Interpreting Winograd Schemas via the SP Theory of Intelligence and its realisation in the SP Computer Model

J Gerard Wolff*

April 4, 2019

Abstract

In sentences in the form of a 'Winograd Schema' (WS), like *The city councilmen refused the demonstrators a permit because they feared violence*, it is easy for adults to understand what "they" refers to but it can be difficult for AI systems. This paper describes with examples how the *SP System*—the *SP Theory of Intelligence* and its realisation in the *SP Computer Model*—provides a distinctive new method for solving this kind of problem in the interpretation of natural language and in 'commonsense reasoning' and 'commonsense knowledge', three areas of interest in artificial intelligence. The central idea is that a knowledge of associations amongst linguistic features, including those that are scattered widely, and an ability to recognise such patterns of association, provide a robust means of determining what, in WS examples, a pronoun like "they" refers to. For any AI system to solve this kind of problem, it needs appropriate knowledge of relevant syntax and semantics which, ideally, it should learn for itself. Although the SP System has strengths in unsupervised learning, its capabilities in this area are not yet good enough to learn the kind of knowledge needed to interpret WS examples, so it must be supplied with such knowledge at the outset. However, its existing strengths in unsupervised learning suggest that it has potential to learn the necessary knowledge. By contrast with *ad hoc* solutions to WS problems, the SP System has been developed in an extended programme of research seeking to simplify and integrate observations and concepts across a broad canvass, and it has corresponding strengths in diverse aspects of AI. Its strengths and potential with WSs are bonuses that come with no added cost in terms of the complexity of the core model.

Keywords: Winograd Schema, information compression, SP Theory of Intelligence, SP Computer Model

1 Introduction

In a paper published in 1972 [21], Terry Winograd gave an example of a pair of sentences:

> *The city councilmen refused the demonstrators a permit because they feared violence*
> and
>
> *The city councilmen refused the demonstrators a permit because they advocated revolution*

*Dr Gerry Wolff, BA (Cantab), PhD (Wales), CEng, MIEEE; CognitionResearch.org, Menai Bridge, UK; jgw@cognitionresearch.org; +44 (0) 1248 712962; +44 (0) 7746 290775; *Skype*: gerry.wolff; *Web*: www.cognitionresearch.org.

which are easy for people to understand but are problematic for most AI systems—because the meaning of the word "they" in each sentence depends on a "sophisticated knowledge of councilmen, demonstrators, and politics" (p. 33).

Pairs of examples of natural language like this, known as *Winograd Schemas* (WSs), are seen to embody much of what is so challenging in the organisation of 'commonsense reasoning', with corresponding challenges in 'commonsense knowledge' [8], and also in the understanding of natural language. Accordingly, finding good general solutions to WS examples(WSEs)[1] is seen to be an important part of artificial intelligence—so much so that it has been suggested [6] that resolving ambiguities in WSEs might be an alternative to the Turing test for artificial intelligence [20], with advantages compared with that test.

This paper describes how the *SP System*—meaning the *SP Theory of Intelligence* and its realisation in the *SP Computer Model*—provides a distinctive new method for solving this kind of problem.

In a 'background' section, next: there is, first, more information about the concept of a Winograd Schema; then there is a selective review of related research; next, the SP System is introduced, with a link to Appendix A where a fuller description may be found; and finally there is a summary of distinctive features and advantages of the SP System.

The main body of the paper (Section 3) presents three pairs of WSEs, each with one or more demonstrations of how it may be interpreted by means of the SP System.

2 Background

2.1 More about Winograd Schemas

Since Levesque's paper [6], researchers have gathered together many examples of WSs, each of which, as in Winograd's original example [21, p. 33], contains two WSEs which differ in only one or two words, which contains an ambiguity that is resolved differently in the two cases, and which requires the use of world knowledge and reasoning to resolve the ambiguity. Some of those WSs may be seen on bit.ly/2MPm64B.

Important features of any WSE include:

- "It should be easily disambiguated by the human reader. Ideally, this should be so easy that the reader does not even notice that there is an ambiguity ..." [8, p. 557].

- It should not be possible to interpret a candidate WSE correctly using "cheap tricks" *aka* "heuristics" [7, Section 2.2]. For example, with a question like *Could a crocodile run a steeplechase?*, we can say "no" because we have never heard of such a thing, without bothering to consider the physical demands of a steeplechase, the short legs and aquatic habits of crocodiles, and so on.

  In a similar way, with a pair of sentences like these: *The women stopped taking the pills because they were [pregnant/carcinogenic]*, the meaning of "they" is obvious in each case because only women can be pregnant, and only pills would be carcinogenic [6, Section 'Pitfall 1'].

- The test should be *Google-proof*. With respect to the previous example, Levesque says "In linguistics terminology, the anaphoric reference can be resolved using selectional restrictions alone. Because selectional restrictions like this might be learned by sampling a large enough

[1]The expression 'WS example' has been used, rather than 'WS sentence', because some examples comprise two or more sentences.

corpus (that is, by confirming that the word "pregnant" occurs much more often close to "women" than close to "pills"), we should avoid this sort of question." (*ibid.*).

- There should not be too much ambiguity about the answers, as, for example, in: *Frank was pleased when Bill said that he was the winner of the competition.* Here, the word "he" could refer equally well to Frank or Bill [6, Section 'Pitfall 2'].

A comprehensive solution to the problems posed by WSEs depends on two things: 1) An ability to learn relevant syntax and semantics, including knowledge of the world like the "sophisticated knowledge of councilmen, demonstrators, and politics" noted by Winograd, and 2) An ability to make use of that syntactic and semantic knowledge in the interpretation of WSEs.

For the avoidance of any misunderstanding, this paper concentrates on the second aspect of the problem, although it does have something to say (in Sections 2.3 and 7, and in Appendix A.3), about the learning of syntax and semantics and how that learning relates to WSEs and their interpretation.

2.2 Related research

This section presents a selective review of research on the interpretation of WSs, concentrating on more recent studies.

As previously mentioned, Hector Levesque [6] presents an alternative to the Turing Test that involves responding to typed English sentences, like the Turing Test. But, unlike the Turing Test, the AI taking the test is not required to engage in a conversation and fool an interrogator into believing that he or she is dealing with a person. Instead, this task requires the AI to disambiguate WSEs. Levesque suggests that it is reasonable to believe that: "with a very high probability, anything that answers [a WSE problem] correctly is engaging in behaviour that we would say shows thinking in people." and he adds: "Whether or not a subject that passes the test is really and truly thinking is the philosophical question that Turing sidesteps." (*ibid.*, Discussion and Conclusion).

There is more about WSs and associated issues in [8]. In that paper, the authors make a connection between the study of WSs and the study of commonsense knowledge (and, by implication, commonsense reasoning): "... we believe that in order to pass the WS Challenge [(WSC)], a system will need to have commonsense knowledge about space, time, physical reasoning, emotions, social constructs, and a wide variety of other domains. Indeed, we hope that the [WSC] will spur new research into representations of commonsense knowledge." (*ibid.*, p. 558). But they add: "However, nothing in the [WSC] insists on this approach, and we would expect researchers [in natural language processing] to try different approaches." (*ibid.*).

Leora Morgenstern and Charles Ortiz [10] describe the 'WSC competition', now run roughly once a year and sponsored by Nuance Communications, Inc., to encourage efforts to develop programs that can correctly interpret WSEs. The first of these events was run on July 11, 2016, at the International Joint Conference on Artificial Intelligence (IJCAI-16). "The first round of the challenge was a collection of 60 PDPs [Pronoun Disambiguation Problems]. The highest score achieved was 58% correct, by Quan Liu, from University of Science and Technology, China. Hence, by the rules of that challenge, no prizes were awarded, and the challenge did not proceed to the second round." (from bit.ly/2R4gSFJ).

Altaf Rahman and Vincent Ng [11] examine the task of determining ('resolving') the meanings of pronouns in sentences where there are no obvious clues from the syntax—much as in the WSC. They present results from a "Combined Resolver"—a combination of the "Stanford resolver" and the "Baseline Ranker"—which "significantly outperforms state-of-the-art resolvers" but has "a lot of room for improvement" (*ibid.*, Section 5.4).

In a paper entitled "On our best behaviour", Hector Levesque [7] discusses some general issues relating to the WSC including: "... what does it tell us when a good semblance of a behaviour can be achieved using cheap tricks that seem to have little to do with what we intuitively imagine intelligence to be?" and "... are the philosophers right, and is a behavioural understanding of intelligence simply too weak?" He suggests that both those ideas are wrong and that "... we should put aside any idea of tricks and short cuts, and focus instead on what needs to be *known*, how to represent it symbolically, and how to use the representations." (*ibid.*, p. 34, emphasis in the original). Two main hurdles need to be overcome: much of what we know comes via language, but we need knowledge to interpret language; and even at the level of what is known by children, there appears to be much complexity and heavy computational demands.

With respect to the WSC, Peter Schüller [14] considers pairs of sentences like these: *Sam's drawing was hung just above Tina's and it did look much better with another one [below / above] it.* Here, people naturally assume that "another one" is Sam's drawing or Tina's, although it could be a third drawing by someone else. He suggests that what people naturally assume may be explained in terms of "Relevance Theory" and develops these ideas with a simplified version of Roger Schank's graph framework for natural language understanding.

Daniel Bailey and colleagues [1], in discussing the problem of interpreting WSEs, say that "we treat coreference resolution as a by-product of a general process of establishing discourse coherence: a resolution for a pronoun is acceptable if it makes the discourse 'coherent'." (p. 17). Later, they explain that, in a sentence like *Joan made sure to thank Susan for all the help she had given*, the phrases *Joan made sure to thank Susan* and *Susan had given help* "are correlated in the sense that either one would cause the hearer to view the other as more plausible." (p. 18). They go on to present a "correlation calculus" for representing and reasoning about such relationships which, although it is not founded on statistical concepts of correlation, "Nevertheless, our correlation calculus turned out to be closely related to correlation in the sense of probability theory ..." (p. 18).

Ali Emami and colleagues [3] propose a three-stage knowledge hunting method for interpreting any given WSE: 1) Perform a partial parse of the WSE to isolate the main elements of the meaning; 2) Generate queries to send to a search engine in order to extract text snippets that resemble the given WSE; 3) Obtain a set of text snippets that resemble the given WSE, and use this information to resolve the ambiguity in the WSE. They say that their method compares well with AI alternatives but is not as good as a person.

Adam Richard-Bollans and colleagues [12] consider the WSC as a means for highlighting problems in the field of commonsense reasoning. Approaches that they discuss include machine learning approaches and 'commonsense rules'. Key challenges include: pragmatics (extra-linguistic factors, such as context, and how they allow the understanding of a speaker's intended meaning), assumptions about the world (e.g., if we throw something down to someone, that person must be below you), and the level of detail or vagueness that may be needed in formalising commonsense knowledge. They propose an alternative approach, with knowledge bases for the meanings of commonsense terms and the use of heuristics based on pragmatics.

Arpit Sharma and colleagues [16, 17] present an approach that identifies the knowledge needed to answer a challenge question, hunts down that knowledge from text repositories, and then reasons with the knowledge and the question to come up with an answer. This research includes the development of a semantic parser. They show that their approach works well on a subset of Winograd schemas.

