Semantic Forensics:
An Application of Ontological Semantics to Information Assurance

Victor Raskin, Christian F. Hempelmann, and Katrina E. Triezenberg
NLP Lab and CERIAS
Purdue University
{vraskin, hempelma, kattriez}@purdue.edu

Abstract

The paper deals with the latest application of natural language processing (NLP), specifically of ontological semantics (ONSE) to natural language information assurance and security (NL IAS). It demonstrates how the existing ideas, methods, and resources of ontological semantics can be applied to detect deception in NL text (and, eventually, in data and other media as well). After stating the problem, the paper proceeds to a brief introduction to ONSE, followed by an equally brief survey of our 5-year-old effort in "colonizing" IAS. The main part of the paper deals with the following issues:

- human deception detection abilities and NLP modeling of it;
- manipulation of fact repositories for this purpose beyond the current state of the art;
- acquisition of scripts for complex ontological concepts;
- degrees of lying complexity and feasibility of their automatic detection.

This is not a report on a system implementation but rather an application-establishing proof-of-concept effort based on the algorithmic and machine-tractable recombination and extension of the previously implemented ONSE modules. The strength of the approach is that it emphasizes the use of the existing NLP applications, with very few domain- and goal-specific adjustments, in a most promising and growing new area of IAS. So, while clearly dealing with a new application, the paper addresses theoretical and methodological extensions of ONSE, as defined currently, that will be useful for other applications as well.

1 The Problem

The proposed application falls within the rapidly growing domain of cyber forensics, and the inclusion of NLP in it is a desirable and mutually beneficial goal. Forensic science, in general, encompasses any scientific discipline that is concerned with the direct application of scientific principles and theories to law enforcement (Saferstein 2004), that is, the systematic search for, discovery, and application of clues to event reconstruction—for the purposes of justice. As with any of the other traditional forensic sciences (e.g., DNA, Serology, Latent Fingerprint Analysis etc.) the development of the cyber forensics

