressed in English ([Farkas et al., 1997]).

Trito-symbols correspond to the symmetric relation between two constituents which are both self-sufficient and require the presence of the other, e.g. the relation between noun and verb. Such a relation is called *predication*(p).

Deutero-symbols correspond to the asymmetric relation between an action/state or participant, and its properties: both are self-sufficient, and the latter requires the presence of the former, but the reverse does not hold. In English, two instantiations of this type of relation can be identified: *modification*(m), e.g. the relation between adjective and noun; and *qualification*(q), e.g. the relation between determiner and noun.

The third type of symbols, proto-symbols, correspond to the morphological relation of *affixation*(a). An affixation relation distinguishes between a root (or base), e.g. a free morpheme, and an affix: the root is self-sufficient; the affix has only potential meaning actualised by the root.

From the semiotic point of view, q- and m-signs form subsets of deutero-symbols. Using the analogy of the relational triad, q- and m-signs represent iconic and indexical meaning, respectively, and, therefore, these signs may be said to define the ordering $q<m$. Eventually, this yields the ordering $a<q<m<p$ which, in turn, defines the *levels* of the English syntactic signs.

5.2 Syntactic relations

Syntactic relations emerge due to the combinatorial need of syntactic symbols. In general, a syntactic symbol can have argumental need, optionally, but its relational need is a function of that of its constituents, or, in the case of a lexical item, it is some constant value. Lexical items can contribute to the relational need of syntactic signs, on each level.

A lexical item has a potential combinatorial need, which is a finite set. The combinatorial need of a syntactic symbol generated by a symbol relation is the disjoint *union* of the combinatorial need of its constituents, possibly modified (i.e. restricted) by the interaction itself. The potential relational need of the types of lexical items is exemplified in fig. 2 (respectively, a '+' or '-' represents the presence or absence of a relational need on the level indicated by the column). The relational need of a particular lexical item is the subset of that of its type.

For example, the q-level relational need of adjectives and adverbs allows symbol-events like *keep awake*, or *walk by*; and their m-level relational need the modification relation like *happy girl*, or *walk quickly*. In the case of a preposition, the q-level relational need contributes to the relation with the obligatory argument (qualification), and the m-level one to the modification of the optional argument by the qualification yielded.

Verb–complement relation is classified as modification. Such a symbol-nexus fits the definition of a deutero sign: verb and complement are both self-sufficient, but the verb derives its full content from the complement. Because the descendant symbol of a verb–complement relation has an indexical character (the verb points in the direction of its complement), this type of relation must be identical with modification (we admit that the terminology might be confusing for the linguist).

In sum, a verb relates with its complement(s) due to its m-level relational need (which is fulfilled when all necessary complements are found), and with the subject, due to its p-level one. A copula or an auxiliary relates with its complement due to its q-level relational need, but the copula relates with the subject due to its p-level

	a	q	m	p
primary	-	-	-	-
affix	+	-	-	-
noun	-	-	-	-
determiner	-	+	-	-

	a	q	m	p
preposition	-	+	+	-
adjective	-	+	+	-
adverb	-	+	+	-
verb	-	+	+	+

Fig. 2: *Potential relational need*

one. The SV(O) rule of English is modelled by demanding that, a sign having p-level relational need entering some level X, is incompatible with any X_1 sign except for a p_1 one, potentially.

The development of the relational need of syntactic signs can be illustrated as follows. The potential m-level relational need of a preposition will be actual if the q-level relation it is involved in does not disallow that. For example, there will be such need in the case of *in London*, and there won't be, in the case of *drive in.*

5.3 Parsing English syntactic signs

The ordering of the classes of a level is depicted in fig. 3a (edges represent the "<" relation). In the case that degenerate signs are allowed, this graph can be paraphrased as a two-level scheme consisting of a finite automaton (FA) and a number of stacks. A state of the FA corresponds to a sign class, and a transition to an application of an α or β rule, represented by solid and dashed lines, respectively (but in later graphs we will use solid lines for both types of transition). The resulting graph is depicted in fig. 3b (an edge which is a cycle is omitted).

