A Note-taking Appliance for Intelligence Analysts

Ronald M. Kaplan, Richard Crouch, Tracy Holloway King,
Michael Tepper, Danny Bobrow
Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, California, 94304, USA
kaplan@parc.com

Keywords: Novel Intelligence from Massive Data, Knowledge Discovery and Dissemination, Usability/Habitability, All Source Intelligence

Abstract

Note-taking is a very simple and quite common activity of intelligence analysts, especially all-source analysts. Common as this activity is, there is little or no technology specifically aimed at making it more effective and efficient: it is mostly carried out by cumbersome copy-paste interactions with standard applications (such as Internet Explorer and Microsoft Word). This paper describes how sophisticated natural language processing technologies, user-interest specifications, and human-interface design have been integrated to produce a lightweight, fail-soft appliance aimed at reducing the cognitive load of note-taking. This appliance condenses user-selected source passages and adds them to a note-file. The condensations are grammatical, preserve relations of interest to the user, and avoid distortions of meaning.

1. Introduction

Note-taking is a very simple but quite common activity of intelligence analysis, especially for all-source analysts. They read documents that come across their screens, often from web searches, identify interesting tidbits of information, and make notes of their findings in a separate "shoebox" or note file for later consideration and report preparation. Common as this activity is, there is little or no technology specifically aimed at making it more effective and efficient: it is mostly carried out by cumbersome copy-paste interactions with standard applications (Internet Explorer, Microsoft Word).

This paper describes how sophisticated natural language processing technologies, user-interest specifications, and human-interface design have been integrated to produce a proof-of-concept prototype of a light-weight appliance that reduces the cognitive load of note-taking. After the analyst highlights a relevant passage in a source-document browser window, a single key-stroke causes an interest-sensitive condensation of the passage to appear in the shoe-box. A profile of interesting topics can be associated with a current project and is easy to specify and modify. Uninteresting aspects of the passage are dropped out of the note, but the NLP technology ensures that the condensation is grammatical (and thus easily readable) and that it does not distort the meaning of the original. The original passage is retained with the note and can be popped up for later review. A source identifier (e.g. URL) is also kept with the note, again for later and more detailed consideration of the full document. The appliance presents an elegantly simple user interface, and it is also fail-soft: if the automatic condensation technology misses or misrepresents a crucial part of the passage, the analyst can just edit the note in the shoebox—in the worst case reverting to what is the best case of current approaches, hand editing.

This paper briefly describes the note-taking appliance from two perspectives. In the next section we discuss the appliance from the point of view of the user, indicating how the user interacts with the appliance to add specific items of interest to the note-file. Subsequently, we briefly outline some of the underlying language processing mechanisms that support the functionality of the appliance. We indicate the mechanisms by which a selected passage is condensed to a grammatical reflection of its salient meaning, and how the condensation process is made sensitive to specifications of the user's interests.

At the outset it is important to stress the difference between note-taking as contemplated here and summarization. A summarizer typically operates on a document off-line and as a whole, attempting to identify automatically the key sentences or paragraphs that are particularly indicative of the overall content. The summary is then assembled by concatenating together those identified chunks of text with little or no additional processing. In contrast, a note-taker is a tool tightly integrated into the analyst's on-line work process. The analyst, not the system, decides which passages to select, and the system operates within the passage sentences to eliminate uninteresting or unimportant detail.