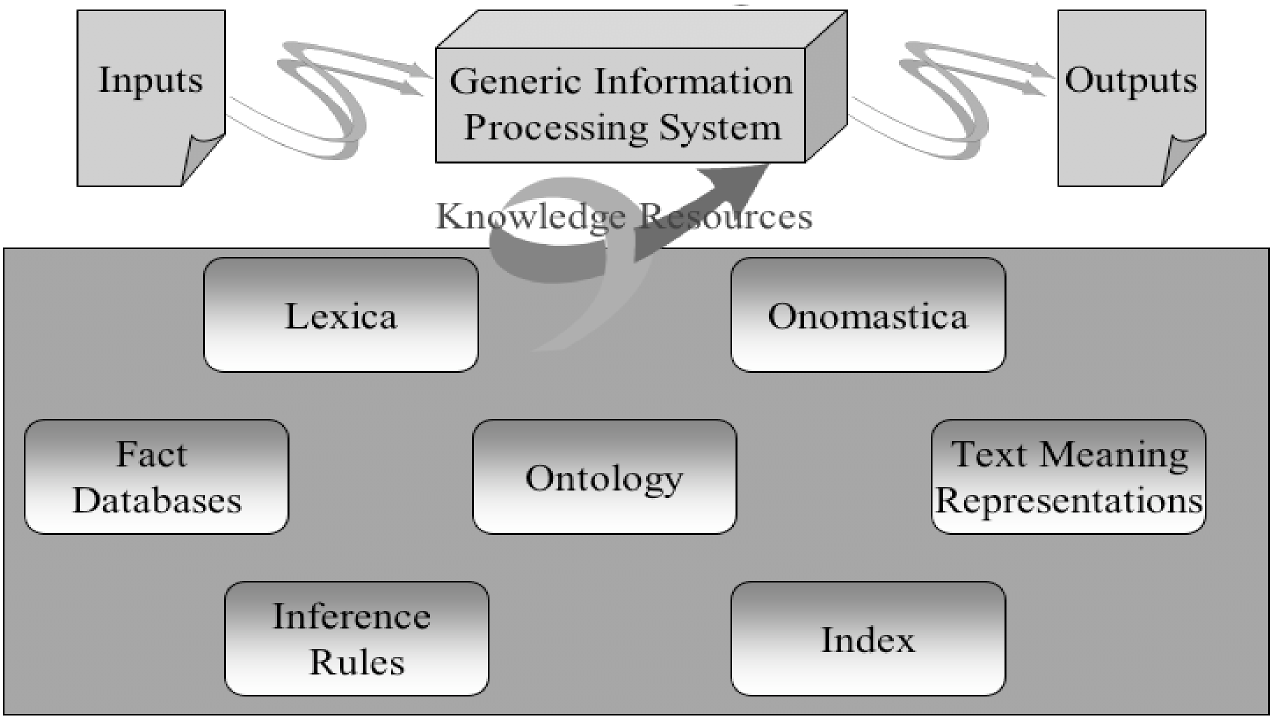

Figure 1: Resources of Ontological Semantics

discipline is based on a sound scientific foundation, compliance with legal requirements, and a unified maturation process (Palmer 2002; Rogers & Seigfried 2004; Whitcomb 2002). The current foundation for cyber forensics is multidisciplinary in nature and combines established pure sciences (e.g., computer science, math) and applied sciences (e.g., information technology, engineering, and now NLP). Cyber forensics is still in search of its theory and methodology; NLP comes into it on strong and explicit theoretical foundations (see Nirenburg and Raskin 2004: 34-91).

As far as we know, no similar research has been undertaken yet. The Patrick group at the University of Sydney reported on a "bag-of-words" system comparing the Nigerian e-mails with Reuters' financial reports and concluding that the presence of the personal pronouns *I/me* and *you* in the former but not in the latter violates the interpersonal relations of the register in the former, thus leading to characterizing them as scams (Herke-Couchman *et al.* 2003; Whitelaw and Argamon 2004). The difference in the proposed approach is that it goes beyond treating words as just character strings and represents their meanings, as well as those of the sentences and of their text, explicitly and manipulates them logically, as it were, rather than statistically. The result, typical for the comparison of computational semantics with computational statistics, is that a semantic forensics system is capable of identifying specific facts/events that contribute to the deception and of understanding what those events are—the characterization of the text as fraudulent comes then as a trivial side effect. The Patrick approach can, however, be used to assists semantic forensics by pre-identifying some texts as suspicious cheaply and thus reducing the general semantic forensics to a more targeted and feasible task (see Section 6 below). Also, their ScamSeek Project has declared an intention to move towards true meaning features (http://www.cs.usyd.edu.au/~lkmrl/scam-seek.htm).

Outside of NLP, the British-centered linguistic forensics community (IAFL) has been engaged in traditional largely qualitative stylistic research in the spirit of the 1960s' text attribution, an effort to break the anonymity of a text and to identify its authorship for the purposes of law enforcement (McMenamin 2002, Gibbons 2003, Olsson 2004).

While other disciplines within cyber forensics explore largely non-textual materials—and those which look at texts, with the above-mentioned exceptions. do not do so linguistically—semantic forensics, as defined here, uses NLP to identify the clues of deception in NL texts in order to reconstruct the described events as they actually occurred. Now, it can be argued, with reason, that the truthful elements in NL texts are also clues for event reconstruction and should be included in semantic forensics, and, of course, in a way they are. But those, if indeed truthful, do not come under the task of reconstructing events; rather, they establish the events. Besides, the truthful elements of NL texts get the text meaning representation (TMR) in the normal course of events, so no special semantic forensic effort needs to be developed with regard to them. This is not set in concrete, however, because the identification and exploration of deception clues clearly involves the non-deceptive TMRs and their fragments.

