PADS Restoration and Its Importance in Reading Comprehension and Meaning Representation

Shian-jung Chen
National Taiwan University of Science and Technology
shianjungchen@yahoo.com

Abstract

Unlike competent human readers capable of inferring, tracing, and filling out gaps or hurdles left behind by authors' use of transformations in their writing such as permutation, addition, deletion, and substitution (PADS), these operations are challenging to computer readers and new foreign language learners. This paper reports a parser's use of a suite of NLP technologies - clause boundary detection, resolution of different anaphors, inter-event relation finding, and case frame building - to fill out PADS gaps and output a much more explicit kernel-like meaning representation that includes case relation tuples of "Who Did What to Whom" and the inter-event relations based on conjoining, embedding, branching, insertion and apposition. According to Halliday and Hasan (1976), those gaps serve as cohesive devices to achieve better texture of the text organization. The transformations are ruled-based and they are important part of native speakers' competence. Though the rule-based parser is still short of perfection, the necessary design is in place and it has quite a few encouraging results. This report will also show the usefulness of PADS restoration technology in CALL and information extraction.

Key words: English parser, PADS gaps, PADS restoration, case frame building, clause boundary detection, zero anaphor resolution, anaphor resolution, event relation finding, pronoun co-reference resolution, PP attachment, garden-path, information extraction, computer reading, meaning representation, CALL

1 Introduction

According to Lyons (1977), two different conceptions of kernel-sentences have been formalized in transformational grammar: one by Harris and the other by Chomsky. Zellig Harris defined a kernel as one that is not derived from any other sentence by means of a transformation rule; while Chomsky (1957) regarded a kernel as one generated in the grammar without the operation of "optional" transformations. Without looking into how kernels are conceptualized differently, kernels refer to "simple, complete, active, affirmative declaratives (or statements)", from which surface structures are derived. When Chomsky postulated theory of transformational grammar, he has PADS (permutation, addition, deletion and substitution) in mind as the stumbling rules that alienate Deep Structure from Surface Structure. For example, active sentences are transformed into passive either because the Agent is unknown or so that the Agent is moved to the end of a clause to serve as a link to the following clause. This need to link in texture organization might cause a careless reader to misread since the Agent and the Patient are swapped. Misreading is even more likely if the passive is in a participial, in which the verb-to-be is deleted. Ambiguity or misreading caused by PADS is sometimes referred to by reading researchers as the garden-path phenomenon.

Whether they are called PADS gaps or garden-path, the derivations are causes of misinterpretation for computer reading and for underachieved readers. Nevertheless, for Halliday and Hasan (1976), they are great devices for cohesion, which refers to "the relations of meaning that exist within the text". Halliday and Hasan classify cohesive devices into five categories: reference, ellipsis, substitution, lexical cohesion, and conjunction. The mechanism "reference" relates one element of the text to another for its interpretation because they express

27th Pacific Asia Conference on Language, Information, and Computation pages 563－570

the same referent. "Ellipsis" is used to omit an item to avoid repetition. "Substitution" refers to the use of pronouns or pro-forms to avoid using the same phrase for the same referent mentioned earlier. "Lexical cohesion" refers to two elements that share a lexical field or collocation. "Conjunction" refers to particular expressions used to create parallel connections.

It's interesting to note that two linguistic schools established two decades away from each other should use similar mechanisms to refer to two very different concepts, one for generating surface sentences and the other in achieving text meaning. Chomskyan Generative Grammar and Hallidayan cohesion concept are mentioned here to draw attention to two things: 1) they point out that transformation rules and cohesive devices both involve missing, displaced or surrogate words or phrases that are extremely difficult for sequential or distance-based computation or for L2 learners; 2) the answer to their restoration should be in the study of language knowledge.