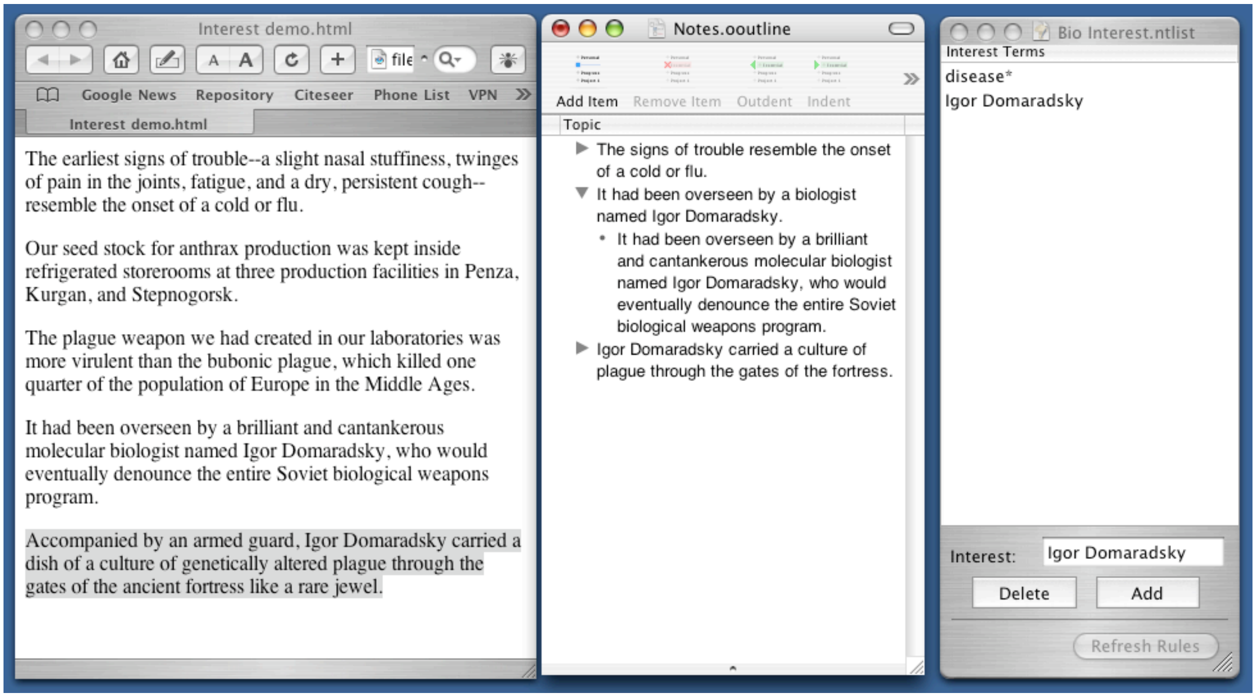

Figure 1. Screen image showing source browser window (left), note-file (middle), interest-profile (right).

2. The Note-Taking Interface

The basic set up is illustrated in the Macintosh screen-image shown in Figure 1. On the left is a window of a standard browser (in this case, the Macintosh Safari browser) that displays portions of a text that the analyst has been reading (in this case, some sentences from Alibek 1999). The analyst has used the mouse in the ordinary way to highlight a passage containing information that he would like to see inserted into his note-file. The note-file is shown in the middle window; in the prototype appliance the note-file is maintained by OmniOutline, a standard outline application on the Macintosh, Whenever a hotkey is pressed in the browser (or any other similarly configured text-reading application), the currently selected passage is carried over for insertion into the note-file.

There are two parts to the insertion. A condensed version of the passage is computed, and it is entered as the header of an outline item. The original passage and information about its source are entered as the body of that new item. The screen image illustrates the situation immediately after the sentence "Accompanied by an armed guard, Igor Domaradsky carried a dish of a culture of genetically altered plague through the gates of the ancient fortress like a rare jewel." The note-taker has computed the condensation "Igor Domaradsky carried a culture of plague through the gates of the fortress." The item-header is much shorter than the original passage because uninteresting, even if poetic, descriptions have been omitted (armed guards, dish, rare jewel). But the header is a well-formed grammatical sentence and hence easy to understand during later review.

The original passage is available as the item-body and will be revealed if the user clicks on the disclosure triangle. Indeed, the previous item has been opened up in that way, so that both the condensation and the original passage are displayed. The original passage is also useful for later review: the analyst can easily drill down to see the detail and context of the information that appears in the summary condensation. Also, if crucial data is left out of the automatically generated condensation, the user can promote the passage from body to header and construct a note by hand-editing. The analyst is thus protected from errors that might occasionally be made by the automatic machinery, since he can quickly and easily construct his own abbreviation of the passage.