Semantic forensics is firmly based on ONSE, and a semantic forensic analysis of the text presupposes and follows the regular ONSE process of automatic meaning representation of each input sentence and, ultimately, the text. The next section offers a brief introduction into the ONSE process of meaning representation, with an emphasis on analysis rather than generation and with a bias towards IAS applications. It can be largely skipped by those familiar with ONSE.

2 Ontological Semantics in Brief

ONSE contains several modules, with an ontology at the center; the other important modules are lexicons of languages and a fact repository, in which information about the world is stored, and, of course, the analyzer and generator. The analytical goal of ONSE is to produce a TMR for NL input as well as NL or other output for each TMR (see figure 1).

PAY			
	definition	value	"to compensate somebody for goods or services rendered"
	is-a	value	EVERYDAY-FINANCIAL-EVENT
	subclasses	value	PAY-TAX SUBSCRIBE-TO
	agent	sem	HUMAN
		relaxable-to	ORGANIZATION
	theme	default	MONEY
		sem	COMMODITY
		relaxable-to	EVENT
	patient	sem	HUMAN
		relaxable-to	ORGANIZATION

Figure 2: Ontological Concept PAY

The ontology is a tangled hierarchy (lattice) of concepts, beginning at the root ALL, branching into OBJECT, EVENT, and PROPERTY, and so forth. Each node of the hierarchy is a concept with a set of properties, many of which are inherited from its ancestors, and at least one property other than the IS-A property is distinguished from its parent node as well as from its sibling nodes. The ontological concept for PAY might therefore look like figure 2 (cf. Nirenburg and Raskin 2004: 196ff.).

As we see, the IS-A and SUBCLASSES slots are filled with other ontological concepts, as are AGENT, THEME, and PATIENT, the case-role slots. VALUE, SEM, RELAXABLE-TO and DEFAULT are all facets of their slots.

```
good-adj1
    cat  adj
    SYN-STRUC
        1    root    $var1
             cat     n
             mods    root     good
        2    root    $var0
             cat     adj
             subj    root     $var1
             cat     n
    SEM-STRUC
        modality     type     evaluative
                     value    value              >0.75
                              relaxable-to       >0.6
                     scope    ^$var1
                     attributed-to     *speaker*
```

Figure 3: Lexical Entry for “good”

Lexicons contain the actual words of a language, in contrast to the ontology’s universal, language-independent concepts. The entry for each word in the lexicon contains all possible senses of that word, labeled with a part of speech and a sense number. The lexical entry for the English word *pay* contains three senses, respectively *pay-n1, pay-n2*, and *pay-v1*. Each of the senses is then assigned, most importantly, the information about the acceptable syntactic environments for the sense, or SYN-STRUC, and information about the word’s meaning, or SEM-STRUC. It is in SEM-STRUC that each lexical item is linked to one or more ontological concepts, or to literals. The lexical entry for the English adjective *good* looks something like figure 3.

When a text is fed into the ONSE system, its lexical items are identified, as well as several TMR parameters, such as discourse relations including modalities, aspect, information about ordering and duration in time, style, and sets of concepts working together. The first step in building a TMR is finding meanings for heads of clauses in the syntactic representation of input, which are most commonly verbs. The TMR, however, will typically end up containing more event instances than there are verbs in the original text. After identifying these events, building the TMR is a (non-trivial) matter of fitting all the other information of the text into the filler slots of the events and the additional parameters. In figure 4 are the much-simplified TMRs for three related sentences, which demonstrate how small changes in texts affect TMRs.

```
Who won the match?
win-1
    theme     value sports-match-2
request-information-1
    theme     value win-1.agent

Did Arsenal win the match?
win-1
    agent     value Arsenal
    theme     value sports-match-1
request-information-1
    theme     value win-1

Was it Arsenal who won the match?
win-1
    agent     value Arsenal
    theme     value sports-match-1
request-information-1
    theme     value win-1
modality-1
    type salience
    scope     win-1.theme
    value     1
```

Figure 4: TMR Example

The next section reviews the NL IAS applications discovered and explored—from initial steps to pilot implementations—in the ongoing effort to export NLP into computer and information security.