2.3 The *SP System*

The *SP System*—meaning the *SP Theory of Intelligence* and its realisation in the *SP Computer Model*[2]—has been under development since about 1987, with a break between early 2006 and late 2012. There is an outline of the SP System in Appendix A. Since an understanding of the evidence and arguments presented in this paper depend on an understanding of the workings of the SP System, *readers who are not already familiar with the system are urged to read Appendix A before reading the rest of the paper.*

As noted in Appendix A, the overarching goal in the development of the SP System has been to create a system that can simplify and integrate observations and concepts across artificial intelligence, mainstream computing, mathematics, and human learning, perception, and cognition. To a large extent, this quest has been successful, as outlined in Appendix A.5 and summarised in Section 2.4.

As indicated in Section 2.1, the main focus of this paper is on how the SP System may function in the interpretation of WSEs, with relatively little to say about the learning of relevant syntax and semantics. However, for reasons given in Appendix A.3, it is anticipated that, when the learning capabilities of the SP Computer Model have been further developed for the unsupervised learning of natural language syntax, those capabilities will generalise with little or no modification to the learning of semantic structures and the learning of structures in which syntax and semantics are integrated.

2.4 Distinctive features and advantages of the SP System

2.4.1 Evaluation of the SP System in terms of 'Simplicity' and 'Power'

A major strength of the SP System is the way that it combines conceptual *Simplicity* with high levels of explanatory and descriptive *Power* (Appendices A.1 and A.5). This has a bearing on how the system may be evaluated in terms of its performance in the interpretation of WSEs.

As we shall see with the examples in Section 3, the SP System, with appropriate data, can provide robust interpretations of WSEs, in accordance with what people judge to be correct. These interpretations are achieved with the SP Computer Model as it was developed to meet its original objectives, and without any adaptation or modification of any kind.

This means that, in terms of 'Simplicity' and 'Power' (Appendix A.1), the discovery that the SP System can achieve robust interpretations of WSEs without any adaptation or modification means that the simplicity of the system is the same as before, but (our knowledge of) its descriptive and explanatory power has been increased. Thus, in terms of Ockham's razor, and in terms of our knowledge of the system and what it can do, the SP System is more plausible than before.

Because of its strengths in terms of 'Simplicity' and 'Power' in AI-related attributes, the SP System has an apparent advantage compared with any *ad hoc* solution to the problem of interpreting WSEs. And for the same reasons, it appears that the SP System provides a much firmer foundation for the development of general, human-level AI than any alternative, including 'unified theories of cognition' such as John Laird's *Soar* [5] and attempts to develop 'artificial general intelligence' such as Ben Goertzel's *CogPrime* [4].[3]

[2] Source code for the SP Computer Model, and Windows executable code, may be downloaded via a link near the bottom of bit.ly/1mSs5XT.

[3] In the light of Ben Goertzel's [4, p. 1] remark that, in the search for 'artificial general intelligence' (AGI), "We have not discovered any one algorithm or approach capable of yielding the emergence of [general intelligence]." it seems that there is still some way to go.

2.4.2 Other distinctive features and advantages of the SP System

Other distinctive features and advantages of the SP System are described in [30]. Features that are especially distinctive are: 1) That all kinds of processing are achieved via the matching and unification of patterns; and, more specifically, 2) All kinds of processing are achieved via the building of SP-multiple-alignments, described in Appendix A.2.

3 Examples of how the SP System may disambiguate WSEs

The three sections that follow present three WSs, where each WSE has one or more SP-multiple-alignments produced by the SP Computer Model that demonstrates how the ambiguity in the WSE may be resolved.

A feature of the SP Computer Model is that it has a tendency to produce SP-multiple-alignments that are very large and cannot easily be fitted on one page. Quite often it is possible to work around this problem by choosing suitable examples or splitting SP-multiple-alignments into two or more parts. But with these examples, even with the choice of relatively short WSEs, those work-arounds are not possible—so it has been necessary to do something different.

The solution that has been adopted with these examples is that each SP-multiple-alignment has been produced at full size in vector-graphic form in a PDF file, and then shrunk to a size that will fit on a page. Then, providing each SP-multiple-alignment is viewed on a computer, it may be magnified to whatever size makes it legible, and it will be sharp at any size because of the vector-graphic encoding. For the convenience of readers, the figures are available in a separate file ('sp_ws_figures.pdf'), in addition to their display in the body of this paper. That means that the figures in the separate file may be magnified without magnifying the main body of the paper. The package 'hyperref' may provide a better solution using thumbnails, each with a link to a corresponding image, but this appears to be complicated to set up and may not be very reliable.

A point to note about these examples is that, although the words 'syntax' and 'semantics' are used in discussing the examples, the SP System makes no formal distinction between the two. In the SP System, both 'syntax' and 'semantics' are information and may be combined flexibly without reference to any formal or informal classification we may apply to them.

It should be stressed again that, up to now in this research, there has been no attempt to apply any kind of learning processes. In each of the examples to be described, the New SP-pattern, and all the Old SP-patterns, have been supplied ready-made to the SP Computer Model (see also Section 7). It is unlikely that a deep neural network or any other system, could learn the necessary knowledge.

It is likely that there is some room for debate about the details of these examples. But any such concerns should not distract from the main point of the examples: that within the framework of the SP system, and with WSEs containing a pronoun that requires disambiguation, it is relatively straightforward to discover the referent of such a pronoun.

All the WSEs in what follows contain at least one capital letter. But in the sentences provided as New patterns for the SP Computer Model, all capital letters have been replaced with their lower-case equivalents. This is to by-pass rules for capitalisation which might otherwise be a distraction from the main point of these examples: how to find the referent for the pronoun in each example.

A final point in this section is that a key idea in all the examples is the way in which, within the SP-multiple-alignment framework, it is quite straight forward to identify correlations or 'dependencies' between one or more parts of a syntactic/semantic structure, and which may be 'discontinuous' in the sense that they may jump over arbitrarily large amounts of intervening structure without being disturbed by any such intervening structure and without disturbing any such structure. This principle is exactly the same as discussed in Appendix A.2 and illustrated in Figure 10.

4 The city councilmen refused the demonstrators a permit

This section is concerned with Terry Winograd's [21, p. 33] two example sentences:[4]

> *The city councilmen refused the demonstrators a permit because they feared violence*, and
>
> *The city councilmen refused the demonstrators a permit because they advocated revolution*

In each case, the sentence is supplied as a New SP-pattern to the SP Computer Model together with a collection of Old SP-patterns which describe the syntax and semantics of the given sentence and other sentences of that kind.

4.1 The city councilmen ... feared violence

With the first sentence, the best SP-multiple-alignment created by the SP Computer Model is shown in Figure 1. Here, the word 'best' means that, out of the many SP-multiple-alignments created by the SP Computer Model, the given SP-multiple-alignment is the one in which the New SP-pattern may be encoded most economically in terms of the Old SP-patterns in that SP-multiple-alignment, as described in [24, Section 4.1] and [22, Section 3.5].

The SP-multiple-alignment shown in Figure 1 is, if effect, a parsing of the sentence *the city councilmen refused the demonstrators a permit because they feared violence*, much like the parsing in Figure 10, except that SP-symbols in the sentence are whole words instead of letters as in the sentence in Figure 10 and, as described in Appendix B, the SP-multiple-alignment in Figure 1, like other examples in this paper, attempts to marry syntax with semantics. It has been moved into Appendix B because it is quite tentative and is not relevant to the main substance of this paper—how to disambiguate pronouns in WSEs. Readers may safely ignore it if they wish.

4.2 Finding what 'they' refers to

Of course, the main point of interest with this example is how it determines the referent for the word "they" in *the city councilmen refused the demonstrators a permit because they feared violence.* This disambiguation is achieved via the SP-pattern '`PEACE_LOVING PN #PN v1 n0`' which appears in column 21 in Figure 1. The connection is made correctly because, within that SP-pattern:

- The SP-symbol '`PEACE_LOVING`' is aligned with the same symbol in the SP-pattern '`N n3 councilmen COUNCILMEN PEACE_LOVING #N`' in column 14, which, in the light of the next bullet point, identifies '`councilmen`' and its associated meaning as the referent for "they" in the given sentence.

[4] Apart from one difference, this is the first example in the 'Collection of Winograd Schemas', compiled by Ernest Davis and colleagues, and shown on bit.ly/2MPm64B, retrieved 2018-09-26. The difference is to use exactly the same two sentences presented by Winograd [21, p. 33] which, at their ends, have ['feared violence'/'advocated revolution'] instead of ['feared violence'/'advocated violence'].

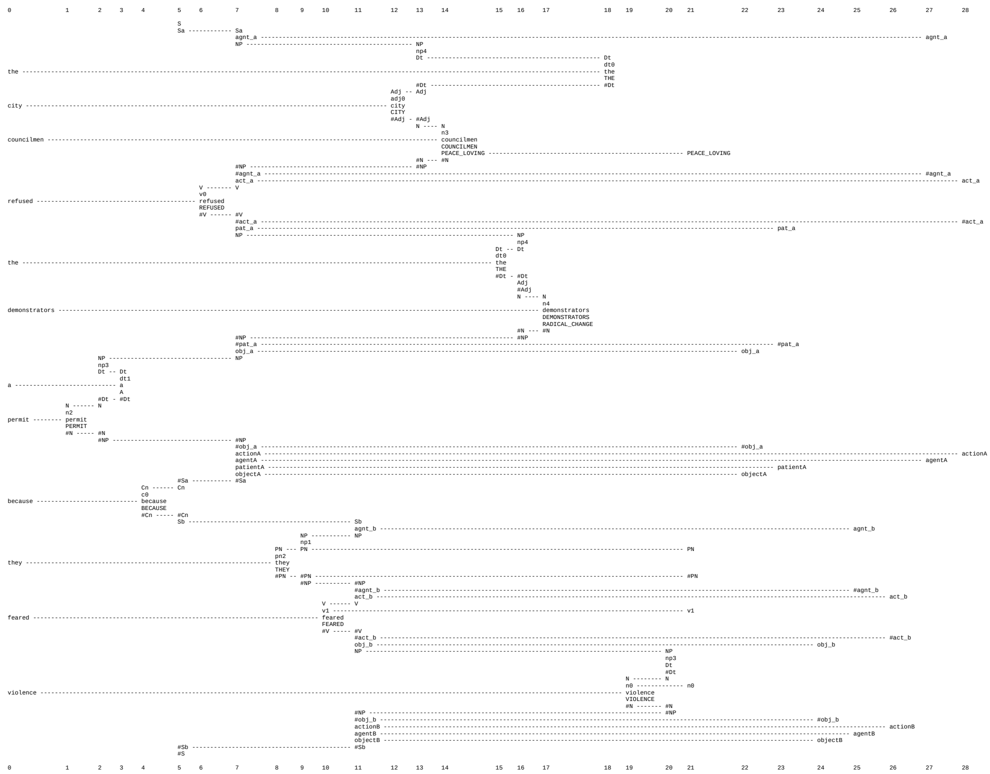

Figure 1: The best SP-multiple-alignment produced by the SP Computer Model with *the city councilmen refused the demonstrators a permit because they feared violence* as a New SP-pattern, which appears in column 0, and a collection of Old SP-patterns as described in the text, some of which are shown in columns 1 to 28, one SP-pattern per column.