In the following sections, the author will first point out that the occurrences of PADS gaps are almost entirely predictable. In other words, we know where they are from and how they are used. With this, how the English parser achieves different goals of PADS restoration, namely, clause boundary detection, PP attachment, zero anaphor resolution, anaphor resolution, pronoun co-reference resolution, event relation finding, will be reported. I will then show that the problems addressed are also causes of garden-path phenomena. The next section illustrates an explicit kernel-like meaning representation that is used to integrate PADS restoration and highlight explicit referents as well as intra-event and inter-event relations. Then, some preliminary results and evaluation methods will be reported. At the end, the paper will show how the parsing outputs in XML form can be used to help with CALL (computer-aided language learning), information extraction and knowledge discovery.

2 Kernels and derived sentences

Kernels are simple, complete, active, affirmative statements. From them compound, complex, incomplete, passive, negative statements, or questions and commands are derived. Although not all derivations have all PADS gaps and not each PADS gap occurs solely to a single derivation, the co-occurrence of a derivation with a PADS transformation is basically predictable.

Kernel	Derivation	PADS	Issues
simple	compound complex	deletion substitution deletion permutation deletion	zero-anaphor relative-anaphor trace
complete	incomplete	deletion	zero-anaphor
active	passive	addition permutation deletion	discontinuity trace anaphor
affirmative	negative	addition permutation	discontinuity trace
statement	question command	addition permutation deletion	discontinuity trace zero-anaphor
reference	pronoun	substitution	anaphor co-reference

Table 1

Table 1 shows the correspondence between PADS operations and constituent types in English. It also shows that the phrase structure type in English dictates the occurrence of PADS or language mechanisms. A relative clause either has a relative pronoun or it can be omitted. The existence of a relative pronoun is the result of substitution. And it's likely that the Object inside the relative is moved (permuted) to the left of the clause. On the other hand, the omission of the relative clause implies an extra operation of deletion, so zero anaphor resolution, rather than relative-anaphor resolution is needed to restore PADS gaps. For passive reduced relative or past participial, deletion and zero-anaphor resolution will be involved because both the Subject and verb-to-be are missing. Permutation also occurs in this situation. Subject- or object-control infinitival also involves zero anaphor and it is the control type that decides which referent to be restored in zero anaphor resolution. Compounds or other kinds of conjoining often involve zero-anaphor, meaning that Subject or Verb or Object might be omitted if repetition is sensed.

3 English parser and NLP resolutions

The deep parser built by the author (Wasson et al. 2010, Chen & Lu 2012) is a spin-off of the parser family based on the generalized transition network grammar (GTN) parsers of Loritz's (1992) which in turn were built on the framework of an augmented transition network (ATN). Some new designs are implemented to

enable the parser to do integrated meaning representation and PADS restoration. To attain these two goals, the parser needs near perfect constituency (deciding the beginning and end of a constituent and its structure type) and the finding of intra-event relations of "Who Did What to Whom" as well as inter-event relations of conjoining, embedding, branching, insertion and apposition.

3.1 Clause boundary detection

Unlike most clause boundary detection tasks reported, the parser here uses case frame as the ultimate judge of clause boundaries because not every clause has a salient boundary marker and most clause markers are ambiguous themselves. This implementation is driven by the idea that if a clause has got enough case roles required by a predicate, a new clause will be expected.

Clause boundary detection is important in this study because the finding of both intra-event and inter-event relations depends on it. So far the parser returns an accuracy rate of over 90%. Its evaluation is simple and clear. (see Table 4).

3.2 Different anaphor resolutions

There are three kinds of anaphor resolution implemented in this parser -- relative anaphor for relative clauses, co-reference resolution for personal pronouns, and zero-anaphor resolution for conjoining construction, pronoun-less relatives, reduced relatives, etc. As mentioned in section 2, most anaphor resolutions are not so hard to implement because their restoration clues are predictable. The difficult parts of anaphor resolution lie in pronoun co-reference resolution and the zero-anaphor resolution that is related to scope of coordination.