3 Applications of NLP to Information Assurance and Security

In the last 5 years, a CERIAS-based team led by a computer scientist and an NLP expert has steadily expanded its groundbreaking effort in improving, focusing, and strengthening information assurance and security by applying the NLP resources to them. The result has been a growing number of applications, some of them NL counterparts of pre-existing applications, others NL extensions and developments of known applications, and still others unique to NL IAS. In the most implemented one, NL watermarking (see Atallah *et al.* 2002), a sophisticated mathematical procedure, based on a secret large prime number, selects certain sentences in a text for watermark bearing and transforms their TMRs into bitstrings

Application	Function	Implementation	Reference
Mnemonic String Generator	Generates jingles corresponding to random-generated passwords	Pilot	Raskin et al 2001a
Syntactic NL Watermarking	Embeds the watermark in the syntactic tree of a sentence	Pilot/demo	Atallah et al. 2001
Semantic NL Watermarking	Embeds the watermark in the TMR tree of a sentence	Pilot	Atallah et al 2002
NL Tamperproofing	Embeds a brittle watermark to detect any changes to the text	Pilot	Atallah et al 2002
NL Sanitization	Seamlessly removes and replaces sensitive information	Proof of concept	Mohamed 2001
Automatic Terminology Standardizer	Translates different terminological dialects in IAS into TMRs	Proof of concept	Raskin et al 2002a
Perimeter Protection	Sanitizes outgoing e-mail online	Proof of concept	Raskin et al 2001b
NL Streaming Processor	Interprets incoming information before it is complete	Research	Raskin et al 2002b
NL Steganalysis	Detects the presence of a hidden message	Research	Raskin et al. 2002b
Semantic Mimicking	Creates a meaningful cohesive text to hide a secret message	Research	Bennett 2003
Web Crawler for Planned Attacks	Crawls the web in search of credible information on computer attacks	Research	Raskin et al. 2002b
Ontological support for Non-NL data	Helps to classify incoming strings in a computer attack	Initial Research	Raskin 2004

Table 1: NL IAS Applications

that contribute up to 4 bits per sentence to the watermark. The goal of the software is, of course, to embed a robust watermark in the hidden semantic meaning of NL text, represented as its TMR in tree structure. The NLP role is to "torture" the TMR tree of the sentence, whose contributing bits do not fit the watermark, so that they do. The tool for that is a number of minuscule TMR tree transformations, resulting in such surface changes as *The coalition forces bombed Kabul* → *The coalition forces bombed the capital of Afghanistan*. The applications are summarized in table 1.

4 Human Deception Detection and Its NLP Modeling

Like all NLP systems, a Semantic Forensic (SF) NLP system models a human faculty. In this case, it is the human ability to detect deception (DD), i.e., to know when they are being lied to and to attempt to reconstruct the truth. The former ability is a highly desirable but, interestingly, not necessary precondition for DD (see an explanation below, in the Feasibility section). The latter functionality is the ultimate goal of SF NLP but, like all full automation in NLP, it may not be easily attainable.

Humans detect lying by analyzing meaning of what they hear or read and compare that meaning to other parts of the same discourse, to their previously set expectations, and to their knowledge of the world. Perhaps the easiest lie to detect is a direct contradiction: If one hears first that John is in Barcelona today and then that he is not, one should suspect that one of the two statements is incorrect and to investigate—if one is interested, a crucial point. The harder type of deception to perceive is by omission: The first author was pushed into SF after having read a detailed profile of Howard Dean, then a leading contender for the Democratic nomination in the US 2004 presidential election, and noticed that the occupation of every single adult mentioned in the article was indicated with the exception of the candidate's father, who had been a stockbroker. Glossing over, such as saying that one has not had much opportunity to talk to John lately, which may be technically true, while covering up a major fallout with John, is yet more complicated. And perhaps topping the hierarchy is lying by telling the truth: when a loyal secretary tells the boss' jealous wife that her husband is not in the office because he is running errands downtown, she may well be telling the truth (though not the whole truth—but, realistically, can one tell the whole truth ever?—is it even a useful notion, especially given the fact that language underdetermines reality (cf. Barwise and Perry 1983)); but what she wants to accomplish is for the wife to infer, incorrectly, that this is all the boss is doing downtown. It is the latter, linguistically interesting type that was the focus of Raskin (1987).