- The SP-symbols '`PN #PN`' are aligned with the matching SP-symbols in '`NP np1 PN #PN #NP`' (column 9) and in '`PN pn2 they THEY #PN`' (column 8), thus identifying the pronoun "they".

- The SP-Symbol '`v1`' matches the same symbol in '`V v1 feared FEARED #V`' in column 10, thus identifying the word "feared" and its associated meaning.

- And the SP-Symbol '`n0`' matches the same symbol in '`N n0 violence VIOLENCE #N`' in column 19, thus identifying the word "violence" and its associated meaning.

In short, within the SP-multiple-alignment shown in Figure 1, the SP-pattern '`PEACE_LOVING PN #PN v1 n0`' has the effect of highlighting the association between peace-loving councilmen, the pronoun 'they', and the councilmen's probable fear of violence. As with the examples in Sections 5 and 6, this works because of the way in which dependencies, which may be discontinuous, may be marked within an SP-multiple-alignment, as described in Appendix A.2 and illustrated in Figure 10.

Possible variations on this scheme are described in Section 4.3, and the potential role for unsupervised learning is described in Section 7.

4.3 Variations

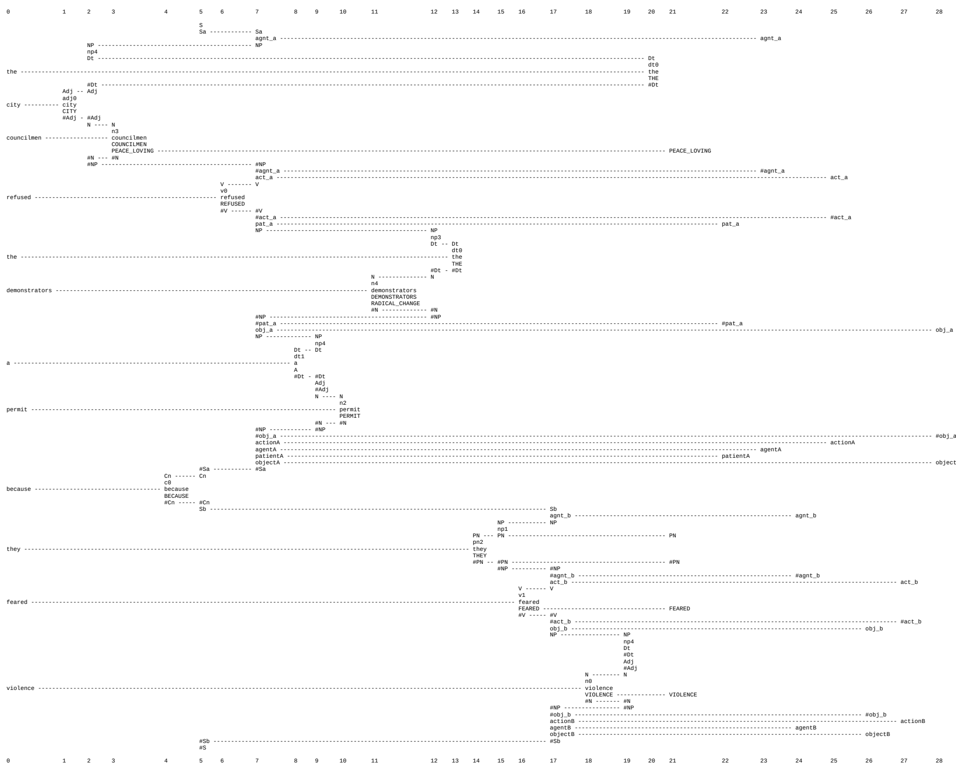

Figure 2: The best SP-multiple-alignment created by the SP Computer Model with Old SP-patterns producing a variant of Figure 1 as described in the text.

As mentioned near the beginning of Section 3, there is plenty of room for debate about the grammatical details of these examples. Much the same goes for the proposed solution to the WS problem where several variations are possible. But, as before, any such variations should not distract from the key point: that, within the framework of the SP System, and with appropriate SP-patterns, finding the referent for a pronoun in a WSE is relatively straightforward.

As an illustration of one possible variant of the solution described in Section 4.2, Figure 2 shows how the ... *feared violence* sentence may be disambiguated via the creation of an SP-multiple-alignment, much as in Figure 1, but with the SP-pattern '`PEACE_LOVING PN #PN FEARED VIOLENCE`' in column 21 instead of '`PEACE_LOVING PN #PN v1 n0`'.

Another possible variant is shown in Figure 3. Here, Old SP-patterns have been supplied to the SP Computer Model which, in a simple way, introduce the concept of class-inclusion hierarchy and the inheritance of attributes.

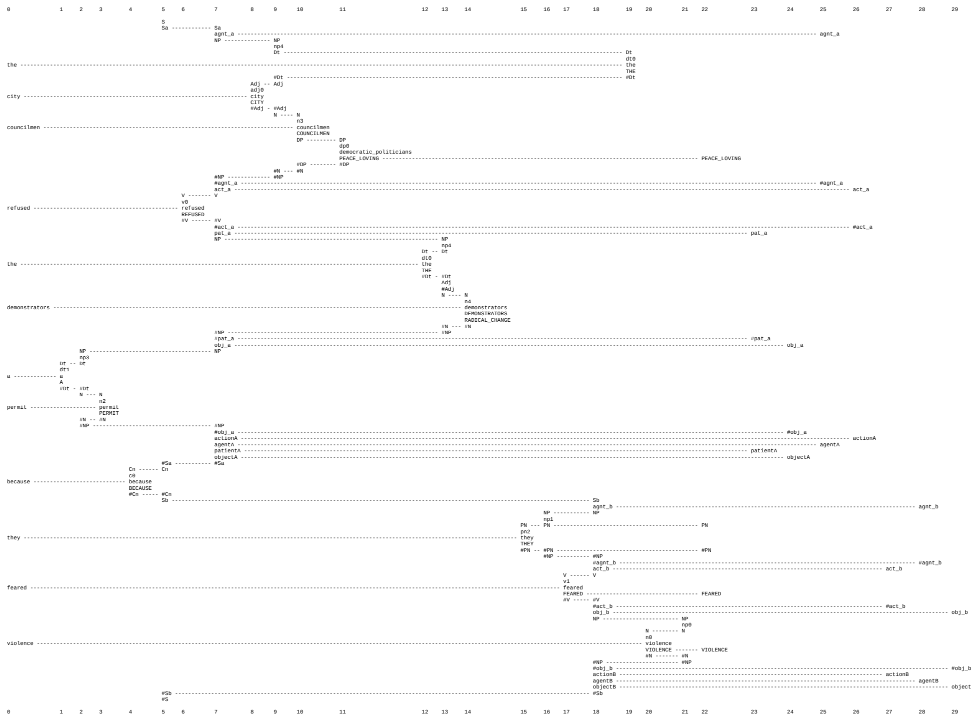

Figure 3: The best SP-multiple-alignment created by the SP Computer Model with Old SP-patterns producing a variant of Figure 2 as described in the text.

In the figure, the '`PEACE_LOVING`' attribute, contrary to fact, appears in a new SP-pattern for 'democratic politicians', '`DP dp0 democratic_politicians PEACE_LOVING #DP`', in column 11. And '`PEACE_LOVING`' is aligned with the same symbol in the SP-pattern '`PEACE_LOVING PN #PN FEARED VIOLENCE`' in column 22, much as in Figure 2. The new SP-pattern, for 'democratic politicians' connects with a modified SP-pattern for 'councilmen', '`N n3 councilmen COUNCILMEN DP #DP #N`', in column 10. The overall effect is that, in the manner of object-oriented programming, the class 'councilmen' inherits the 'peace loving' attribute from the more general class 'democratic politicians'.

4.4 The demonstrators ... advocated revolution

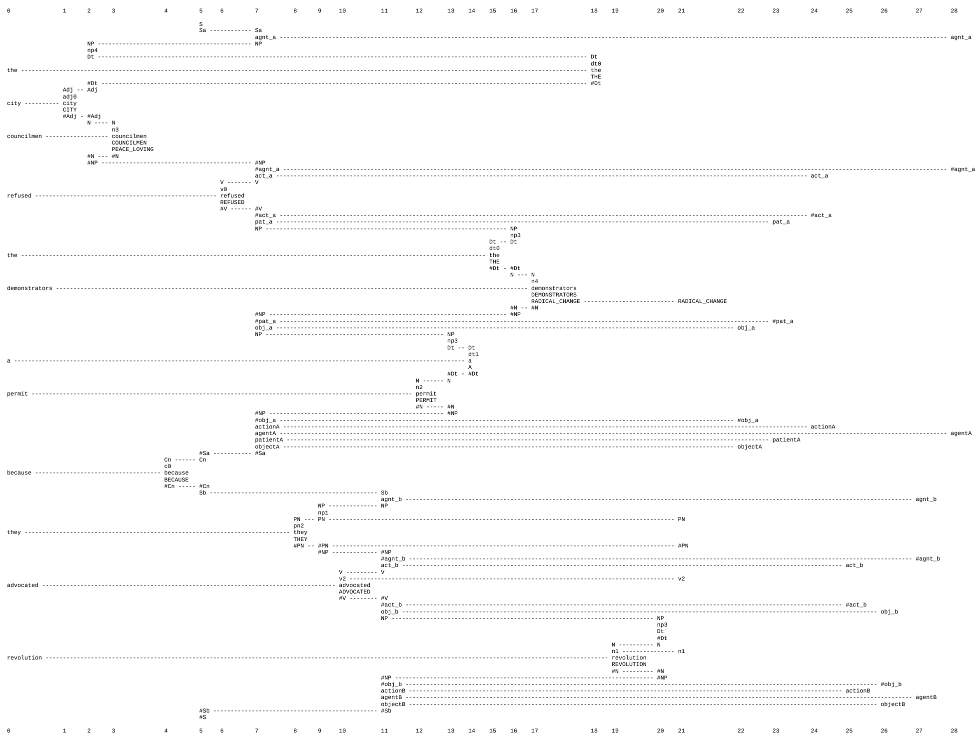

Figure 4: The best SP-multiple-alignment produced by the SP Computer Model with *the city councilmen refused the demonstrators a permit because they advocated revolution* as a New SP-pattern, which appears in column 0, and a collection of Old SP-patterns as described in the text, some of which are shown in columns 1 to 28, one SP-pattern per column.

The second of Winograd's [21] WSEs, *The city councilmen refused the demonstrators a permit because they advocated revolution*, may be processed in a similar way. Figure 4 shows the best SP-multiple-alignment created by the SP Computer Model with the given sentence as the New pattern and the same set of Old SP-patterns as before. This time, the SP-pattern '`RADICAL_CHANGE PN #PN v2 n1`', which appears in column 21, has the effect, in much the same manner as before, of highlighting the association between the demonstrators and their probable interest in radical change, the pronoun 'they', and the demonstrator's advocacy of revolution.

5 Pete envies Martin

The second pair of sentences to be considered are: *Pete envies Martin [because/although] he is very successful.*[5]

[5]This is the 20th example in the 'Collection of Winograd Schemas', compiled by Ernest Davis and colleagues, and shown on bit.ly/2MPm64B, retrieved 2018-09-26.