The concept behind pronoun resolution is easy if adequate mention-lists are built and the priority of different mention-lists is set. The difficulty lies in the fact that it takes time to subcategorize all nouns as person or non-person and to add features of male or female. Scope of coordination is easier for omitted Subject or Verb, but very difficult for omitted Object in parallel construction. It is further complicated by morphological conversion, meaning so many English nouns also function as verbs.

The most important thing for anaphor resolution is the use of the "carry-over" of some register of Subject or Object right at the moment when the old clause ends and a new clause is introduced. At that moment, the co-referent of an anaphor is used to restore the empty element or take the place of *which* or *he*.

3.3 PP attachment

The success or failure of PP attachment is critical to clause boundary detection and constituency in general. Its difficulty mainly lies in the context of a preposition following the grammatical NP object of a verb. The parser makes use of event classification of the object NP as well as the information of two-word verbs to determine whether a PP is attached to an NP or a Verb. So far the only thing that is still troubling the parser's PP attachment is in the case where the PP of interest is itself a parallel construction.

3.4 Case frame building

For the intra-event relation or case relation, according to case grammar (Fillmore 1968), the parser's representation of "Who Did What to Whom" is laid out under the label of Agent, Predicate, MainVerb, Patient and Goal. Among them, Agent refers to Doer or the only participant of the event. By default, it should be the grammatical Subject of the clause unless a passive voice is detected, which in turn moves the Subject to the Patient position. Most PP participants (an NP marked with a case marking preposition) are placed under the label Goal, except when a two-word verb is identified. Goal position is also saved for marginal participants if no other case role is found. This is aimed at accommodating as many participants as possible. It should be noted that embedded noun clauses or non-finite clauses are also included in the case frame representation.

3.5 Inter-event relation finding

Aside from using different kinds of anaphor resolution to upgrade the parser to do more than sentence parsing, the parser is further developed to find inter-event relations. In English, there are actually three relations between events or referents: that of equivalence, embedment and dependency. However, following Chen (2010), "insertion" joins "branching" for the dependency relation and "apposition" is added to share with "conjoining" the equivalence relation. At the junction of clause boundary, a principle of sentence construction is selected among conjoining, embedding, branching, insertion and apposition to describe how the current event is related to another one. It should be noted that

only five principles are needed to capture all inter-event relations in English (Chen 2010).

3.6 Garden-path phenomena

From the parser design and implementation of all kinds of resolution, the author notices that relational function words are the most ambiguous in English. Comma(,) tops the list of ambiguous words. As a boundary marker, it can lead to a new clause, a new phrase, a series of parallel constituents, an insertion or an apposition. Any decision made at the junction of a comma might guarantee a successful parse or ruin the whole thing. Comma junction is a key cross-road of garden-path.

Part-of-speech (POS) ambiguity is ubiquitous among English words. A noun is often a verb or an adjective. The parser is often puzzled by such words when it cannot decide whether to start parsing an NP or a VP, or whether to end an NP for a possible VP or go on taking one more noun for the current NP. The most notorious POS ambiguity is that of verbs with -ed or -ing ending. Words ending with -ing are potential noun, verb or adjective. The ambiguity affects constituency as well as relation type assignment. Words ending with -ed add an extra layer of ambiguity between active and passive. In terms of constituency ambiguity, words of multiple POS's involve garden-path because they put the parser at a cross-road all the time.

A special kind of POS ambiguity involves function words such as *that, as, for, to*, etc. The word *that* might begin an NP, a relative clause or a noun clause. The preposition *to* signifies the beginning of a PP or an infinitival, whereas *as* might start a PP, a subordinate clause or a relative clause. These words lead to garden-path of all kinds.