In virtue of the syntactic mapping introduced in section 5.1, the classes of English syntactic signs define a total ordering as shown in fig. 3c. By using the interpretation of fig. 3b to fig. 3c, we get a two-level system (this is not illustrated). We map the X_3 and X_1' classes, e.g. a_3 and q_1, to same states (same signs, different interpretants). The initial state is prm, all others are final states; each state has a stack. The output language of the system is the set of signs in the different stacks, upon termination.

In [Farkas and Sarbo, 1999] we describe an equivalency transformation of the two-level system depicted in fig. 3c with respect to its input and output languages. In sum, this transformation makes use of the properties of symbol coercion allowing immediate transitions like $m_1 \rightarrow p_1$ (cf. fig. 4a) and $q_1 \rightarrow m_1 \rightarrow p_1$ (cf. fig. 4b), the orthogonality of q_2 and m_1 signs allowing these classes to be merged (q_2m_1), and the fact that the a-level morphological signs may directly enter the states of the q- and m-levels (cf. fig. 5). Notice that the state q_2m_1 exhibits the properties of both q_2 and m_1.

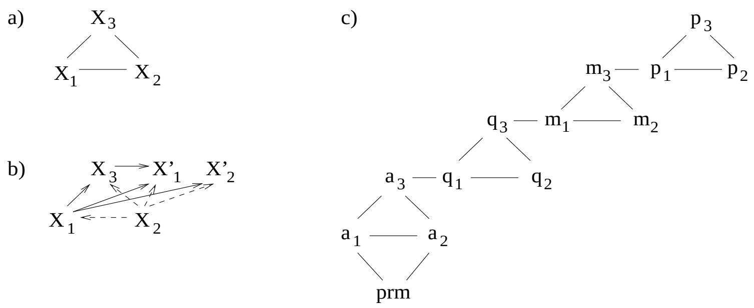

Fig. 3: *English syntactic signs*

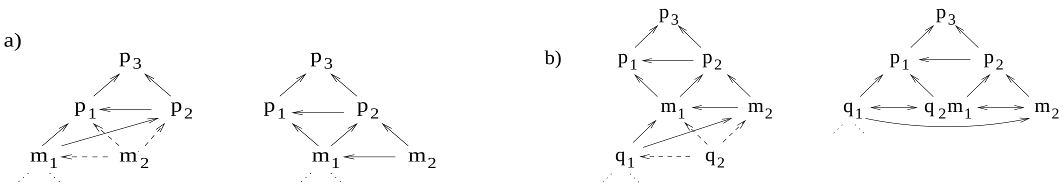

Fig. 4: *Transformation*

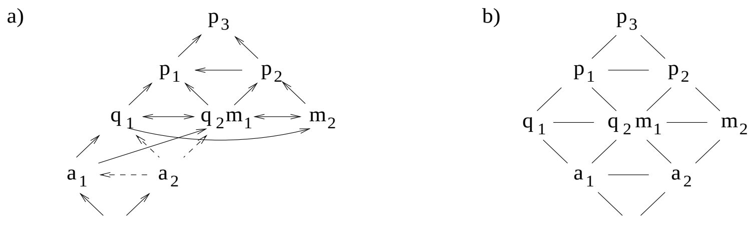

Fig. 5: *Transformation (cont.)*

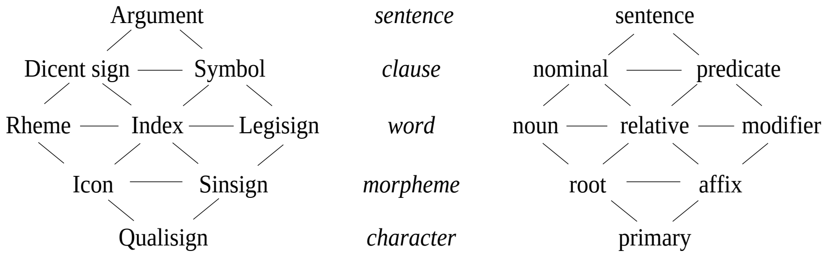

Fig. 6: *Peirce's signs and English syntactic signs*

5.4 Peirce's signs and English syntactic signs

The resulting system of fig. 5b can be interpreted, the other way round, as a classification of syntactic signs: states can be mapped to sign classes, the ordering of which is derived from the transitions (cf. fig. 3a). A comparison of this classification with Peirce's triads shows their isomorphism. The analogy between the corresponding signs is justified as follows:

prm a pure quality, unrelated to anything else; a *primary.*
$\mathbf{a}_2$ a particular quality, referring to an actually existing argument; an *affix.*
$\mathbf{m}_2$ a sign involving the convention that arguments have certain properties which are general types; a *modifier.*
$\mathbf{a}_1$ an image, a name of some 'thing', e.g. a free morpheme; a *root.*
$\mathbf{q}_2\mathbf{m}_1$ a morphological sign (a_3), or a qualifier (q_2), or an m-level sign, each involving a reference to the argument; we call them collectively, a *relative.*
$\mathbf{p}_2$ a sign involving the convention that arguments have some more basic properties; a *predicate.*
$\mathbf{q}_1$ a sign representing the possible existence of some 'thing'; a *noun.*
$\mathbf{p}_1$ a sign used to assert the actual existence of something, e.g. a clause; a *nominal.*
$\mathbf{p}_3$ a sign expressing a lawlike relation between subject (p_1) and predicate (p_2), a 'thought'; a *sentence.*

This completes our English syntactic mapping. In fig. 6, it is illustrated how English implements the signs of the 'real' world, syntactically. The upward-right diagonals represent the material, relational and formal ontological types; and modifier-, predicate- and nominal-formation, on the left- and righthand side of fig. 6, respectively; and, similarly, the upward-left diagonals represent the quality, indexicality and mediation phenomenological types; and, respectively, word-, expression- and sentence-formation.

5.5 Parsing algorithm revisited

The above classification of syntactic signs can also be seen as a specification of the formal rules of a parsing algorithm. Indeed, fig. 5(b) can be considered a 'grid' having sign classes as nodes and transitions as edges along which signs are 'moved' from one class to another as a result of symbol interactions. A descendant sign is moved upward left if it has an argumental need, and upward right if it has a relational one.

5.6 Example

In this section we show the analysis of a sentence (cf. fig. 7) which is taken from our later example illustrating meaning extraction in section 6. We omit the signs of the a-level morphological analysis (also in other examples), and assume that the input signs leave the a-level and enter q_1 or q_2m_1 conform to their combinatorial need. The potential relational need of the lexical items is as follows: *there*=$\{\}$, *are*=$\{q,p\}$, *several*=$\{q\}$, *document*=$\{\}$, *base*=$\{\}$.

In the table below, an item represents the content of the storage of a class (column) prior to the evaluation of an input symbol (row). We denote by '/' a left-associative stack constructor, and by $[\_]$ the operator that converts a stack of signs to a single sign. Symbols having a non-empty relational need are written in capitals.

step	next input	q_1	q_2m_1	m_2	p_1	p_2	p_3
0	there(t)						
1	are(A)	t					
2	several(S)	t	A				
3	document(d)		S			t-A	
4	bases(b)	d	S			t-A	
5	.	d/b	S			t-A	
6	.	.	.	.	s-[d/b]	t-A	
7	.	.	.	.	.		t-a-s-[d/b]

Fig. 7: "*There are several document-bases*"

In $step_0$, the symbol *there* enters q_1. The symbol *are*, which is compatible with *there* (a notional subject) on the q-level, enters q_2m_1 in $step_1$. The next symbol, *several*, enters q_2m_1 in $step_2$. This symbol is not compatible with *are*, and this triggers the relation between q_1 and q_2m_1. Their descendant symbol has p-level relational need, and enters p_2, accordingly. In $step_3$ and $step_4$ the parts of the compound noun are accumulated in q_1. Triggered by the terminator symbol, this q_1 relates with q_2m_1 in $step_5$, and their descendant with the predicate (p_2) in $step_6$.

5.7 Coordinate structures

Because of its importance in regard to our example of meaning extraction, we must briefly discuss the parsing of coordinate structures. Coordination is a complex phenomenon which is considered too sophisticated to be described adequately in traditional modelling. It turns out however that in the Peircean approach the analysis of such structures is most simple (because of space, in this paper we will only concentrate on the technical aspects of parsing).