5.1 Pete envies Martin because ...

Figure 5 shows the best SP-multiple-alignment produced by the SP Computer Model with *pete envies martin because he is very successful* as a New SP-pattern, which appears in column 0, and a collection of Old SP-patterns describing the syntax and semantics of that kind of sentence, some of which appear in columns 1 to 22, one SP-pattern per column.

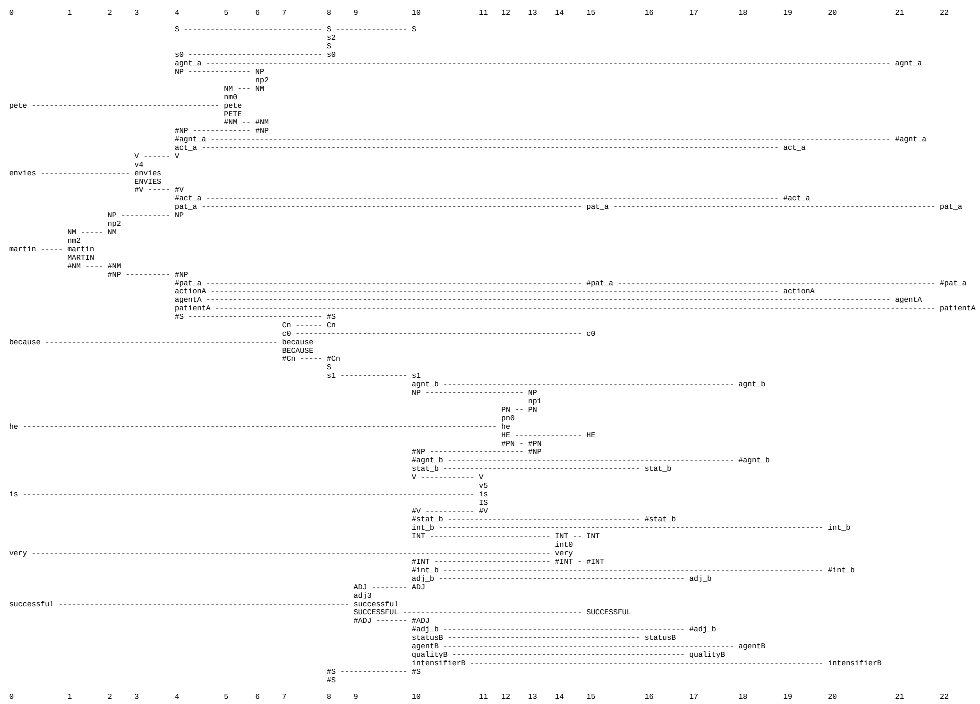

Figure 5: The best SP-multiple-alignment produced by the SP Computer Model with *pete envies martin because he is very successful* as a New SP-pattern, which appears in column 0, and a collection of Old SP-patterns as described in the text, some of which are shown in columns 1 to 22, one SP-pattern per column.

In a manner much like the examples in Section 4, disambiguation is achieved via an SP-pattern that connects relevant parts of the SP-multiple-alignment. Here, that SP-pattern is ‘`pat_a #pat_a c0 HE INT #INT SUCCESSFUL`’ which appears in column 15 in the figure. In that SP-pattern:

- The pair of SP-symbols ‘`pat_a #pat_a`’, which are mnemonic for ‘patient’, are aligned with matching symbols in the SP-pattern in column 4 which span the structure in columns 2, 1, and 0, which identify ‘`martin`’ as having the semantic role of ‘patient’ in the given sentence.
- The SP-symbol ‘`c0`’ is aligned with the matching SP-symbol in column 7, thus identifying the special word ‘because’ in ‘`Cn c0 because BECAUSE #Cn`’.
- The SP-symbol ‘`HE`’ is aligned with the matching SP-symbol within the SP-pattern ‘`PN pn0 he HE #PN`’ in column 12.

- The pair of SP-symbols '`INT #INT`' are aligned with matching SP-symbols within the SP-pattern '`INT int0 very #INT`' in column 14.
- And the SP-symbol '`SUCCESSFUL`' is aligned with the matching SP-symbol within the SP-pattern '`ADJ adj3 successful SUCCESSFUL #ADJ`' in column 9.

In short, the SP-pattern '`pat_a #pat_a c0 HE INT #INT SUCCESSFUL`' has the effect of identifying 'martin' as the referent of 'he' within the given sentence, with the other SP-symbols within the SP-pattern, especially the special word 'because', providing the necessary context. As with the examples in Sections 4 and 6, this works because of the way in which dependencies, which may be discontinuous, may be marked within an SP-multiple-alignment, as described in Appendix A.2 and illustrated in Figure 10.

5.2 Pete envies Martin although ...

Figure 6 shows the best SP-multiple-alignment produced by the SP Computer Model with *pete envies martin although he is very successful* as a New SP-pattern, which appears in column 0, and a collection of Old SP-patterns describing the syntax and semantics of that kind of sentence, some of which appear in columns 1 to 22, one SP-pattern per column.

```
0            1      2     3        4          5        6     7          8     9             10             11   12        13    14     15           16        17        18            19        20          21          22

                                                                        S ----------------- S
                                                                        s2
                                   S ---------------------------------- S
                                   s0 --------------------------------- s0
                                   agnt_a -------------------------------------------------------------------------------------------- agnt_a ------------------------------------------------------------------------ agnt_a
                    NP ----------- NP
                    np2
             NM --- NM
             nm1
pete ------- pete
             PETE
             #NM -- #NM
                    #NP ---------- #NP
                                   #agnt_a ------------------------------------------------------------------------------------------- #agnt_a ----------------------------------------------------------------------- #agnt_a
                                   act_a -------------------------------------------------------------------------------------------------------------------------------------------- act_a
                          V ------ V
                          v4
envies ------------------ envies
                          ENVIES
                          #V ----- #V
                                   #act_a ------------------------------------------------------------------------------------------------------------------------------------------- #act_a
                                   pat_a ------------------------------------------------------------------------------------------------------------------------------------------------------------------ pat_a
                                   NP ---------------- NP
                                                       np2
                                              NM ----- NM
                                              nm2
martin -------------------------------------- martin
                                              MARTIN
                                              #NM ---- #NM
                                   #NP --------------- #NP
                                   #pat_a ----------------------------------------------------------------------------------------------------------------------------------------------------------------- #pat_a
                                   actionA ------------------------------------------------------------------------------------------------------------------------------------------ actionA
                                   agentA ---------------------------------------------------------------------------------------------------------------------------------------------------------------------------- agentA
                                   patientA --------------------------------------------------------------------------------------------------------------------------------------------------------------- patientA
                                   #S --------------------------------- #S
                                                             Cn ------- Cn
                                                             c2 ---------------------------------------------------------------------- c2
although --------------------------------------------------- although
                                                             ALTHOUGH
                                                             #Cn ------ #Cn
                                                                        S
                                                                        s1 ---------------- s1
                                                                                            agnt_b ------------------------------------------------ agnt_b
                                                                                            NP -------------------------- NP
                                                                                                                          np1
                                                                                                                PN ------ PN
                                                                                                                pn0
he ------------------------------------------------------------------------------------------------------------ he
                                                                                                                HE ------------------- HE
                                                                                                                pronoun
                                                                                                                #PN ----- #PN
                                                                                            #NP ------------------------- #NP
                                                                                            #agnt_b ----------------------------------------------- #agnt_b
                                                                                            stat_b ---------------------------------------------------------- stat_b
                                                                                            V ------------ V
                                                                                                           v5
is ------------------------------------------------------------------------------------------------------- is
                                                                                                           IS
                                                                                            #V ----------- #V
                                                                                            #stat_b --------------------------------------------------------- #stat_b
                                                                                            int_b --------------------------------------------------------------------- int_b
                                                                                            INT ------------------------------- INT -- INT
                                                                                                                                int0
very -------------------------------------------------------------------------------------------------------------------------- very
                                                                                            #INT ------------------------------ #INT - #INT
                                                                                            #int_b -------------------------------------------------------------------- #int_b
                                                                                            adj_b --------------------------------------------------------------------------------------------- adj_b
                                                                              ADJ --------- ADJ
                                                                              adj3
successful ------------------------------------------------------------------ successful
                                                                              SUCCESSFUL --------------------------------------------- SUCCESSFUL
                                                                              #ADJ -------- #ADJ
                                                                                            #adj_b -------------------------------------------------------------------------------------------- #adj_b
                                                                                            statusB --------------------------------------------------------- statusB
                                                                                            agentB ------------------------------------------------ agentB
                                                                                            qualityB ------------------------------------------------------------------------------------------ qualityB
                                                                                            intensifierB -------------------------------------------------------------- intensifierB
                                                                        #S ---------------- #S
                                                                        #S

0            1      2     3        4          5        6     7          8     9             10             11   12        13    14     15           16        17        18            19        20          21          22
```

Figure 6: The best SP-multiple-alignment produced by the SP Computer Model with *pete envies martin although he is very successful* as a New SP-pattern, which appears in column 0, and a collection of Old SP-patterns as described in the text, some of which are shown in columns 1 to 22, one SP-pattern per column.

As before, there is a key SP-pattern which identifies the referent for the pronoun 'he' in the given sentence. Here, that SP-pattern is '`agnt_a #agnt_a c2 HE INT #INT SUCCESSFUL`' which appears in column 15 of the figure. By contrast with the example in Section 5.1, the pair of SP-symbols '`agnt_a #agnt_a`' are aligned with matching symbols in column 4 which have the effect of marking '`pete`' as an object of interest. In much the same way as in the example in Section 5.1, '`pete`' is identified as the referent of '`HE`'.

6 The fish ate the worm

The third example to be considered is: *The fish ate the worm. It was [tasty/hungry].*[6]

6.1 The worm ... tasty

Figure 7 shows the best SP-multiple-alignment produced by the SP Computer Model with *the fish ate the worm it was tasty* as a New SP-pattern, which appears in column 0, and a collection of Old SP-patterns describing the syntax and semantics of the two target sentences. Some of those SP-patterns are shown in columns 1 to 21, one SP-pattern per column.

As readers may guess, the solution in this case is similar to what has presented in previous examples. In this case, the key SP-pattern is '`PREY PN #PN adj2`'. The first SP-symbol, '`PREY`' matches the same SP-symbol in the SP-pattern '`N n1 worm WORM PREY #N`' in column 1, thus marking 'worm' as an object of interest. The pair of SP-symbols '`PN #PN`' match the same two SP-symbols in column 12 and in the SP-pattern '`PN pn2 it IT #PN`' thus identifying 'it' as a pronoun. And the SP-symbol '`adj2`' selects the SP-pattern '`ADJ adj2 tasty TASTY #ADJ`.

In short, the SP-pattern '`PREY PN #PN adj2`' tells us that the 'worm' is the referent of the pronoun 'it' because, as a kind of prey for animals like fish, it can be 'tasty'. As with the examples in Sections 4 and 5, this works because of the way in which dependencies, which may be discontinuous, may be marked within an SP-multiple-alignment, as described in Appendix A.2 and illustrated in Figure 10.