3.7 Context Grammar Parser

The parser is based on a context grammar for several reasons: 1) The parser lets the context disambiguate POS and word senses by giving each word only one entry in the lexicon so as to free the parsing from selecting a sense out of several lexical meanings; 2) Each entry of word is provided with multiple POS's if the word has more than one possible syntactic category so that the parser can test on possible POS and pick one among the candidates according to the context; 3) Different senses of the same part of speech will be disambiguated based on different lexical features or subcategories. For example, most verbs are potentially transitive and intransitive, but only transitive verbs can be passive. A passive context will decide that the verb is transitive. Similarly, a verb taking a person Patient, an event Patient or a clause Patient will eventually let the collocation context decides its own lexical sense. In other words, there is no need to burden the lexicon with several predetermined senses and further burden the parser with unnecessary decision making that is unattainable without accommodating the lexicon with the entire world knowledge.

The parser is taught to use the left context that has been decided by the words already parsed and the right context made available by all the unparsed words. The parser can check on every word in the sentence in terms of POS, subcategories and any other lexical feature. For the words in the left context, the information derived from the grammar and the finished parse will tell the parser where the current word is situated, in what type of clause or phrase it is, inside a Subject or still expecting an Object, inside a series of NPs or parallel clauses, and so on. All these are made possible by having the parser registers structured hierarchically.

4 Meaning representation

Historically, many AI or NLP (natural language processing) systems preferred logical forms to other forms of meaning representation for an obvious reason in accessibility. However, this advantage can also be achieved even with natural-language-like representation if it turns into a structured data type from the unstructured text. For this very reason, the parser in this study outputs Excel-like tables to represent the meaning of each sentence with its automatic annotation, i.e. adding new derived information back to the document.

As for the content, the meaning representation used here is based on the author's three aspects of meaning (Chen 1996): referential, relational and specificational. The author believes every text or sentence is all about referents and relations among them. However, words, phrases or clauses only go so far as designating possible entities and possible worlds. This is why specificational meaning is added to referents and relations. In this parser's output representation, three tables are generated from parsing: an NP table that annotates all NPs discovered; a case frame table that annotates each new-found clause with "Who Did What to Whom" event representation plus an

inter-event relation; a term definition table that annotates sentences in which certain terms are defined.

Four advantages result from this meaning representation. First, it outputs searchable data which can be merged easily. Second, the three tables are present in a single XML file, i.e. the database contains everything that is needed for information extraction. Third, it is natural-language-like. There is no need to depend on some artificial symbolic forms to make it accessible or readable only by machines. Human readers or reviewers need no additional training to use the database or evaluate the system performance. Four, necessary PADS restorations have been done in the annotation so that no gaps will hinder human comprehension or machine reading. For underachieved readers, the filling of the gaps makes the sentences easy to understand. For computer systems or search engines, the explicit information added by PADS restoration makes it a powerful tool for unearthing buried information and hidden links.

Table 2 shows the case frames of the sentence *He first examined his childhood memories and came to realize the intense hostility he had felt for his father*. Three rows of "Who Did What to Who" are shown in the table. They indicate a success in doing clause boundary detection. In terms of PADS restoration, two pronouns are found in the sentence. The nominative *he* is given back its co-referent *Freud*, which is absent in this sentence. However, the parser manages to restore the co-reference by getting the right one from previous sentences.

Agent	Predicate	Main Verb	Patient	Goal	EventRelation
He: Freud {pro-ana}	examined	examined	his childhood memories		m-clause=
He: Freud {zero-ana}	came to realize	realize	the intense hostility		conjoin= m-clause=
He: Freud {pro-ana}	had felt	felt	the intense hostility {zero-ana} {rel-ana} {trace}	for his father	branch=rel