A new TMR contradicting a previously processed one should lead to a fact repository flag, and this is where we are moving next.

5 Using the Fact Repository for Deception Detection

The fact repository (FR—see Nirenburg and Raskin 2004: 350-1), so far the least developed static resource in ONSE, records the remembered event instances. In principle, it should record all of them. Realistically, it records them selectively to suit the needs of an implementation. Thus, in CRESP, a small QA system for queries about the 2000 Olympics in Sydney, the FR remembered all the nations, all the participants, all the competitive events, and all the results. A SF NLP system may start at the same level of prominence (and detect one's lie about having participated in the Games and/or achieved a better result), but like almost all NLP systems with reasoning abilities, it will be only as powerful as its FR allows.

A contradiction will be flagged when two TMR (fragments) are discovered: For example, one having been just processed for *John is/was/will be in Barcelona at noon on the 25th (of July 2004)* and the other in the FR for *John is/was/will be in Valencia at noon on the 25th (of July 2004)*—or their paraphrases (see figure 5).

human-17		
	name	John-23
	location	Barcelona
	time	noon, July 25, 2004

human-89		
	name	John-23
	location	Valencia
	time	noon, July 25, 2004

Figure 5: Fact Repository Sample Entries

In the case of Papa Dean's occupation, apparently too shameful for the reporter to mention even after he had divulged the Park Avenue childhood, hereditary Republicanism, and discriminatory country club and even though there are still a few stockbrokers on this side of the bars, the FR will easily detect it by presenting this information, very simplistically, as in figure 6.

To detect a gloss-over, it is not quite enough to receive a new TMR which contains an event involving a different interaction between these two individuals at the same time. The co-reference microtheory (see Nirenburg and Raskin 2004: 301-5) will have to be able to determine or at least to suspect that these events are indeed one and the same event rather than two consecutive or even parallel events. Even the time parameters are not a trivial task to equate, as in the case of *I have not much opportunity to talk to John lately* and *John insulted me last May*. It would be trivial, of course, if the temporal adverbials were *since that night at Maude's* and *that night at Maude's*, respectively, but a human sleuth does not get such incredibly easy clues most of the time and has to operate on crude proximity and hypothesizing. Also helping him or her is a powerful inferencing engine, obviously a must for an NLP system of any reasonable complexity, reinforced by a microtheory of euphemisms, which must contain representative sets of event types that people lie about and of fossilized, cliché-like ways of lying about them, as in *How is this paper?—Well… it's different!*

human-1		
	name	Howard Dean
	age	adult
	occupation	physician
human-2		
	name	Judy Dean
	age	adult
	occupation	physician
human-3		
	name	Papa Dean
	age	adult (very: rather dead, actually)
	occupation	unknown

Figure 6: Fact Repository Sample Entries

The reason we think that the loyal secretary's type of lying is harder to detect is not because it may involve more inferencing of a more complex kind—this is not necessarily so. It has to do with the notion of the whole truth: It is not realistic to expect a human, let alone an SF NLP to suspect any information to be incomplete and subject every single TMR to the 'and what else did he do downtown' type of query. But, in many cases, this is necessary to do, which brings up the useful distinction between general and targeted SF.