6.2 The fish ... hungry

Figure 8 shows the best SP-multiple-alignment produced by the SP Computer Model with *the fish ate the worm it was hungry* as a New SP-pattern, which appears in column 0, and a collection of Old SP-patterns describing the syntax and semantics of the two target sentences. Some of those SP-patterns are shown in columns 1 to 21, one SP-pattern per column.

In much the same way as in Section 6.1, the SP-pattern '`PREDATOR PN #PN adj1`' makes a connection between the 'fish', the pronoun 'it' and the fact that the fish was 'hungry'.

[6]This is the 52nd example in the 'Collection of Winograd Schemas', compiled by Ernest Davis and colleagues, and shown on bit.ly/2MPm64B, retrieved 2018-09-26.

```
0       1      2     3     4          5     6     7         8     9         10        11    12    13    14      15         16    17         18        19        20        21

                                                                                      S
                                                                                      s0
                                                  Sa -------------------------------- Sa
                                                  sa0
                                                  agnt_a -------- agnt_a
                                      NP -------- NP
                                      np3
                                      Dt ------------------ Dt
                                                            dt1
the ------------------------------------------------------- the
                                                            THE
                                      #Dt ----------------- #Dt
                           N -------- N
                           n0
fish --------------------- fish
                           FISH
                           PREDATOR
                           #N ------- #N
                                      #NP ------- #NP
                                                  #agnt_a ------- #agnt_a
                                                  act_a ----------------------------------------------------------------------------------- act_a
                                            V --- V
                                            v3
ate --------------------------------------- ate
                                            ATE
                                            #V -- #V
                                                  #act_a ---------------------------------------------------------------------------------- #act_a
                                                  obj_a ------------------- obj_a
               NP ------------------------------- NP
               np3
               Dt -- Dt
                     dt1
the ---------------- the
                     THE
               #Dt - #Dt
        N ---- N
        n1
worm -- worm
        WORM
        PREY ------------------------------------------------------------------------------------------------------------------------------------------------------------ PREY
        #N --- #N
               #NP ------------------------------ #NP
                                                  #obj_a ------------------ #obj_a
                                                  actionA --------------------------------------------------------------------------------- actionA
                                                  agentA -------- agentA
                                                  objectA ----------------- objectA
                                                  #Sa ------------------------------- #Sa
                                                                                      Sa ---------------------- Sa
                                                                                                                sa1
                                                                                                                agnt_b ---------------------------------------- agnt_b
                                                                                                  NP ---------- NP
                                                                                                  np1
                                                                                            PN -- PN -------------------------------------------------------------------- PN
                                                                                            pn2
it ---------------------------------------------------------------------------------------- it
                                                                                            IT
                                                                                            #PN - #PN ------------------------------------------------------------------- #PN
                                                                                                  #NP --------- #NP
                                                                                                                #agnt_b --------------------------------------- #agnt_b
                                                                                                                stat_b ------------------------------ stat_b
                                                                                                                V -------- V
                                                                                                                           v1
was ------------------------------------------------------------------------------------------------------------------------ was
                                                                                                                           WAS
                                                                                                                #V ------- #V
                                                                                                                #stat_b ----------------------------- #stat_b
                                                                                                                adj_b ---------- adj_b
                                                                                                        ADJ --- ADJ
                                                                                                        adj2 ------------------------------------------------------------ adj2
tasty --------------------------------------------------------------------------------------------------- tasty
                                                                                                        TASTY
                                                                                                        #ADJ -- #ADJ
                                                                                                                #adj_b --------- #adj_b
                                                                                                                statusB ----------------------------- statusB
                                                                                                                agentB ---------------------------------------- agentB
                                                                                                                qualityB ------- qualityB
                                                                                      #Sa --------------------- #Sa
                                                                                      #S

0       1      2     3     4          5     6     7         8     9         10        11    12    13    14      15         16    17         18        19        20        21
```

Figure 7: The best SP-multiple-alignment produced by the SP Computer Model with *the fish ate the worm it was tasty* as a New SP-pattern, which appears in column 0, and a collection of Old SP-patterns as described in the text, some of which are shown in columns 1 to 21, one SP-pattern per column.

```
0        1     2     3      4          5     6     7         8     9         10        11    12    13    14    15         16       17         18        19        20        21

                                                                                       S
                                                                                       s0
                                                   Sa -------------------------------- Sa
                                                   sa0
                                                   agnt_a -------- agnt_a
                                       NP -------- NP
                                       np3
                                       Dt ------------------ Dt
                                                             dt1
the -------------------------------------------------------- the
                                                             THE
                                       #Dt ----------------- #Dt
                            N -------- N
                            n0
fish ---------------------- fish
                            FISH
                            PREDATOR -------------------------------------------------------------------------------------------------------------------------------------- PREDATOR
                            #N ------- #N
                                       #NP ------- #NP
                                                   #agnt_a ------- #agnt_a
                                                   act_a ------------------- act_a
                                             V --- V
                                             v3
ate ---------------------------------------- ate
                                             ATE
                                             #V -- #V
                                                   #act_a ------------------ #act_a
                                                   obj_a ---------------------------------------------------------------------------------------------- obj_a
               NP -------------------------------- NP
               np3
         Dt -- Dt
         dt1
the ---- the
         THE
         #Dt - #Dt
               N --- N
                     n1
worm --------------- worm
                     WORM
                     PREY
               #N -- #N
               #NP ------------------------------- #NP
                                                   #obj_a --------------------------------------------------------------------------------------------- #obj_a
                                                   actionA ----------------- actionA
                                                   agentA -------- agentA
                                                   objectA -------------------------------------------------------------------------------------------- objectA
                                                   #Sa ------------------------------- #Sa
                                                                                       Sa -------------------- Sa
                                                                                                               sa1
                                                                                                               agnt_b ----------------------- agnt_b
                                                                                                   NP -------- NP
                                                                                                   np1
                                                                                             PN -- PN --------------------------------------------------------------------- PN
                                                                                             pn2
it ------------------------------------------------------------------------------------------ it
                                                                                             IT
                                                                                             #PN - #PN -------------------------------------------------------------------- #PN
                                                                                                   #NP ------- #NP
                                                                                                               #agnt_b ---------------------- #agnt_b
                                                                                                               stat_b ------------------------------------------- stat_b
                                                                                                         V --- V
                                                                                                         v1
was ---------------------------------------------------------------------------------------------------- was
                                                                                                         WAS
                                                                                                         #V -- #V
                                                                                                               #stat_b ------------------------------------------ #stat_b
                                                                                                               adj_b ------------- adj_b
                                                                                                               ADJ ------ ADJ
                                                                                                                          adj1 -------------------------------------------- adj1
hungry ------------------------------------------------------------------------------------------------------------------ hungry
                                                                                                               #ADJ ----- #ADJ
                                                                                                               #adj_b ------------ #adj_b
                                                                                                               statusB ------------------------------------------ statusB
                                                                                                               agentB ----------------------- agentB
                                                                                                               qualityB ---------- qualityB
                                                                                       #Sa ------------------- #Sa
                                                                                       #S

0        1     2     3      4          5     6     7         8     9         10        11    12    13    14    15         16       17         18        19        20        21
```

Figure 8: The best SP-multiple-alignment produced by the SP Computer Model with *the fish ate the worm it was hungry* as a New SP-pattern, which appears in column 0, and a collection of Old SP-patterns as described in the text, some of which are shown in columns 1 to 21, one SP-pattern per column.

7 The potential of unsupervised learning

As noted in Section 3, each of the SP-multiple-alignments presented in this paper has been created by the SP Computer Model with with a collection of SP-patterns supplied by the user: one New SP-pattern and a repository of Old SP-patterns including those that appear in the given SP-multiple-alignment. This is because unsupervised learning in the SP Computer Model has shortcomings (Appendix A.3) which mean that it is not yet good enough for demonstrating how the SP System can learn the knowledge needed for disambiguating WSEs.

However, a comprehensive account of how WSEs may be disambiguated should include an account of how relevant knowledge may be learned. In the light of what has already been learned about the strengths and weaknesses of unsupervised learning in the SP Computer Model as it is now, and how the weaknesses may be overcome (Appendix A.3), here are some observations about how the system may be developed to learn the kind of knowledge needed for the disambiguation of WSEs:

- It is anticipated that it should be relatively straightforward to develop the SP System for the learning of segmental structures (words and phrases) and classes of such structures (e.g., nouns, verbs, and adjectives). As noted in Appendix A.3, the main hurdle to be overcome is the learning of structures at intermediate levels (between words and sentences) such as phrases and clauses.

- Likewise, it should be feasible to develop the system for the learning of discontinuous dependencies such as plurality dependencies or gender dependencies in syntax (more below). As with segmental structures, this is a weakness of the SP Computer Model as it is now (Appendix A.3).

- More challenging will be the learning of non-syntactic 'semantic' knowledge of the world, mainly because it will be necessary to generalise the system for the learning of two-dimensional and three-dimensional structures, and the learning of 'commonsense' knowledge of time, speed, distance, and the like.

- Even more challenging will be the learning of structures that combine syntactic and semantic knowledge such as, for example, the SP-pattern '`agnt_a NP #NP #agnt_a act_a V #V #act_a pat_a NP #NP #pat_a obj_a NP #NP #obj_a`' in Figure 1, part of the discussion in Appendix B.

In connection with the problem of finding the referent of a pronoun in a WSE, an SP-pattern like '`PEACE_LOVING PN #PN v1 n0`' in column 21 in Figure 1 is particularly important (Section 4.2), and likewise for the SP-pattern '`PEACE_LOVING PN #PN FEARED VIOLENCE`' in column 21 in Figure 2 (Section 4.3).

The importance of SP-patterns like these is that they provide connections between things like '`PEACE_LOVING`', '`PN #PN`', '`FEARED`', and '`VIOLENCE`', which may be and usually are separated from each other by other things. It is precisely the ability to learn discontinuous dependencies which is required to learn those kinds of SP-patterns.

As previously mentioned, it is unlikely that any other system, such as 'deep learning', could learn the necessary knowledge.

7.1 Discussion

The method that is proposed in this paper for finding the referent for a pronoun in a WSE relies in part on a rather simple principle: that any person or group of people that is 'peace loving' is also likely to 'fear violence', and likewise for the association between a person or group having an interest in 'radical change' and a willingness to 'advocate revolution', and so on. The concept of SP-multiple-alignment within the SP Computer Model provides a means for this kind of association to exert its influence across intervening structures without disruption or disturbance by those intervening structures.

The essential simplicity of this idea seems to be in conflict with Terry Winograd's suggestion that resolving the ambiguity in each of his example sentences requires a "sophisticated knowledge of councilmen, demonstrators, and politics" [21, p. 33]. Is this a genuine conflict or is there some way in which the two views may be reconciled? The suggestion here is that they can indeed be reconciled, something like this:

- It is likely that, in real life, most adults will indeed have a sophisticated knowledge of councilmen, demonstrators, and politics, and this knowledge will indeed be applied in the task of determining the referent of "they" in sentences like *The city councilmen refused the demonstrators a permit because they feared violence*. Their knowledge may, for example, contain many class-inclusion hierarchies like that shown in Figure 3 in Section 4.3.
- But in the same way that we use relatively short verbal labels for complex entities like 'New York' or 'Scotland', our non-verbal knowledge will contain many such labels in a 'chunking-with-codes' method for achieving compression of information ([35, Section 5], [22, Section 2.2.8]).
- As we have seen in each of the examples presented in Section 3, short identifiers are sufficient for determining the referent of the pronoun in each WSE. In more realistic examples, most such identifiers would each be a label or 'code' for a relatively complex chunk of information.
- In short, the method that has been proposed in this paper for determining the referent of a pronoun in a WSE will work with short identifiers, but in more realistic settings, it is likely that most such identifiers would be associated with relatively complex bodies of knowledge.