The condensation for a given passage is determined by a deep linguistic analysis, as briefly described below (also see Riezler, *et al.* 2003; Crouch *et al.* 2004). The passage is parsed into a representation that makes explicit the predicate-argument relations of the clauses it contains, identifies modifiers and qualifiers, and also recognizes subordination relationships that hold between clauses. General rules are used to eliminate pieces of this representation that are regarded as typically uninformative. These rules delete modifiers, appositives, subordinate clauses, and various other flourishes and excursions from the main line of discourse. The representation that remains after the deletion rules have operated is converted back to a well-formed sentence by a grammar-based generation process.

The general deletion rules are constrained in two different ways. First, they are not allowed to delete terms in the representation that refer to concepts or entities that are known to be of specific interest to the analyst. The note-taking appliance is thus sensitive to the user's interests and not just to properties of grammatical configurations. The window on the right of the screen image illustrates one way in which the user's interests may be determined, namely, by explicit entries in a file containing a user-interest profile. In this example, the user has indicated that he is interested in any sort of diseases and in any reference to Igor Domaradsky. This ensures that "plague" and "Igor Domaradsky" are maintained in the example while "jewel" and "guards" are discarded. If "jewel" was included as a term of interest, it would have been retained in the condensation.

The interest profile can specify particular entities or classes of entities picked out by nodes in an ontology or other knowledge base. The specification "disease*" declares that all entities classified as diseases are of interest, and the analyst does not have to list them individually. The interest profile can also includes terms that are marked as particularly uninteresting, and the note-taker will make every effort to form a grammatical sentence that excludes those terms.

The interest profile in principle may also be determined by indirect means. Observations of previous browsing patterns may indicate that the analyst is drawn to sources containing particular terms or entity-classes, and these can be used to control the condensation process. A user's interests may also be project-dependent, with different profiles active for different tasks. And interest specifications may be shared among members of a team who are scanning for the same kinds of information. These possibilities will be explored in future research,

As a second constraint on their operation, general deletion rules are prohibited from removing pieces of the representation if it can be anticipated that the resulting condensation would distort the meaning of the original passage. A trivial case is the negative modifier "not". It is not appropriate to condense "Igor did not carry plague" to "Igor carried plague", even though the result is shorter, because the condensation contradicts the original. On the other hand, "Igor carried plague" is a reasonable condensation of "Igor managed to carry plague", because in this case the condensation will be true whenever the passage is. Meaning-distortion constraints can be quite subtle: "Igor caused a serious epidemic" can be condensed to "Igor caused an epidemic", but "Igor prevented a serious epidemic" should not be condensed to "Igor prevented an epidemic." In the latter case there could still have been an epidemic, but not a serious one.

In sum, the prototype note-taking appliance presents a very simple, light-weight interface to the analyst. The analyst works with his normal source-reading applications augmented only by a single note-taking hot-key. The notes and passages show up in a simple outline editor with visibly obvious and fail-soft behaviors.

3. Under the Covers

While the note-taking appliance creates the illusion of simplicity at the user interface, the production of useful condensations in fact depends on a sequence of complex linguistic transformations. Obvious expedients, such as chopping out uninteresting words, would leave mangled fragments that are difficult to interpret. Even for the trivial example of "Igor managed to carry plague", deleting "managed to" would produce the ungrammatical "Igor carry plague", with the verb exhibiting the wrong inflection.

Instead, we parse the longer sentence into a representation that makes explicit all the grammatical relations that it expresses, including in this case that "Igor" is not only the subject of "managed" but also the understood subject of the infinitive "carry". Condensation rules preserve the understood subject relation when "managed" is discarded, and a process of generation re-expresses that relation in the shortened result. The effect is to properly inflect the remaining verb in the past tense.

We use the well-known functional structures (f-structures) of Lexical Functional Grammar (LFG) (Kaplan and Bresnan 1982) as the representations of grammatical relations that the parser produces from the passages the user selects. The parsed f-structures are transformed by the condensation rules, and the generator then converts these reduced f-structures to sentences.