6 Feasibility of Semantic Forensic Systems

A general SF (GSF) task is, basically, a fishing expedition. An SF NLP system may indeed expose obvious contradictions and many omissions. It is a long and expensive process, however, definitely overloading the system's FR. Inferring from every established contradiction or omission, while possibly valuable forensically, is an unaffordable luxury in this kind of task. It may, however, be a necessary evil: for instance, if an SF NLP system is to address a source that is known to be tainted or if it to be used to classify texts by the degree of their trustworthiness—quite a possible assignment.

Humans do a degree of general SF under similar circumstances. But even in an exchange without a prior agenda, such as a conversation with a stranger under neutral, casual, indifferent circumstances, the SF/DD module may not be activated unless flagged by, again, a contradiction, an omission, etc. And such a flag will transform general SF into targeted SF (TSF).

Now, TSF is what professional forensics does for a living, and there is no reason why the entry-level SF NLP systems should not be all TSF. Even in the case of the Dean text, a TSF system ("look for anything compromising in the candidate's background") will be able to detect the occupation omission much faster. A TSF is simpler and cheaper, and the FR use is much more reasonable and manageable: it can store only very selective, limited material. The flip side of a TSF system is the easy ability to overlook highly related information an inference away, so we have reasons to suspect that a quality TSF NLP system is not that much simpler than, say, a limited domain GSF system.

What is important to realize is that some NLP systems with SF capabilities are within reach in ONSE, using the already available resources, possibly with some modifications, primarily if not entirely on the static side, and that is not much different than changing domains for a "regular" NLP system (see Raskin *et al.* 2002b).

7 Using Scripts of Complex Events for Deception Detection

A main tool for DD, in particular TSF, is the expansion of the ontology by acquiring scripts of complex events, already found necessary for other higher-end NLP tasks (see Raskin *et al.* 2003).

There are strong connections among elements of many texts. These have to do with the understanding that individual propositions may hold well-defined places in "routine," "typical" sequences of events (often called complex events, scripts or scenarios) that happen in the world, with a well-specified set of object-like entities that appear in different roles throughout that sequence. A script captures the entities of such an event and their temporal and causal sequences, as shown for the complex event BANKRUPTCY in figure 7.

As a general tool in ONSE, the scripts that get instantiated from the text input provide expectations for processing further sentences in a text. Indeed, if a sentence in a text can be seen as instantiating a script in the nascent TMR, the analysis and disambiguation of subsequent sentences can be aided by the expectation that propositions contained in them are instantiations of event types that are listed as components of the activated script.

```
BANKRUPTCY
    is-a          financial-event
    agent         corporation-1
                  human-1
                  lending-institution-1
                  corporation-2
                  human-2
    precondition  approach-bankruptcy
    has-parts     (IF
                          AND
                                  modality.scope = pay
                                  modality.value < .5
                  THEN bankruptcy-chapter-7
                  ELSE bankruptcy-chapter-11)

BANKRUPTCY-APPROACH-STATE
    is-a          financial-event
    agent         bankruptcy.agent
    destination   bankruptcy
                          agent       bankruptcy.agent
    has-parts
       (IF
          AND
             owe
                agent          corporation-1
                               human-1
                beneficiary    human-2
                                  employed-by   corporation-1
                               lending-institution-2
                               corporation-2
                theme          money
             pay
                agent          corporation-1
                               human-1
                beneficiary    human-2
                               lending-institution-1
                               corporation-2
                theme          money
         THEN  bankruptcy
                agent          corporation-1
                               human-1

CONCEAL
    is-a          sales-event
    agent         bankruptcy.agent
    theme         assets
                          owned-by  bankruptcy.agent
    precondition  bankruptcy
                          agent       bankruptcy.agent
    time.sales-event ≥ time.bankruptcy-approach-state
```

Figure 7: Simplified Fragments of Scripts in the BANKRUPTCY Domain

In addition, the expectations that scripts provide play a crucial role for DD, namely in the detection of omission, in two complementary ways.