The treatment of coordinate structures is as follows. First, the signs preceding the coordinator are analysed (non-deterministically) and saved temporarily. Second, the input following the coordinator is analysed stepwise. Whenever a sign of some class is found, such that, there is a sign among the saved ones, of the same class and compatible with the sign found, then the two signs are coordinated. This involves the inheritance of relations between the saved sign and the coordinated one, in agreement with their combinatorial properties.

Third, upon a successful coordination, the analysis of signs preceding the coordinator is resumed starting from the last sign coordinated. If, eventually, all signs preceding the coordinator are known, the analysis proceeds with parsing the remaining input. Technically, a coordinated sign is treated as a single sign, the future relations of which must be checked for both signs involved, separately. Information for keeping track of corresponding signs of a coordinate structure is maintained (but omitted in the examples).

6 Meaning extraction

The goal of this section is an attempt to illustrate the potential of the Peircean approach in meaning extraction from text documents. We will consider a sample text taken from [Huibers, 1996] which specifies the notion of information retrieval (IR) as follows:

(1) There are several document-bases.
(2) Each document-base contains different types of information.
(3.1) There are various types of users and
(3.2) there are vast differences between their information needs.
(4.1) There are various kind of search-tasks,
(4.2) or stated differently,
(4.3) there are several ways in which
(4.4) a user can be satisfied with
(4.5) the returned information.

We first analyse the above sentences and determine their syntactic structure, which, subsequently, will form the basis of the classification of these sentences as syntactic signs. It will be argued that the resulting classification provides us with a representation of the meaning of the given text.

6.1 Sentence level analysis

We will assume that the preposition '*of*' establishes, respectively, a q- and an m-level relation with its optional argument and its complement, whereas the prepositions ‘*between*’ and ‘*in*’ do relate the other way round. The analysis of (2) is depicted in fig. 8, the one of (1) has been shown in fig. 7.

step	next input	q_1	q_2m_1	m_2	p_1	p_2	p_3
0	each(E)						
1	document(do)		E				
2	base(b)	do	E				
3	contains(C)	do/b	E				
4	different(Di)			C	e-[do/b]		
5	types(t)			C/Di	e-[do/b]		
6	of(O)	t		C/Di	e-[do/b]		
7	information(i)	t	O	C/Di	e-[do/b]		
8	.	i		C/Di/t-O	e-[do/b]		
9	.	.	i	C/Di/t-O	e-[do/b]		
10	.		di-t-o-i	C	e-[do/b]		
11	.	.	.	.	e-[do/b]	C-di-t-o-i	
12	.	.	.	.	.	.	e-[do/b]-c-di-t-o-i

Fig. 8: “*Each document-base contains different types of information*”

The third sentence consists of two clauses, (3.1) and (3.2), the analysis of which must be clear by now (cf. fig.9). The signs available in step_6 are sufficiently analysed for the coordination which takes place in step_{15} yielding $m_2{=}\&^{\text{various/types-of}}_{\text{vast/between-their-info.needs}}$ and $q_2m_1{=}\&^{\text{users}}_{\text{differences}}$ where $\&^{a}_{b}$ is a shorthand for ‘a and b’.

The analysis of the last sentence reveals the presence of an or-coordination in which the coordinator itself is modified. The parse of (4.1) yields: $q_2m_1{=}$ *search/tasks*, $m_2{=}$ *Various/kinds-Of*, and $p_2{=}$*there-Are*. The other conjunct contains a subordinate clause (4.4–5) which is analysed recursively (cf. fig. 10). The resulting sign, that we denote by ‘σ’, arises in step_{11}. The partial analysis of (4.3–5) is displayed in fig. 11. Coordination takes place between the signs of (4.1) above, and those available in step_8. The completion of the example is simple and left to the reader.

6.2 Text level analysis

We argue that sentence and text level analysis are basically the same except that a text consists of signs which, considered individually, are *completed* symbols. In the remaining we will refer to such symbols simply as completed signs. Because of their completedness, in their classification we will capitalise on their analogy with the ‘real’ world signs, and refer to the classes of fig. 1, accordingly.