8 Conclusion

In 'Winograd Schema' kinds of sentence like *The city councilmen refused the demonstrators a permit because they feared violence* and *The city councilmen refused the demonstrators a permit because they advocated revolution* it is easy for people to decide what "they" refers to but it can be challenging for AI systems [21, p. 33].

The problem of interpreting examples like these is relevant to two topics in the study of artificial intelligence: how natural languages may be understood, and how to meet the challenges in 'commonsense reasoning' and the representation and processing of 'commonsense knowledge'. Examples like these are also relevant to AI because it has been suggested [6] that they may provide a way of testing AI systems for their human-like qualities which is superior to the 'Turing test': seeing whether, in a typewritten conversation between a person and a putative AI, the person would be deceived into thinking that they were conversing with another person.

This paper describes a unique and effective new method for interpreting 'Winograd Schema' (WS) forms of natural language. This new method is part of the *SP System*—meaning the *SP Theory of Intelligence* and its realisation in the *SP Computer Model*.

With the first sentence, above, as an example, the key to finding what "they" refers to is: 1) To record the association between 'peace loving' councilmen, the pronoun "they", and 'fear of violence', notwithstanding the fact that the elements of the association are scattered amongst other structures; 2) To take advantage of the SP System's strengths in finding patterns of association, including those whose constituents are widely scattered.

At present, information about patterns of association like that just described, and other relevant information about syntax and semantics, must be supplied to the SP System at the outset. But a comprehensive AI solution to these kinds of problem depends on an ability of the SP System, or any other kind of AI system, to learn relevant syntax and semantics for itself.

Existing strengths and weaknesses of the SP System in the unsupervised learning of English-like artificial languages suggest that it is not yet good enough to learn the kinds of syntactic and semantic knowledge that is needed to solve Winograd Schema types of problem, but there is clear potential for unsupervised learning in the SP System to be developed to a point where it can learn such knowledge for itself. In particular, there is clear potential for it to learn the kinds of association described under point (2) above—a kind of association which is similar to the kinds of discontinuous associations in syntax, such as number agreements and gender agreements—which often bridge intervening kinds of structure—which are found in most natural languages.

By contrast with any *ad hoc* method of disambiguating WS examples, the SP System is the product of an extended period of research seeking to simplify and integrate ideas across a broad canvass. For that reason it has strengths in diverse aspects of AI, in the representation of diverse forms of knowledge, and in the seamless integration of diverse aspects of AI and diverse forms of knowledge, in any combination (Appendix A.5). Its strengths in the interpretation of WS forms of natural language, and its potential in the unsupervised learning of the necessary knowledge, are bonuses that come with no added cost in terms of the complexity of the core model.

Appendices

A Outline of the SP system

The *SP System*—which means the *SP Theory of Intelligence* and its realisation in the *SP Computer Model*—is the product of an extended programme of research, seeking to simplify and integrate observations and concepts across artificial intelligence, mainstream computing, mathematics, and human learning, perception, and cognition, with information compression as a unifying theme. Despite its ambition, this objective has been largely met, as summarised in Appendix A.5, below.

The SP System is described most fully in [22] and more briefly in [24]. Other information about the SP System may be found on www.cognitionresearch.org/sp.htm.

The SP System is conceived as a brain-like system that receives *New* information through its senses and stores some or all of it as *Old* information, as shown schematically in Figure 9.

In the SP System, all kinds of information or knowledge are represented in arrays of atomic *SP-symbols* in one or two dimensions called *SP-patterns*. At present, the SP Computer Model works only with one-dimensional SP-patterns but it is envisaged that it will be generalised to work with two-dimensional SP-patterns. And at present, an SP-symbol is an ASCII character, or a string of such characters bounded by spaces or the end of the SP-pattern in which it appears.

A.1 Information compression, Ockham's razor, 'simplicity', and 'power'

Information compression, mentioned above, may be seen to be equivalent to a process of maximising the *Simplicity* of any given body of information, **I**, by extraction of *redundancy* from **I**, whilst retaining as much as possible of its descriptive *Power*. Hence the name 'SP'.

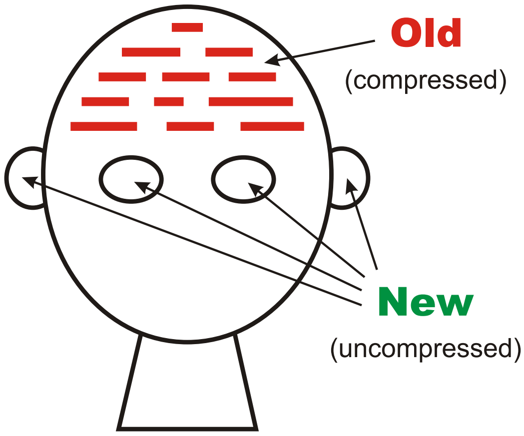

Figure 9: Schematic representation of the SP System from an 'input' perspective.

Given that equivalence, information compression may be seen to have a two-fold significance in the SP programme of research:

- Simplification and integration of observations and concepts across a broad canvass (the overarching goal of the SP research) is, in accordance with Ockham's razor, a process of developing a system that combines conceptual *Simplicity* of the system with high levels of explanatory or descriptive *Power*.

- Information compression is central in the workings of the SP System. In that connection, we can make a broad distinction between two kinds of processes in the SP System:

  - *The building of SP-multiple-alignments.* The building of *SP-multiple-alignments* in the SP System, described in Section A.2, is a powerful means of compressing New information from the SP System's environment. It is also the key to the SP System's strengths in most aspects of intelligence, but with additional processing in unsupervised learning (next).

  - *Unsupervised learning.* The process of unsupervised learning in the SP System (Appendix A.3) is another means of compressing information which incorporates the building of SP-multiple-alignments but which also includes a process of building one or more *SP-grammars*, meaning collections of Old SP-patterns which are relatively effective in the compression of a given set of New SP-patterns.

The reason that information compression has been adopted as a unifying principle in the workings of the SP System is because of abundant evidence for the importance of information compression in human learning, perception, and cognition [35]. The success of the system in modelling several aspects of human intelligence (Appendix A.5.1) provides corroborating evidence for the importance of information compression in human cognition.

A last point to mention in this connection is that it has been known for some time that there is an intimate relation between information compression and concepts of inference and probability [15, 18, 19, **?**]. This makes it relatively straightforward for the SP System to calculate probabilities for inferences made by the system ([24, Section 4.4], [22, Section 3.7]).

The emphasis on IC in the SP System accords with research in the tradition of Minimum Length Encoding (see, for example, [9]), with the qualification that most research relating to MLE assumes that the concept of a universal Turing machine provides the foundation for theorising, whereas the SP System is founded on concepts of the matching and unification of patterns and SP-multiple-alignment (Section 2.4.2).

A.2 SP-multiple-alignments

The concept of *SP-multiple-alignment* in the SP System, which has been borrowed and adapted from the concept of 'multiple sequence alignment' in bioinformatics, means the arrangement of two or more SP-patterns alongside each other, and the 'stretching' of SP-patterns in a computer so that matching symbols are brought into line.

An example of an SP-multiple-alignment, created by the SP Computer Model, is shown in Figure 10. This is the best SP-multiple-alignment produced by the SP computer model with: a New SP-pattern, '`f o r t u n e f a v o u r s t h e b r a v e`' shown in column 0, representing a sentence to be analysed or 'parsed'; and a repository of user-supplied Old SP-patterns representing grammatical categories, including morphemes and words. Some of those Old SP-patterns appear in columns 1 to 15 of the figure, one SP-pattern per column. The overall effect of this SP-multiple-alignment is to parse the sentence into its grammatical parts and subparts, including words and morphemes.

```
0   1    2      3      4     5    6      7      8    9     10    11    12   13     14   15

                       S
                       Num ------------------------------------- Num
                                                                 PL
                       ; --------------------------------------- ;
                       NP -------------------------------- NP
                                                           np0
                                  N ---------------------- N
                                  Np --------------------------- Np
                             R -- R
                             R0
w -------------------------- w
i -------------------------- i
n -------------------------- n
d -------------------------- d
                             #R - #R
                                  SUF -- SUF
s -------------------------------------- s
                                  #SUF - #SUF
                                  #N --------------------- #N
                                                           : --- :
                                                     Q --- Q
                                                P -- P
                                                p1
f --------------------------------------------- f
r --------------------------------------------- r
o --------------------------------------------- o
m --------------------------------------------- m
                                                #P - #P
                                                     NP ------------------------------- NP
                                                                                        np1
                                                                                   D -- D
                                                                                   d0
t -------------------------------------------------------------------------------- t
h -------------------------------------------------------------------------------- h
e -------------------------------------------------------------------------------- e
                                                                                   #D - #D
                                                                            N --------- N
                                                                            Np
                                                                       R -- R
                                                                       R1
w -------------------------------------------------------------------- w
e -------------------------------------------------------------------- e
s -------------------------------------------------------------------- s
t -------------------------------------------------------------------- t
                                                                       #R - #R
                                                                            SUF
                                                                            #SUF
                                                                            #N -------- #N
                                                     #NP ------------------------------ #NP
                                                     #Q -- #Q
                       #NP ------------------------------- #NP
         VP ---------- VP
    V -- V
    Vp --------------------------------------------------------- Vp
a - a
r - r
e - e
    #V - #V
         ADJ -- ADJ
                aj1
s ------------- s
t ------------- t
r ------------- r
o ------------- o
n ------------- n
g ------------- g
         #ADJ - #ADJ
         #VP --------- #VP
                       #S

0   1    2      3      4     5    6      7      8    9     10    11    12   13     14   15
```

Figure 10: The best SP-multiple-alignment produced by the SP Computer Model with a New SP-pattern, ‘`w i n d s f r o m t h e w e s t a r e s t r o n g`’ (in column 0), representing a sentence to be parsed and a repository of user-supplied Old SP-patterns representing grammatical categories, including morphemes and words. Some of those Old SP-patterns appear in columns 1 to 15 of the SP-multiple-alignment, one SP-pattern per column.

Here, the meaning of 'best' is that the given SP-multiple-alignment is the one which allows the New SP-pattern to be encoded most economically in terms of the Old SP-patterns in the SP-multiple-alignment, as described in [24, Section 4.1] and [22, Section 3.5].

For any given New pattern and collection of Old SP-patterns, the number of possible SP-multiple-alignments is normally too large to be searched exhaustively. Hence, the SP Computer Model uses heuristic techniques which can normally find SP-multiple-alignments that are reasonably good, and may quite often be theoretically ideal, but which cannot guarantee to find such ideal SP-multiple-alignments.