To be concrete, the f-structure representation for "Igor managed to carry plague" is the following:

```
[PRED     'manage<[1:Igor], [97:carry]>'
 SUBJ   1[PRED 'Igor']
          [PRED     'carry<[1:Igor], [144:plague]>'
           SUBJ     [1:Igor]
 XCOMP     OBJ   144[PRED 'plague']
        97 TNS-ASP  [PERF -_, PROG -_]]
 TNS-ASP [PERF -_, PROG -_, TENSE past]]
```

This shows that "Igor" bears the SUBJect relation to both PREDicates, and that "plague" is the OBJect of "carry". The condensation rules copy the past-tense from the outer structure to the complement (XCOMP) structure and then remove all the information outside of the XCOMP, resulting in

```
[PRED     'carry<[1:Igor], [97:plague]>'
 SUBJ   1[PRED 'Igor']
 OBJ   97[PRED 'plague']
 TNS-ASP [PERF -_, PROG -_, TENSE past]]
```

The generator re-expresses this as "Igor carried plague".

The mappings from sentences to f-structures and from f-structures to sentences are defined by a broad-coverage LFG grammar of English (Riezler *et al.* 2002). This grammar was created as part of the international Parallel Grammar research consortium, a coordinated effort to produce large-scale grammars for a variety of different languages (Butt *et al*. 1999). The parsing and generation transformations are carried out by the XLE system, an efficient processor for LFG grammars. XLE incorporates special techniques for avoiding the computational blow-up that often accompanies the rampant ambiguity of natural language sentences (Maxwell and Kaplan, 1993). This enables condensation to be carried out for most sentences in a user-acceptable amount of time; the system goes into fail-soft mode and inserts the original passage when a reasonable time-bound is exceeded.

F-structure reduction is performed by a rewriting system that was originally developed for machine translation applications (Frank 1999), and in a certain sense condensation can be regarded as the problem of translating between two languages, "Long English" and "Short English" (Knight and Marcu 2000). A useful reduction, for example, eliminates various kinds of modifiers, so that the key entities mentioned in a sentence stand out in the note. This transformation is specified by the following rule:

ADJUNCT(%F,%A) inset(%A,%M) ?=> delete(%M)

Modifiers appear in f-structures as elements of the set value of an ADJUNCT attribute. The variable %F matches an f-structure with adjunct set %A containing a particular modifier %M, and the rule then optionally removes %M from that f-structure. This rule would apply to eliminate the modifier "carefully" from the following f-structure for the sentence "Igor carefully carried a dish of plague":

```
PRED      'carry<[1:Igor], [143:dish]>'
SUBJ    1 [PRED 'Igor']
            [PRED     'dish'
             ADJUNCT  { 186 [PRED   'of<[194:plague]>'
OBJ                          OBJ 194 [PRED 'plague']] }
        143  SPEC     [DET [PRED 'a']]]
ADJUNCT  { 31 [PRED 'carefully'] }
TNS-ASP  [PERF -_, PROG -_, TENSE past]
```

The rule is optional because it may or may not be desirable to delete a particular adjunct. Thus, this same rule could be applied (optionally) also to delete the "plague" modifier of "dish", so altogether there are four possible outcomes of this rule, corresponding to the sentences:

Igor carefully carried a dish of plague.
Igor carried a dish of plague.
Igor carefully carried a dish.
Igor carried a dish.

In the absence of further constraints, we might apply a statistical model to choose the most probable condensation (Riezler *et al*. 2003), and this might very well select the last (and shortest) of the four candidates.

However, this choice would not respect the specifications of user interest shown in the right window of Figure 1. The user has indicated (by "disease*") that anything classified as a disease is of interest and must be preserved in the note. By consulting an ontological hierarchy we discover that plague is a kind of disease. This rule therefore cannot apply to the ADJUNCT "of plague", so only the first two sentences are produced as candidates. The statistical model then might choose the second of the two. A more desirable version of the modifier-deletion rule further restricts its application to avoid meaning distortions that arise when modifiers are eliminated in the scope of verbs like "prevent" as oppose to "cause".