The class of a completed sign will be determined on the basis of the relations involved in it. If such a sign is one containing a p-level relation, then, in the case of English, its class is determined by the type of the verb participating in the relation. Functionally, verbs can be expressive of the types *existence*, *state* or *event*, which, respectively, correspond with quality, indexicality and mediation. A completed sign

step	next input	q_1	$q_2 m_1$	m_2	p_1	p_2	p_3
0	there(t_1)						
1	are(A_1)	t_1					
2	various(V_1)	t_1	A_1				
3	types(t_2)			V_1		t_1-A_1	
4	of(O)	t_2		V_1		t_1-A_1	
5	users(u)	t_2	O	V_1		t_1-A_1	
6	and(&)		u	V_1/t_2-O		t_1-A_1	
7	there(t_3)						
8	are(A_2)	t_3					
9	vast(V_2)	t_3	A_2				
10	diff.s(d)			V_2		t_3-A_2	
11	between(B)		d	V_2		t_3-A_2	
12	their(T_4)		d/B	V_2		t_3-A_2	
13	info(i)		d/B	V_2/T_4		t_3-A_2	
14	needs(n)	i	d/B	V_2/T_4		t_3-A_2	
15	.	i/n	d/B	V_2/T_4		t_3-A_2	
16	.	.	d	V_2/B-t_4-[i/n]		t_3-A_2	
17	.	.	.	.	$\&^{[v_1/t_2\text{-}o]\text{-}u}_{[v_2/b\text{-}t_4\text{-}[i/n]]\text{-}d}$	t_3-A_2	
18	.	.	.	.			t_3-a_2-$\&^{[v_1/t_2\text{-}o]\text{-}u}_{[v_2/b\text{-}t_4\text{-}[i/n]]\text{-}d}$

Fig. 9: "*There are various types of users and there are vast differences between their information needs.*"

step	next input	q_1	$q_2 m_1$	m_2	p_1	p_2	p_3
0	a(A)						
1	user(u)		A				
2	can(C)	u	A				
3	be(B)		C		a-u		
4	satf.(S)	B	C		a-u		
5	with(W)	S	c-B		a-u		
6	the(T)		c-b-S/W		a-u		
7	ret.(R)		c-b-S/W/T		a-u		
8	inf.(i)		c-b-S/W/T	R	a-u		
9	.	i	c-b-S/W/T	R	a-u		
10	.	.	c-b-S	W-t-r-i	a-u		
11	.	.			a-u	c-b-S-w-t-r-i	
12	.	.					a-u-c-b-s-w-t-r-i

Fig. 10: "*a user can be satisfied with the returned information*"

step	next input	q_1	$q_2 m_1$	m_2	p_1	p_2	p_3
0	there(t)						
1	are(A)	t					
2	several(S)	t	A				
3	ways(wa)		S			t-A	
4	in(I)	wa	S			t-A	
5	which(Wh)		s-wa/I			t-A	
6	σ		s-wa/I	Wh		t-A	
7	.	σ	s-wa/I	Wh		t-A	
8	.		s-wa	I-wh-σ		t-A	

Fig. 11: "*there are several ways in which* σ"

not containing a p-level relation is expressive of the existence of some 'thing'. The structural types of completed signs are the material, relational and formal types, as usual.

A verb always refers to some existing quality which has the aspect of firstness. For example, the completed sign '*the clock strikes*' involves a reference via the verb to some 'clock-striking-quality'. A verb expressive of a state points in the direction of its object, and has the aspect of secondness. Such verbs are, for example, *have*, *contain*, *is* complemented by a preposition, an adjective, or an adverb, and most intransitive verbs. Verbs expressive of an event (e.g. most transitive verbs) refer to some general or lawlike property, and have the aspect of thirdness. The above interpretation of verbs complies with the epistemological view of predication, according to which, the subject is understood as an instance of the 'concept' described by the predicate.

Meaning extraction is initiated when all completed signs of the text are input. Each classification of signs developed during the analysis will correspond with one outcome of meaning extraction. From the monotonicity property of symbol interactions it follows that we will find all such classifications, eventually.