Table 2: case frame

The annotation {pro=ana} is to show that co-referent *Freud* is restored for pronoun *he* while {zero-ana} is to show that Subject of the second clause is missing because the first two clauses conjoin to each other by *and*, meaning that the Subject of the second clause is omitted to avoid repetition and it is restored by zero-anaphor resolution. The conjoining of these two clauses is indicated by inter-event relation "conjoin" while "m-clause" is to signify "main clause". A relative clause is added to the second clause as the right-branched modifier of *the intense hostility*. Since the relative pronoun is omitted, {zero-ana} is used to show the effect of zero-anaphor resolution. While the omitted relative pronoun is supposed to replace the antecedent, relative anaphor resolution is involved. Furthermore, since the antecedent is originally the Object of the kernel relative, a trace is left behind. It is then moved back to the Object position, thanks to successful case frame building. As to sentence construction principle, it's a branching relation between the head NP and the relative clause. Table 2 shows that the parser is able to capture both referential meaning in each Case Role and relational meaning inside the case frame (intra-event relation) and between two events (inter-event relation). The aspect of specificational meaning is evident in this table from having *came to realize* as the filler of the Predicate slot. There is no event of *coming* here. The parser treats *came to* as a specifier of the verb *realize*. *Came to realize* presents a particular world out of the possible worlds denoted by *realizing*.

5 Effects of PADS restoration

Up to now, most language parsers only do sentence parsing. A system can go around the limitation of sentence parsing by using mention-lists to do co-reference resolution for pronouns. As emphasized by Halliday and Hasan (1976), text meaning should be treated as going beyond sentence boundaries. They treat cohesive devices as the texture to better organize the text. The use of PADS restoration by this parser identifies cohesive devices or PADS transformations from parsing and restores what are made implicit. Such a technology benefits both computer systems in information extraction or any search engine, and human readers in overcoming the comprehension gaps left behind by PADS. Here are some examples of its application.

5.1 Extended definition

Typically researchers rely on definition words to find defined terms in sentences. However, to avoid repeating a term so often in writing, the original term is replaced by a pronoun or omitted as known. When *it* is defined, the definition will be missed. This is why PADS restoration finds itself a good use, that is, to find extended definitions for a given term. This is also made possible by the structured meaning representation that is sorted and searchable. Notice that the examples underlined in Table 3 might be past unnoticed because of the missing of the term *the id* or because it is not in a position indicating that a term is defined.

Term	Cohesion	S
the id		The id is ***the biological component***, the ego is the psychological component, and the superego is the social component.
the id		THE ID -- The id is ***the original system of personality***; at birth a person is all id.
the id		The id is ***the primary source of psychic energy and the seat of the instincts***.
the id	it: the id	It ***lacks organization and is blind, demanding, and insistent***.
the id	it: the id	***A cauldron of seething excitement***, the id ***cannot tolerate tension***, and ***it functions to discharge tension immediately***.
the id	(the id) ruled by	***Ruled by the pleasure principle, which is aimed at reducing tension, avoiding pain, and gaining pleasure***, the id is ***illogical, amoral, and driven to satisfy instinctual needs***.
the id	(the id) remaining	The id ***never matures, remaining the spoiled brat of personality***.
the id	it: the id	It ***does not think but only wishes or acts***.
the id		The id is ***largely unconscious, or out of awareness***.
the id		***The ego***, as the seat of intelligence and rationality, ***checks and controls the blind impulses*** of the id.
the id		Whereas the id ***knows only subjective reality***, the ego distinguishes between mental images and things in the external world.
the id	it: anxiety	***It develops out of a conflict among the id***, ego, and superego over control of the available psychic energy.

Table 3: Extended definition

5.2 Debugging tool

Three sentences are used in this section to show why the parser's meaning representation is a great tool for evaluation and debugging. In Table 4, the underlined words and phrases are erroneous. In the first clause of the sentence *Freud's family background is a factor to consider in understanding the development of his theory*, the infinitival should not be placed under label Goal because it is the modifier of the NP *a factor*, which should be a non-finite clause "branching" from the NP. In other words, it should be part of Patient and there should be no Goal in this clause. This is a mistake of attachment and constituency. The error comes from the parser's negligence to attach an infinitival to verb-to-be. Adding a test for the verb-to-be not to take a Goal and forcing the NP following verb-to-be to take the modifier, the parser should be able to make it right.