Another example illustrates how rules can make direct appeal to ontological information in addition to the way a concept hierarchy can extend the domain of interest:

PRED(%F,%P) ADJUNCT(%F,%A) inset(%A,%M)
Container(%P) PRED(%M,of)
?=> delete-between(%F,%M)

This rule matches f-structures whose PREDicate value is classified by the ontology as a Container and which has an ADJUNCT marked by the preposition "of". This implements the principle that the material in a container is generally more salient than the container itself. Assuming "dish" is a Container, this matches the OBJect f-structure, and the effect is to remove the "dish" and promote the "plague" to be the OBJect of "carry". The generator would re-express the result as "Igor carried plague". A passivization rule could produce the slightly shorter condensation "Plague was carried", but this would eliminate Igor, a person of interest.

Our note-taking appliance thus depends on a substantial amount of behind-the-scene linguistic processing to produce the grammatical and interest-sensitive condensations that appear in the note-file. Operations on the underlying grammatical relations represented in the f-structure stand in contrast to the word and phrase chopping methods that have been applied, for example, to the simpler problem of headline generation (Dorr *et al.* 2003). Figure 2 is an architectural diagram that shows the pipeline of parsing, condensation, stochastic selection, and generation that we have briefly described. The figure also shows the data-set resources that control the process. LFG grammars can be used equally well for parsing and generation, so a single English grammar determines both directions of the sentence-to-f-structure mapping. A set of condensation rules produce a fairly large number of candidate condensations, but the same ambiguity management techniques of the XLE parser/generator are also used here to avoid the computational blow-up that optional deletions would otherwise entail. Condensation is also constrained by interest specifications obtained from the analyst, and by the concept classifications of an ontological hierarchy.

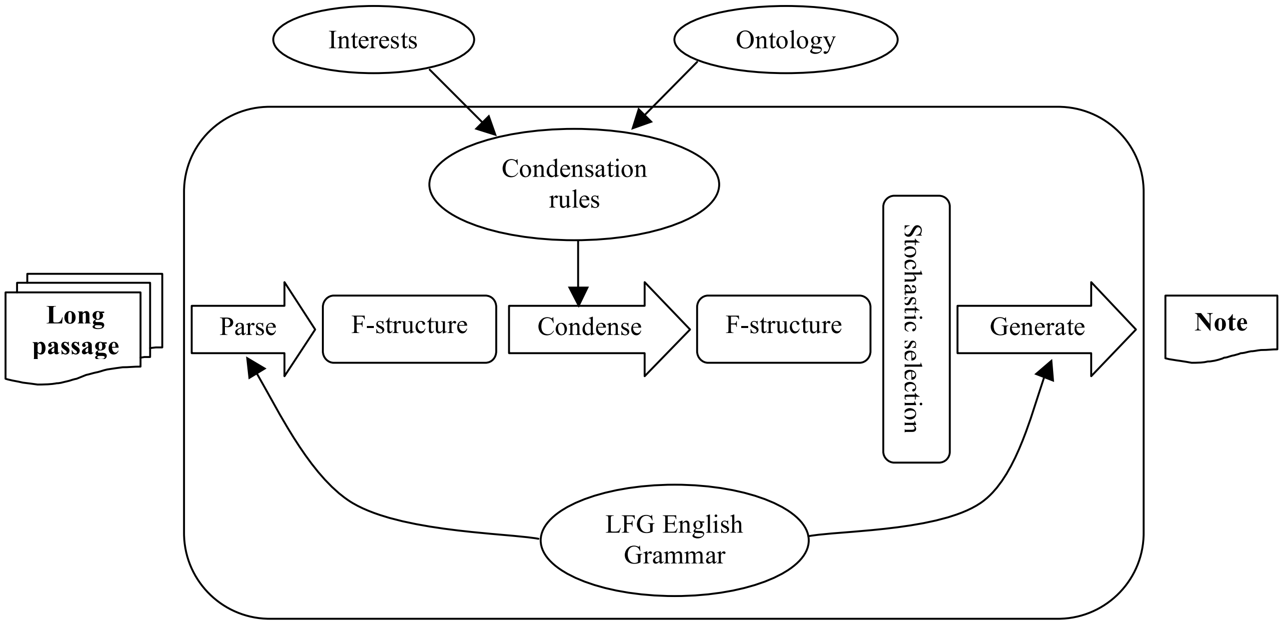

Figure 2. Architectural diagram of underlying language processing components.