A point of interest about the SP-multiple-alignment in Figure 10, which is referred to in the main body of the paper, is that the SP-pattern in column 11 marks the sentence in column 0 as a plural sentence. This is done with the SP-symbol '`PL`' near the beginning of the SP-pattern and with the pair of SP-symbols '`Np`' and '`Vp`' later in the SP-pattern which mark the grammatical dependency between a plural noun-phrase ('`Np`') followed by a plural verb-phrase ('`Vp`'). This kind of grammatical dependency is often described as 'discontinuous' because it can jump over any kind of intervening structure such as "which may bring snow" in a sentence like "Winds from the West, which may bring snow, are strong.", without disturbing or being disturbed by that intervening structure.

A.3 Unsupervised learning

As mentioned in Appendix A.1 above, unsupervised learning in the SP System incorporates the building of SP-multiple-alignments but, in addition, there is a process of creating one or more *SP-grammars*, meaning collections of Old SP-patterns which are relatively good for the economical encoding of a given set of New SP-patterns.

The reason that, with respect to learning, the SP research is concentrating on the development of *unsupervised* learning—meaning learning without assistance from a 'teacher' or anything equivalent—is because it appears that most human learning is unsupervised, and because of the belief that unsupervised learning can provide the foundation of other kinds of learning such as 'supervised learning', 'reinforcement learning', 'learning by being told', 'learning by imitation', and so on.

The main framework for unsupervised learning has been established in the SP Computer Model. It has already demonstrated an ability to discover generative grammars from unsegmented samples of English-like artificial languages, including segmental structures, classes of structure, and abstract patterns. But it has shortcomings summarised in [24, Section 3.3]: failing to learn intermediate levels and discontinuous dependencies in linguistic structures.

How the system may be developed is outlined here:

- *The learning of segmental structures.* The SP Computer Model has already demonstrated an ability to learn segmental structures such as words from data in which significant segments are not marked (Appendix A.3). Thus it has clear potential to learn such things as words and more abstract structures such as sentences. But a weakness of the model, mentioned above, is that it does not learn structures at intermediate levels such as phrases and clauses (*ibid.*). It is anticipated that this problem will prove to be soluble.

- *The learning of class-inclusion structures.* Much the same may be said about the learning of class-inclusion structures such as nouns, verbs, adjectives, and so on. The SP Computer Model can learn such groupings at an abstract level but cannot learn structures at intermediate levels. As before, it is anticipated that this problem will prove to be soluble.

- *The learning of discontinuous dependencies.* Another shortcoming in how the SP Computer Model learns, mentioned above, is that it cannot yet learn discontinuous dependencies in syntax such as gender dependencies (masculine or feminine) in a language like French (Appendix A.3). As before, it is anticipated that this problem will prove soluble.

- *The learning of semantic structures.* An overarching goal in the SP programme of research is that structures and processes in the SP System should be applicable across a broad canvass including artificial intelligence, mainstream computing, mathematics, and human learning, perception, and cognition. In accordance with that principle, it is anticipated that, if the kinds of learning described in the previous three bullet points can be developed for the learning of syntax, they will prove to be applicable to the learning of 'semantic' structures, meaning knowledge of the non-syntactic world. The main change that will be needed will be generalisation of the SP System for SP-patterns in two dimensions, which will facilitate the learning of structures in three dimensions [25, Sections 6.1 and 6.2]. And there will need to progress in the learning of 'commonsense' knowledge of the kinds described by [2].

- *The integration of syntax and semantics.* Probably the most challenging problem to be solved is to develop the SP System to a stage where it can learn plausible structures that integrate both natural language syntax and the non-syntactic knowledge that provides the semantics of natural language.

A.4 SP-Neural

SP-Neural is a version of the SP System expressed in terms of neurons and their inter-connections [29]. This is quite different from the popular 'deep learning in artificial neural networks' [13] and is likely to share the several advantages of the SP System compared with deep learning ([30, Section V], [34]). SP-Neural has potential as a source of hypotheses in neuroscience and as a guide to explorations in that field.

A.5 Strengths and potential of the SP System

In accordance with Ockham's razor (Appendix A.1), the SP System combines conceptual *Simplicity* with high levels of descriptive or explanatory *Power*, the latter summarised in the subsections that follow. Further information may be found in [24, Sections 5 to 12], [22, Chapters 5 to 9], [30], and in other sources referenced in the subsections that follow.

A.5.1 Versatility in aspects of intelligence

In the modelling of human-like intelligence, strengths of the SP System includes: unsupervised learning, the analysis and production of natural language; pattern recognition that is robust in the face of errors in data; pattern recognition at multiple levels of abstraction; computer vision [25]; best-match and semantic kinds of information retrieval; several kinds of reasoning (next paragraph); planning; and problem solving.

With regard to reasoning, the strengths of the SP system include: one-step 'deductive' reasoning; chains of reasoning; abductive reasoning; reasoning with probabilistic networks and trees; reasoning with 'rules'; nonmonotonic reasoning and reasoning with default values; Bayesian reasoning with 'explaining away'; causal reasoning; reasoning that is not supported by evidence; the inheritance of attributes in class hierarchies and part-whole hierarchies. Where it is appropriate, probabilities for inferences may be calculated in a straightforward manner ([22, Section 3.7], [24, Section 4.4]).

There is also potential in the system for spatial reasoning [26, Section IV-F.1], for what-if reasoning [26, Section IV-F.2], and for commonsense reasoning (this paper, and [28]).

A.5.2 Versatility in the representation of knowledge

Although SP-patterns are not very expressive in themselves, they come to life in the SP-multiple-alignment framework. Within that framework, they may serve in the representation of several different kinds of knowledge, including: the syntax of natural languages; class-inclusion hierarchies (with or without cross classification); part-whole hierarchies; discrimination networks and trees; if-then rules; entity-relationship structures [23, Sections 3 and 4]; relational tuples (*ibid.*, Section 3), and concepts in mathematics, logic, and computing, such as 'function', 'variable', 'value', 'set', and 'type definition' ([22, Chapter 10], [27, Section 6.6.1], [31, Section 2]).

The addition of two-dimensional SP patterns to the SP computer model is likely to expand the representational repertoire of the SP system to structures in two-dimensions and three-dimensions, and the representation of procedural knowledge with parallel processing.

A.5.3 Seamless integration of diverse aspects of intelligence, and diverse kinds of knowledge, in any combination

An important third feature of the SP system, alongside its versatility in aspects of intelligence and its versatility in the representation of diverse kinds of knowledge, is that *there is clear potential for the SP system to provide seamless integration of diverse aspects of intelligence and diverse kinds of knowledge, in any combination.* This is because diverse aspects of intelligence and diverse kinds of knowledge all flow from a single coherent and relatively simple source: the SP-multiple-alignment framework.

It appears that seamless integration of diverse aspects of intelligence and diverse kinds of knowledge, in any combination, is *essential* in any artificial system that aspires to the fluidity, versatility and adaptability of the human mind.

Figure 11 shows schematically how the SP system, with SP-multiple-alignment centre stage, exhibits versatility and integration.

A.5.4 Potential benefits and applications of the SP system

Apart from its strengths and potential in modelling aspects of human intelligence, it appears that the SP System has several potential benefits and applications [32, Section 7]. These include: how the SP System may help solve nine problems with big data; the SP system opens up a radically new approach to the development of intelligence in autonomous robots; the SP System may form the basis for a database system with intelligence; the SP System may assist with medical diagnosis; the SP System may help in the development of computer vision and in the understanding of natural vision; SP-Neural may suggest hypotheses and avenues for research in neuroscience; the SP System has potential in the development of commonsense reasoning; other areas of application including [27]: the simplification and integration of computing systems; applications of natural language processing; best-match and semantic forms of information retrieval; software engineering [31]; the representation of knowledge, reasoning, and the semantic web; information compression; bioinformatics; the detection of computer viruses; and data fusion; the concept of information compression via the matching and unification of patterns provides an entirely novel interpretation of mathematics [33] which is quite unlike anything described in existing writings about the philosophy of mathematics or its application in science and has several potential benefits and applications.

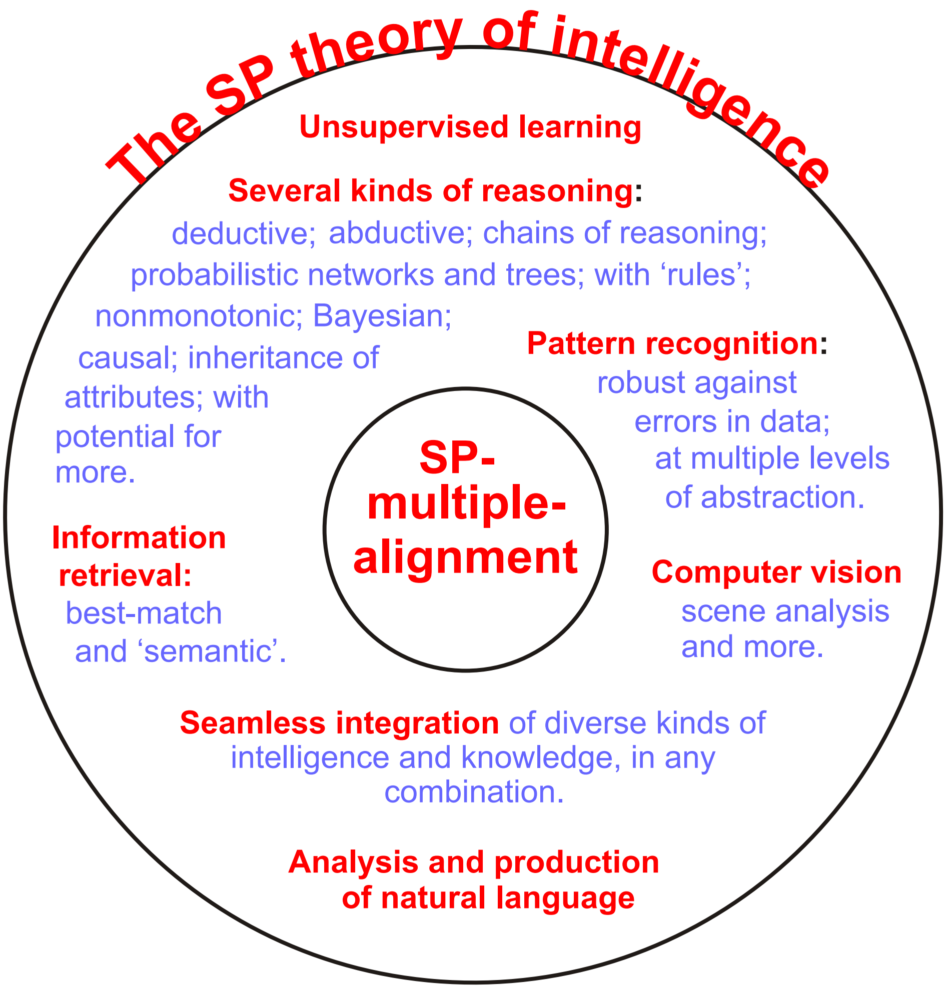

Figure 11: A schematic representation of versatility and integration in the SP system, with SP-multiple-alignment centre stage.

B Making connections between syntax and semantics

As mentioned in Section 4.1, the subject of this Appendix—the representation and processing of syntax and semantics in the example from that section and all other examples in this paper—is quite tentative and not relevant to the main substance of the paper: how WSEs may be disambiguated. Readers may safely ignore this Appendix if they wish.