The more obvious one is the need for an expectation of what information is to be found in a text in order to be able to infer gaps. A common attempt at deception in bankruptcy cases, for example, is concealment of pre-bankruptcy conversions of property from creditors, which is a major factor considered by the courts in determining whether there was an intent to hinder, delay or defraud in a bankruptcy. Thus, if a sale of assets by a company prior to its filing bankruptcy is found in a text and there is no mention of how closely to the filing this conversion took place, this needs to raise a flag that possibly concealment took place. This can be established since CONCEALMENT is defined as part of the script BANKRUPTCY, which is instantiated for the TMR of the text. If it can be established, from the text itself or the FR, that the sale of the assets took place while the company was approaching the state of bankruptcy, the omission of the specific time of sale in the report constitutes deception. Here, the script facilitates the targeting of SF (see previous section) by mapping where omissions in the text point to the omission of crucial information.

The second mechanism by which scripts facilitate DD is when an event that occurs commonly or exclusively as a subevent of a script, which is otherwise not mentioned, is found in a text. Here, the inference should be that the larger context of this subevent, captured by the script, is to be concealed. If, for example, a company issues a report that mentions the layoff of some of its employees, this should lead to the inference that it approaches the state of bankruptcy, for which layoffs are a possible subevent.

Simplified to a few subevents, these two DD mechanisms on the basis of scripts can be summarized as follows (cf. figure 8): 1. If a necessary element of a script is missing it is likely to be intentionally omitted. 2. If an element that commonly occurs as part of a script is found in a text, but no other element of it, that is, the script is underinstantiated, the script is likely to be intentionally omitted.

```
SCRIPT
    has-part
        AND
            event-1     found in text
            event-2     found in text
            event-3     not found in text
            event-4     found in text

SCRIPT
    has-part
        AND
            event-1     not found in text
            event-2     not found in text
            event-3     found in text
            event-4     not found in text
```

Figure 8: Simplified Script Structures

8 Conclusion

The main thrust of the paper has been not so much the establishment of a sexy application as to demonstrate that the rich resources of NLP, in general, and ONSE, in particular, are versatile enough to be extended to interesting new uses and that getting there involves theoretical and methodological developments that are generally good for the field rather than just for SF (e.g., who will refuse a microtheory of euphemisms?). Throughout, we have insinuated, ever so subtly, that the tasks in hand are not manageable by any of the past or current meaning-avoiding, non-representational approaches. This is not to say that a good SF NLP system must be statistics-free: Crude measures are good to have for heuristic and other startup purposes—but it is TMR elements that such statistics will be counting. We have left out many aspects of SF, such as potential demand, which is great, and other practical considerations. As resources permit, we have been moving consistently to enrich our ONSE resources with IAS capabilities and functionalities, and SF is the latest but, very probably, not the last of those.

9 Acknowledgments

We are grateful to the two anonymous referees for their thought-provoking remarks. We appreciate Professor Jon Patrick's unsolicited comments as well as his making two hard-to-access papers from his research group available to us. We have profited greatly from the cooperation with our CERIAS colleagues and from the Center's support for the NL IAS endeavors over the last 5 years.

10 References

Atallah, M. J., V. Raskin, M. Crogan, C. F. Hempelmann, F. Kerschbaum, D. Mohamed, and S. Naik. 2001. Natural Language Watermarking: Design, Analysis, and a Proof-of-Concept Implementation. In: I. S. Moskowitz (ed.), *Information Hiding: 4th International Workshop, IH 2001, Pittsburgh, PA, USA, April 2001 Proceedings*. Berlin: Springer, 185-199.

Atallah, M. J., V. Raskin, C. F. Hempelmann, M. Karahan, R. Sion, U. Topkara, and K. E. Triezenberg. 2002. Natural Language Watermarking and Tamperproofing. In: F. A. P. Petitcolas (ed.), *Information Hiding: 5th International Workshop, IH 2002, Proceedings*. Berlin: Springer, 196-210.