4. Summary

We have described a prototype note-taking appliance from two points of view. The analyst sees the appliance from the outside as a light-weight application that reduces the cognitive load of note-taking. He reads a source document with a normal browser, from time to time highlighting passages to be reflected in a note-file. Rather than a copy-paste-edit sequence of commands, a single key-stroke creates a grammatical condensation of the passage that eliminates information that does not match the analyst's interest profile. The condensation is moved to the note-file but is also accompanied by the full passage so that it is available for later drill-down review.

We have also described the system from the inside, indicating how a number of sophisticated natural language processing components are configured to create the grammatical condensations at the simple user interface. The passage is parsed into an f-structure according to the specifications of an LFG grammar, this is transformed to a smaller structure by interest-constrained condensation rules, and an LFG generator produces a shortened output sentence.

At this stage our system clearly is only a prototype and further research and development must be carried out before its effectiveness in an operational setting can be evaluated. We must port the appliance to the computing platforms that analysts typically use and tune the system to relevant tasks and domains. We expect to extend and refine our initial condensation rules and background ontology, to implement new ways of inferring user interests, and to determine better stochastic selection parameters on the basis of larger and more representative training sets. We must also understand how to integrate the appliance into the analysts' work routine for maximum gains in productivity.

We suggested at the outset that note-taking is a common activity of intelligence analysts, especially all-source analysts, and that little attention has been directed towards the problem of making this activity more effective and more efficient. Our prototype appliance combines a simple front-end with complex back-end processing in a fail-soft application intended to reduce the cognitive load of note-taking.

Acknowledgments

This research has been funded in part by contract # MDA904-03-C-0404 of the Advanced Research and Development Activity, Novel Intelligence from Massive Data program.

References

Alibek, K. 1999. *Biohazard*. New York: Dell Publishing.

Butt, M.; King, T.; Niño, M-E.; and Segond, F. 1999. *A grammar writer's cookbook.* Stanford: CSLI Publications.

Crouch, R.; King, T.; Maxwell, J.; Riezler, S.; and Zaenen, A. 2004. Exploiting f-structure input for sentence condensation. M. Butt and T. King (eds.), *Proceedings of the LFG04 Conference. Stanford: CSLI Publications*. http://csli-publications.stanford.edu

Dorr, B.; Zajic, D; and Schwartz, R. 2003. Hedge Trimmer: A parse-and-trim approach to headline generation. *Proceeddings of the HLT-NAACL Text Summarization Workshop and Document Understanding Conference (DUC 2003),* Edmonton, Canada.

Frank, A. 1999. From parallel grammar development towards machine translation. *Proceedings of MT Summit VII*. "*MT in the Great Translation Era*", September 13-17, Kent Ridge Digital Labs, Singapore, 134-142,

Kaplan, R. and Bresnan, J. 1982 Lexical Functional Grammar: A formal system for grammatical representation. J. Bresnan (ed.), *The mental representation of grammatical relations*. Cambridge: MIT Press, 173-281.

Knight, K. and Marcu, D. 2000. Statistics-based summarization—step one: Sentence compression. *Proceeedings of the 17th National Conference on Aritificial Intelligence (AAAI-2000)*, Austin, Texas.

Maxwell, J. and Kaplan, R. 1993. The interface between phrasal and functional constraints. *Computational Linguistics*, 19, 571-590.

Riezler, S.; King, T.; Kaplan, R.; Crouch, R.; Maxwell, J.; and Johnson, M. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, Philadelphia, 271-278.

Riezler, R.; King, T.; Crouch, R.; and Zaenen, A. 2003. Statistical sentence condensation using ambiguity packing and stochastic disambiguation methods for Lexical-Functional Grammar. *Proceedings of the Human Language Technology Conference and the 3rd Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'03)*, Edmonton, Canada.