The combinatorial properties of the text level symbols is basically due to the anaphoric relations existing between them. Technically, text and sentence level analysis differ only in one aspect. Because a text consists of completed signs, these signs can enter their class directly (we can model this aspect by defining each state, the one corresponding to a sign class, as an initial state). From this it follows that a descendant may preceed its constituents in the analysis (which implies that the rules of symbol interactions must be adjusted, accordingly). But even in such a case, we demand that the descendant derives from its 'constituents' by symbol interaction, the verification of which, however, might be beyond our scope.

6.3 Example

Sentence (1) involves a reference to an existent quality recognised as an icon (docb). (2) is a reference to an actually existing thing via a quality, a sinsign (cont.inf), which

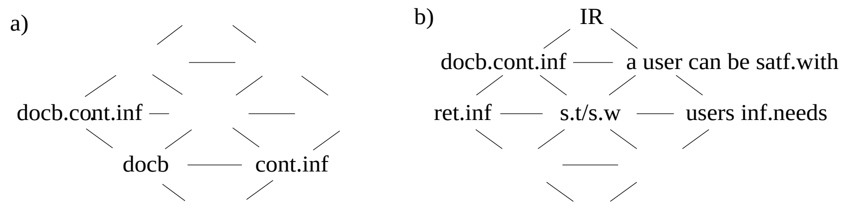

Fig. 12: *Signs yielded from the text level analysis*

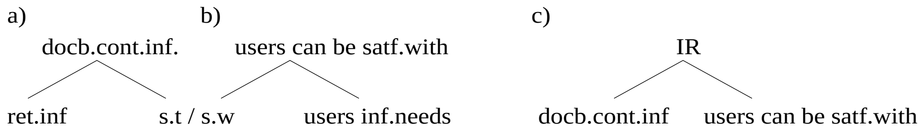

Fig. 13: *Sign triads*

classification is justified by the anaphoric relation of its subject. The symbol relation of (1) and (2) yields a rheme (docb.cont.inf), a 'thing' identified via its qualities, a sign of essence. The result of the analysis so far is displayed in fig. 12a.

Similarly, (3.1) is recognised as an icon, and (3.2) as a sinsign. Their interaction yields a sign of a general quality referring to some 'thing', a legisign (users inf.needs), which is justified by the anaphoric relation of its predicate. In the last sentence we find two icons derived from (4.1) and (4.3), respectively, (s.t) and (s.w). In as much as (4.3) is also related to (4.4–5) via the modification of '*in*', it points to that sign and, by virtue of the coordination, (4.1) must do the same. Therefore the coordination of these icons must be an index sign (s.t/s.w). Notice that this index sign and the current rheme sign are not compatible. Because (4.4) is expressive of an event referring to an iconic subject (*user*), it must be a symbol sign (a user can be satf.with). Notice that the event aspect of the verb is reinforced by its qualification.

The binding of the preposition (*with*) to the verb is explained as follows. A verb cannot be deprived of the subject and/or the complement(s) without changing its meaning, but a modification due to a qualification (known as a PP) can be removed from it (and considered as a feature, e.g. location, or condition). In as much as prepositions can be regarded as complex predicates ([Jolly, 1993]), we will consider them, semiotically, as part of the verb (we demand, however, that the argument of such a preposition is a completed sign).

Finally, (4.5) is classified as a rheme (ret.inf). The appearance of this rheme sign triggers the symbol coercion of (1)–(2), which, thereby, becomes a dicent sign. The differences in meaning between (4.5) and (1)-(2) do not allow their interaction to be implemented by symbol stacking. The resulting classification of symbols is depicted in fig. 12b.

6.4 Extracted meaning analysed

We consider the signs of fig. 12b as a representation of the meaning of our text. According to this, the meaning of Information Retrieval is that, *users* with *information needs can be satisfied* with the *returned information*, which are *document-bases containing information* yielded by the *search tasks*. We argue that this paraphrasis can be derived from the symbol relations indicated in fig. 12b.