Barwise, J., and J. Perry. 1983. *Situations and Attitudes*. Cambridge, MA: MIT Press.

Bennett, K. 2003. *Semantic mimicking*. CERIAS TR (www.cerias.purdue.edu), Purdue University, W. Lafayette, IN.

Herke-Couchman, M., C. Whitelaw, and J. Patrick 2003. Identifying interpersonal features using systemic features, AAAI Symposium.

Gibbons, J. 2003. *Forensic Linguistics: An Introduction to Language in the Justice System*. Oxford: Blackwell.

McMenamin, G. R. 2002. *Forensic Linguistics: Advances in Forensic Stylistics*. Boca Raton, LA: CRC Press.

Mohamed, D. 2001. *Ontological Semantics Methods for Automatic Downgrading*. An unpublished Masters' thesis, Program in Linguistics and CERIAS, Purdue University.

Nirenburg, S. and V. Raskin. 2004. *Ontological Semantics*. Cambridge, MA: MIT Press (forthcoming).

Olsson, J. 2004. *Forensic Linguistics: An Introduction to Language, Crime and the Law*. London-New York: Continuum.

Palmer, G. 2002. Forensic analysis in a digital world. *International Journal of Digital Evidence,* Spring 2002, 1.

Raskin, V. 1987. The semantics of lying. In: R. Crespo, B. D. Smith, and H. Schultinik (eds.), *Aspects of Language: Studies in Honour of Mario Alinei, Vol. II. Theoretical and Applied Semantics.* Amsterdam: Rodopi, 443-469.

Raskin, V. 2004. *Natural Language Information Assurance and Security*. Tutorial, COLING 2004, Geneva, Switzerland. August 22.

Raskin, V., M. J. Atallah, C. J. McDonough, and S. Nirenburg. 2001a. Natural Language Processing for Information Assurance and Security: An Overview and Implementations. In: M. Schaefer (ed.), *Proceedings. New Security Paradigm Workshop. September 18th-22nd, 2000, Ballycotton, County Cork Ireland.* New York: ACM Press, 51-65.

Raskin, V., M. J. Atallah, C. F. Hempelmann, and D. Mohamed. 2001b. *Hybrid Data and Text System for Downgrading Sensitive Documents*. CERIAS TR.

Raskin, V., C. F. Hempelmann, K. E. Triezenberg, and S. Nirenburg. 2002a. Ontology in information security: A useful theoretical foundation and methodological tool. In: V. Raskin & C. F. Hempelmann (eds.), *Proceedings. New Security Paradigms Workshop 2001. September 10th-13th, Cloudcroft, NM, USA,* New York: ACM Press, 53-59.

Raskin, V., S. Nirenburg, M. J. Atallah, C. F. Hempelmann, and K. E. Triezenberg. 2002b. Why NLP should move into IAS. In: Steven Krauwer (ed.), *Proceedings of the Workshop on a Roadmap for Computational Linguistics,* Taipei, Taiwan: Academia Sinica, 2002, 1-7.

Raskin, V., S. Nirenburg, C. F. Hempelmann, I. Nirenburg, K. E. Triezenberg. 2003. The Genesis of a Script for Bankruptcy in Ontological Semantics. Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning. Available at: http://acl.ldc.upenn.edu/W/W03/W03-0905.pdf

Rogers, M. and K. Seigfried. 2004. The future of computer forensics: A needs analysis survey. *Computers and Security*, 23, 1, 12-16.

Saferstein, R. 2004. *Criminalistics: An introduction to forensic science*. New York: Prentice Hall.

Whitcomb, C. 2002. A historical perspective of digital evidence: A forensic scientist's view. *International Journal of Digital Evidence,* Spring 2002, 1

Whitelaw, C., and Sh. Argamon 2004. Systemic functional features in stylistic text classification. Ms., Sydney Language Technology Research Group, University of Sydney, Sydney, Australia.