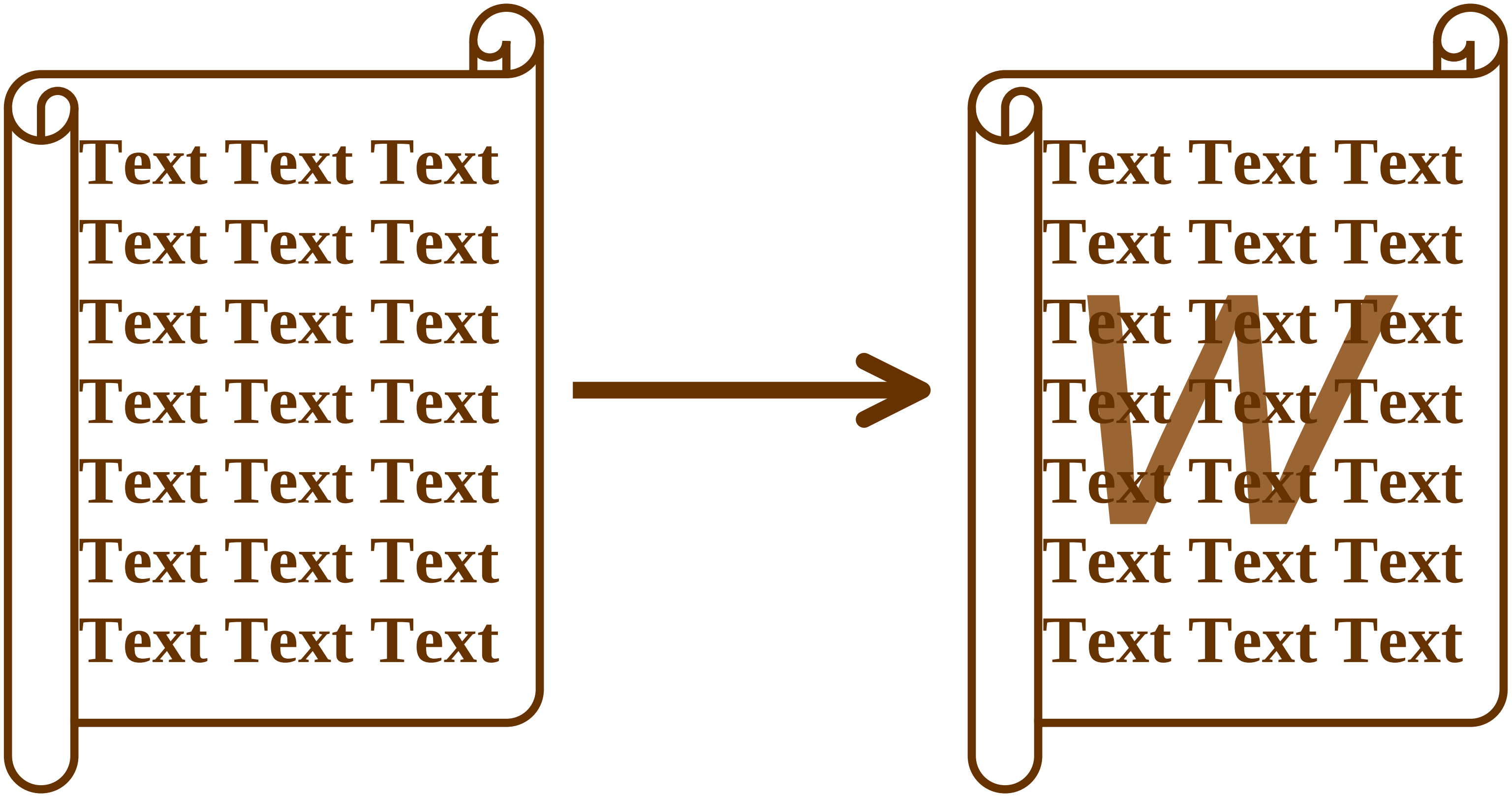
Natural Language Watermarking

Mikhail J. Atallah, Victor Raskin
with
Michael Crogan, Christian Hempelmann, Florian Kerschbaum, Dina Mohamed, Sanket Naik

CERIAS
Center for Education and Research
in Information Assurance and Security

The Problem


Text Text Text
Text Text Text
Text Text Text
Text Text Text
Text Text Text
Text Text Text
Text Text Text
Text Text Text
Text Text Text
Text Text Text
Text Text Text
Text Text Text
Text Text Text
Text Text Text
W

Properties of Watermarked Text

- Same meaning as original text
- Holds up in court
- Watermark is only readable with key
- Original document need not be stored
- Removing watermark is hard
- Watermarking algorithm is public
- Ability to produce multiple watermarks

Model of Adversary

- Meaning-preserving transformations (e.g., translation into another language)
- Meaning-modifying transformations (destroys original)
- Insert new sentences
- Move sentences, paragraphs, chapters, etc.