By contrast with the SP-multiple-alignment in Figure 10, the SP-patterns in Figure 1 attempt to show in a simplified way, how, with the SP System, syntactic structures may be associated with corresponding semantic structures. For example:

- Each word is represented by an SP-pattern like this: ‘`N n3 councilmen COUNCILMEN PEACE_LOVING #N`’ in column 14, where the lower-case string of characters ‘`councilmen`’ is intended to represent the surface form of the word, while the associated string of characters in capital letters, '`COUNCILMEN`', is intended to represent, in a highly simplified form, the meaning of the word.

  The SP-symbol '`PEACE_LOVING`', which is discussed in Section 4.2, may be seen to be a supplement to, or part of, the semantic information associated with '`councilmen`'.

  The other symbols, '`N`', '`n3`', '`#N`' in '`N n3 councilmen COUNCILMEN peace_loving #N`', are grammatical categories, much as in the SP-patterns in Figure 10.

- The SP-pattern '`Sa agnt_a NP #NP #agnt_a act_a V #V #act_a pat_a NP #NP #pat_a obj_a NP #NP #obj_a actionA agentA patientA objectA #Sa`' in column 7 is intended to represent both the syntax and semantics of '`the city councilmen refused the demonstrators a permit`'.

  Here, the first part of the pattern, '`agnt_a NP #NP #agnt_a act_a V #V #act_a pat_a NP #NP #pat_a obj_a NP #NP #obj_a`' is largely concerned with syntax, while '`actionA agentA patientA objectA`' is largely about semantics. Notice that the order of '`agnt_a ... #agnt_a act_a ... #act_a pat_a ... #pat_a obj_a ... #obj_a`' (meaning 'agent, action, patient, object') in the 'syntax' part is different from '`actionA agentA patientA objectA`' ('action, agent, patient, object') in the 'semantics' part.

- In a similar way, the SP-pattern '`Sb agnt_b NP #NP #agnt_b act_b V #V #act_b obj_b NP #NP #obj_b actionB agentB objectB #Sb`' in column 11 is intended to represent both the syntax and semantics of '`they feared violence`'. Here, the first part of the pattern, '`Sb agnt_b NP #NP #agnt_b act_b V #V #act_b obj_b NP #NP #obj_b`' ('agent', 'action', 'object') is largely concerned with syntax, while '`actionB agentB objectB`' ('action', 'agent', 'object') is largely about semantics, and the ordering is different in the two cases.

In English and many other languages, a given idea may often be expressed in more than one way. For example, the meaning of *John hit the ball* is essentially the same as *The ball was hit by John*. This suggests that there is a single semantic structure that may be expressed via two or more surface structures and that, for at least one of those surface structures, perhaps all of them, there is a process of mapping the surface structure on to the underlying semantic structure.

With the SP-multiple-alignment shown in Figure 1, that kind of mapping may be achieved like this:

- With the SP-pattern in column 27, '`agnt_a #agnt_a agentA`':
  - The pair of SP-symbols '`agnt_a #agnt_a`' in column 27 are aligned with matching symbols in column 7 that bridge the pair of symbols '`NP #NP`' in that column, thus identifying the noun phrase 'the city councilmen'.
  - The SP-symbol '`agentA`' in column 27 is aligned with the matching symbol in column 7.
  - The overall effect is to make a connection between 'the city councilmen' and '`agentA`'.
- There is a similar mapping of symbols: in '`act_a #act_a actionA`' in column 28 to symbols in column 7; in '`pat_a #pat_a patientA`' in column 23 to symbols in column 7; and in '`obj_a #obj_a objectA`' in column 22 to symbols in column 7.
- The overall effect is to map the surface form '`the city councilmen refused the demonstrators a permit`' ('actor', 'action', 'patient', 'object') into the form ('actionA', 'agentA', 'patientA', 'objectA') in column 7, which may be regarded as a 'semantic' version of the surface form.

- In a similar way, the surface form '`they feared violence`' may be seen to be mapped into the semantic form ('actionB', 'agentB', 'objectB') in column 11 via the SP-patterns '`agnt_b #agnt_b agentB`' (column 25), '`act_b #act_b actionB`' (column 26), and '`obj_b #obj_b objectB`' (column 24).

References

[1] D. Bailey, A. Harrison, Y. Lierler, V. Lifschitz, and J. Michael. The Winograd Schema Callenge and reasoning about correlation. In *Logical Formalizations of Commonsense Reasoning: Papers from the 2015 AAAI Spring Symposium*, pages 17–24, 2015.

[2] E. Davis and G. Marcus. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM*, 58(9):92–103, 2015.

[3] A. Emami, A. Trischler, K. Suleman, and J. C. K. Cheung. A generalized knowledge hunting framework for the Winograd Schema Callenge. In *Proceedings of NAACL-HLT 2018: Student Research Workshop, New Orleans, Louisiana, June 2 - 4, 2018*, pages 25–31, 2018.

[4] B. Goertzel. Cogprime: an integrative architecture for embodied artificial general intelligence. Technical report, The Open Cognition Project, 2012. PDF: bit.ly/1Zn0qfF, 2012-10-02.

[5] J. E. Laird. *The Soar Cognitive Architecture*. The MIT Press, Cambridge, Mass., 2012. ISBN-13: 978-0-262-12296-2.

[6] H. J. Levesque. The Winograd Schema Callenge. In *Proceedings of the Tenth International Symposium on Logical Formalizations of Commonsense Reasoning (Commonsense-2011)*, 2011. Part of the AAAI Spring Symposium Series at Stanford University, March 21-23, 2011.

[7] H. J. Levesque. On our best behaviour. *Artificial Intelligence*, 212:27–35, 2014.

[8] H. J. Levesque, E. Davis, and L. Morgenstern. The Winograd Schema Callenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, pages 552–561, 2012.

[9] M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, New York, 4th edition, 2019.

[10] L. Morgenstern and C. L. Ortiz Jr. The Winograd Schema Callenge: evaluating progress in commonsense reasoning. In *Proceedings of the Twenty-Seventh Conference on Innovative Applications of Artificial Intelligence, 2015*, pages 4024–4025, 2015.

[11] A. Rahman and V. Ng. Resolving complex cases of definite pronouns: the Winograd Schema Challenge. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 777–789. Association for Computational Linguistics, 2012.

[12] A. Richard-Bollans, L. G. Álvarez, and A. G. Cohn. The role of pragmatics in solving the Winograd Schema Callenge. In A. S. Gordon, R. Miller, and G. Turan, editors, *Proceedings of the Thirteenth International Symposium on Commonsense Reasoning (Commonsense 2017), 06-08 Nov 2017, London, UK*, 2018.

[13] J. Schmidhuber. Deep learning in neural networks: an overview. *Neural Networks*, 61:85–117, 2015.

[14] P. Schüller. Tackling Winograd Schemas by formalizing relevance theory in knowledge graphs. In *Proceedings of the Fourteenth International Conference on Principles of Knowledge Representation and Reasoning*, pages 358–367, 2014.

[15] C. E. Shannon and W. Weaver. *The Mathematical Theory of Communication.* University of Illinois Press, Urbana, 1949.

[16] A. Sharma, N. H. Vo, S. Aditya, and C. Baral. Towards addressing the Winograd Schema Callenge—building and using a semantic parser and a knowledge hunting module. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 1319–1325, 2015.

[17] A. Sharma, N. H. Vo, S. Gaur, and C. Baral. An approach to solve Winograd Schema Callenge using automatically extracted commonsense knowledge. In *Logical Formalizations of Commonsense Reasoning: Papers from the 2015 AAAI Spring Symposium*, pages 141–144, 2015.

[18] R. J. Solomonoff. A formal theory of inductive inference. Parts I and II. *Information and Control*, 7:1–22 and 224–254, 1964.

[19] R. J. Solomonoff. The discovery of algorithmic probability. *Journal of Computer and System Sciences*, 55(1):73–88, 1997.

[20] A. M. Turing. Computing machinery and intelligence. *Mind*, 59:433–460, 1950.

[21] T. Winograd. Understanding natural language. *Cognitive Psychology*, 3(1):1–191, 1972.

[22] J. G. Wolff. *Unifying Computing and Cognition: the SP Theory and Its Applications.* CognitionResearch.org, Menai Bridge, 2006. ISBNs: 0-9550726-0-3 (ebook edition), 0-9550726-1-1 (print edition). Distributors, including Amazon.com, are detailed on bit.ly/WmB1rs.

[23] J. G. Wolff. Towards an intelligent database system founded on the SP theory of computing and cognition. *Data & Knowledge Engineering*, 60:596–624, 2007. arXiv:cs/0311031 [cs.DB], bit.ly/1CUldR6.

[24] J. G. Wolff. The SP Theory of Intelligence: an overview. *Information*, 4(3):283–341, 2013. arXiv:1306.3888 [cs.AI], bit.ly/1NOMJ6l.

[25] J. G. Wolff. Application of the SP Theory of Intelligence to the understanding of natural vision and the development of computer vision. *SpringerPlus*, 3(1):552–570, 2014. arXiv:1303.2071 [cs.CV], bit.ly/2oIpZB6.

[26] J. G. Wolff. Autonomous robots and the SP Theory of Intelligence. *IEEE Access*, 2:1629–1651, 2014. arXiv:1409.8027 [cs.AI], bit.ly/18DxU5K.

[27] J. G. Wolff. The SP Theory of Intelligence: benefits and applications. *Information*, 5(1):1–27, 2014. arXiv:1307.0845 [cs.AI], bit.ly/1FRYwew.

[28] J. G. Wolff. Commonsense reasoning, commonsense knowledge, and the SP Theory of Intelligence. Technical report, CognitionResearch.org, 2016. Submitted for publication, arXiv:1609.07772 [cs.AI], HAL: hal-01970147 v2, bit.ly/2eBoE9E.

[29] J. G. Wolff. Information compression, multiple alignment, and the representation and processing of knowledge in the brain. *Frontiers in Psychology*, 7:1584, 2016. arXiv:1604.05535 [cs.AI], bit.ly/2esmYyt.

[30] J. G. Wolff. The SP Theory of Intelligence: its distinctive features and advantages. *IEEE Access*, 4:216–246, 2016. arXiv:1508.04087 [cs.AI], bit.ly/2qgq5QF.

[31] J. G. Wolff. Software engineering and the SP Theory of Intelligence. Technical report, CognitionResearch.org, 2017. Submitted for publication. arXiv:1708.06665 [cs.SE], bit.ly/2w99Wzq.

[32] J. G. Wolff. Introduction to the SP Theory of Intelligence. Technical report, CognitionResearch.org, 2018. arXiv:1802.09924, bit.ly/2ELq0Jq.

[33] J. G. Wolff. Mathematics as information compression via the matching and unification of patterns. Technical report, CognitionResearch.org, 2018. Submitted for publication. arXiv:1808.07004 [cs.AI], bit.ly/2LWbjtK.

[34] J. G. Wolff. Solutions to problems with deep learning. Technical report, CognitionResearch.org, 2018. arXiv:1801.05457 [cs.LG], bit.ly/2AJzu4j.

[35] J. G. Wolff. Information compression as a unifying principle in human learning, perception, and cognition. *Complexity*, 2019:38 pages, February 2019. Article ID 1879746, viXra:1707.0161v3, hal-01624595 v2.