According to the triad of rheme, index, and dicent sign (cf. fig. 13a), when the search-tasks are brought into relation with the returned information, we get document-bases containing information. The descendant of this symbol relation is a completed sign which is part of the notion of IR. Functionally, the rheme (ret.inf) and the index (search-tasks/several ways) together generate the dicent sign (docb.cont.inf) which corresponds with reasoning by deduction, in as much as each document returned must contain some information searched.

The second triad (cf. fig. 13b) tells us that, if the users' information needs are combined with the search tasks, then their relation will result in users which are satisfied, potentially. Again, the descendant sign is part of the notion of IR. This triad corresponds with reasoning by induction, in as much as it postulates that all information needs of the users can be satisfied by the search-tasks/(in)several ways. We can observe that the IR paradigm does not state that the users will be satisfied by the retrieval, but it only states it as a possible.

Finally, from the triad of dicent sign, symbol and argument it follows that, when the documents containing information are brought into relation with the users (which can be satisfied with the returned information), then we get the meaning of Information Retrieval. From the logical reasoning point of view, this triad corresponds with reasoning by abduction, in as much as it postulates the hypothesis that the users gleaning from the returned information and making yes/no judgements (whether the information was, or was not adequate) satisfy their information needs, potentially (or otherwise, they adapt their search tasks), which is precisely what the paradigm of IR says.

Conclusions

The goal of this paper is an attempt to offer a Peircean explanation of meaning in language. First, from properties of signs we derive a parsing algorithm for syntactic signs. We apply this algorithm to English and show that, by its syntactic structures, the English language implements signs, analogous to those introduced by Peirce. Second, we argue that the syntactic analysis of sentences can also be applied to the analysis of texts which, thereby, provides us with a representation of their meaning. We illustrate the proposed approach by a non-trivial example.

References

[Aarts and Aarts, 1982] Aarts, F. and Aarts, J. (1982). *English syntatic structures.* Pergamon Press, Oxford.

[Davey and Priestley, 1990] Davey, B. and Priestley, H. (1990). *Introduction to lattices and order.* Cambridge University Press.

[Debrock et al., 1999] Debrock, G., Farkas, J., and Sarbo, J. (1999). Syntax from a Peircean perspective. In Sandrini, P., editor, *5th International Congress on Terminology and Knowledge Engineering*, pages 180–189, Innsbruck (Austria).

[Farkas et al., 1997] Farkas, J., Kamphuis, V., and Sarbo, J. (1997). Natural Language Concept Analysis. Technical Report CSI-R9717, University of Nijmegen.

[Farkas and Sarbo, 1999] Farkas, J. and Sarbo, J. (1999). A Peircean framework of syntactic structure. In *7th International Conference on Conceptual Structures (ICCS'99)*, number 1640 in Lecture Notes in Artificial Intelligence, pages 112–126, Blacksburg (VA).

[Huibers, 1996] Huibers, T. (1996). *An axiomatic theory for information retrieval.* PhD thesis, University of Nijmegen.

[Jolly, 1993] Jolly, J. (1993). Preposition Assignment in English. In Valin, V., editor, *Advances in Role and Reference Grammar*, pages 272–283, Amsterdam-Philadelphia. John Benjamins Publishing Company.

[Jones, 1993] Jones, K. S. (1993). What Might Be a Summary. In *Information Retrieval '93: von der Modellierung zur Anwendung.* Universitatsverlag Konstanz.

[Peirce, 1931] Peirce, C. (1931). *Collected Papers of Charles Sanders Peirce.* Harvard University Press, Cambridge.

[Sarbo, 1997] Sarbo, J. (1997). Building sub-knowledge bases using concept lattices. *The Computer Journal*, 39(10):868–875.

[Sarbo, 1999] Sarbo, J. (1999). Formal conceptual structure in language. In Dubois, D. M., editor, *Proceedings of Computing Anticipatory Systems (CASYS'98)*, pages 289–300, Woodbury, New York. AIP Conference Proceedings 465.

[Sarbo and Farkas, 1995] Sarbo, J. and Farkas, J. (1995). Knowledge representation and acquisition by concept lattices. In Markovitch, S., editor, *Proc. of the 11th Izraeli Symposium on Artificial Intelligence (ISAI'95)*, Hebrew University of Jerusalem, Izrael.