Agent	Predicate	Main Verb	Patient	Goal	EventRelation
Freud's family background	is	is	a factor	***to consider in understanding the development of his theory***	m-clause=
Freud's family background	to consider	consider	————	in understanding the development of his theory.	branch=to
dummy-subject	understanding	understanding	the development of his theory		embed=ing
Freud's family	***had limited***	***limited***	***finances***		branch=sub
Freud's family	was forced to live	live		in a crowded apartment	conjoin=branch=sub
his parents	made	made	every effort	to foster his obvious intellectual capacities	m-clause=

his parents	to foster	foster	his obvious intellectual capacities		branch=to
	settled	settled	***He: Freud***	on medicine	branch=ed

Table 4: Evaluation and debugging

The second mistake comes from whether verb-to-be is subject-control or object-control. For subject-control verbs, the parser is taught to restore the Subject of the infinitival by using the Subject of the matrix clause, whereas an object-control verb will cause the parser to borrow matrix Object to be the Subject. Unfortunately such a consideration forces the parser to make a wrong decision and take *Freud's family background* as the Subject. In fact, verb-to-be is neither a subject-control verb nor an object-control verb. The burden is still on verb-to-be. By then, the consideration of the NP modifier will force *the factor* to be the Patient because *the factor* is a non-person. This one is very difficult for most parsers.

In terms of clause boundary detection, Table 4 shows a 100% recall of 8/8 but only an 87.5% precision of 7/8 from parsing the three sentences.

For sentence *Even though Freud's family had limited finances and was forced to live in a crowded apartment, his parents made every effort to foster his obvious intellectual capacities*, the only mistake actually comes from POS ambiguity for the word ending with -ed. Since verb-to-have is usually followed by past participle to form a perfective aspect and the plural noun *finances* does not require a determiner, the right constituency of "*had + limited finances*" is mistaken as "*had limited + finances*." This error caused by ambiguous -ed is hard to do right. The only solution might come from using the very low possibility for the word *finances* to function as a countable noun. Nevertheless, the possibility is still not zero.

Similar -ed ambiguity occurs to the third sentence *He finally settled on medicine*. There are more transitive verbs than intransitive in English. Although not all transitive verbs can be passive, a majority of them have passive form. The ambiguity between active and passive has the potential to ruin case role assignment. Passive reduced relatives cannot rely on verb-to-be to pronounce passiveness because it is omitted. As a result, the parser usually depends on a preposition right behind the -ed verb to make the right call, as in this case. However, it is not entirely dependable with the complication caused by two-word verbs. Two-word verbs refer to transitive phrasal verbs, which are passive only when there is another preposition following the second element of the phrasal verbs. In this case, *settled on* should be active because it is a two-word verb.

6 Building of knowledge base out of automatic computer reading

Although the rule-based parser used in this study is slow comparing with most statistical shallow parsers. It is less ambiguous and more powerful in terms of the range of NLP tasks it is able to perform. Nevertheless, it is still faster than human readers. In addition, human reading is characteristic of leaving no record after reading. When a reader finishes reading a book, everything he or she has learned is inside the brain as invisible imprint and only the reader can access it mentally. Computer reading is different. The parser is taught to keep all the reading "results", filed and well-structured for open access. From all the Excel-like tables, we can create knowledge base to serve as annotated surrogate documents for an article, a book, or even a corpus. The knowledge base will then become very powerful corpus to support CALL, information extraction or even knowledge discovery. Such knowledge bases are different from most available corpus tools in that they have precise pieces of information as to telling exactly "Who Did What to Whom" in each clause or event and how events are related and linked, not relying mainly on distance or regular-expression rules to do concordance, collocation, chunking or bag-of-terms search.

6.1 Writing tool for CALL

With all the inter-event relations annotated, the knowledge base can be used to teach how to use different kinds of structure to do, for instance, embedding in an English writing class aimed at teaching sentence making rules.

embed=ing	Freud devoted most of his life to ***formulating and extending his theory of psychoanalysis***.
embed=to	It is a mistake ***to assume that all feelings clients have toward their therapists are manifestations of transference***.
embed=n-	It is a mistake to assume ***that all feelings***