Previous Schemes

- Spacing between letters, words or lines [BraMaxOGo94] [Max94] [LowMaxLap98]
- Manipulating the format, e.g. HTML, TeX, Postscript, etc. [Kat00]
- Generating cover text [Way95] [ChaDav97]
- Synonym substitution [AtaMcDRasNir00]
- Spelling errors, punctuation, etc.

Advantages of Our Scheme

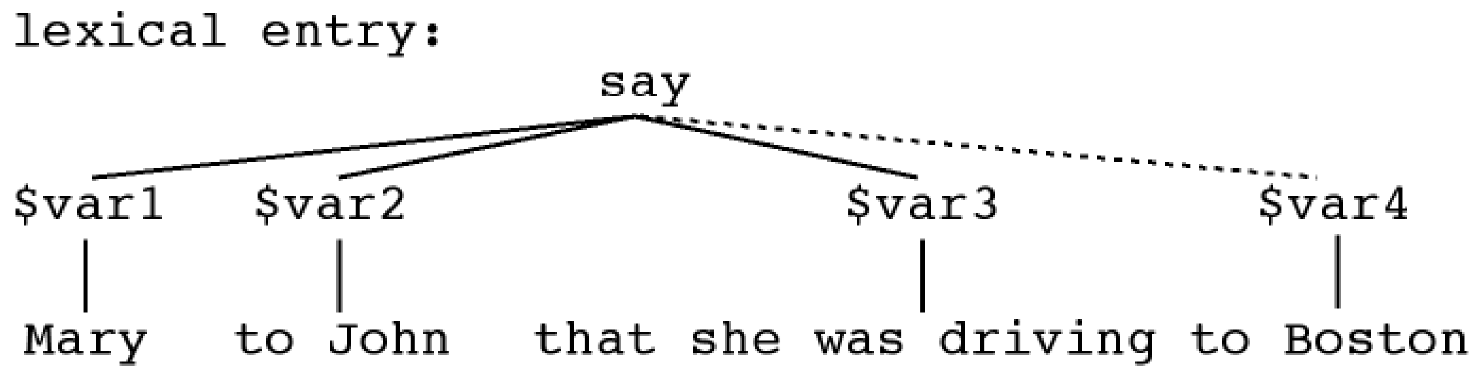
- Embeds watermark in given text
- Resistant to changes in representation, e.g. OCR
- Requires crippling meaning changes to destroy the watermark

Why NLP?

- Automatic text processing
- Best method is meaning-based
- Semantic analysis is “unhiding” information
- Best-developed semantic method is ontological semantics

Ontological Semantics

- Ontology: hierarchy of conceptual nodes
- Lexicon: entries explained in terms of nodes
- Necessary modules:
  Syntactic Parser, Analyzer, Generator

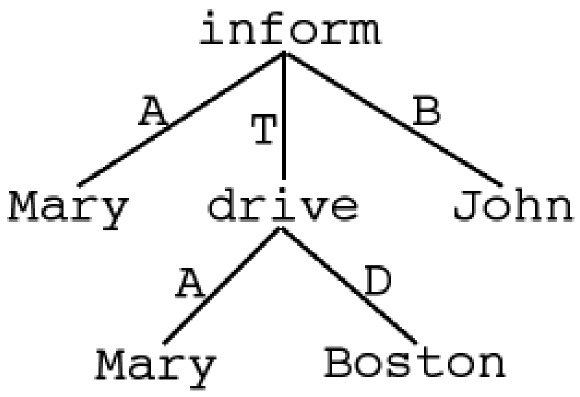


Ontological Semantics - contd.

- Basis for analysis into Text-meaning-representation (TMR):

```
ontological concept:

(inform
   (Agent ^$var1)
   (Theme ^$var3
       (Agent ^$var1)
       (Destination ^$var4))
   (Beneficiary ^$var2))
```



Syntactic Representation

- LINK Parser produces constituent trees

(S (NP$_1$ the dog)
    (VP chased
        (NP$_2$ the cat)))

- Easily translatable into bit strings
- Transparent interface for transformations

Examples of Syntactic Transformations

- Passivization
  - Transitive verb
  - Switch object NP to subject position
  - Make subject PP adjunct
  - Adjust verbal morphology

```
(S (NP_2 the cat)
   (VP was
     (VP chased
        (PP by
            (NP_1 the dog)))))
```

Examples of Syntactic Transformations - continued

- Adjunct movement:
  `(often) the dog (often) chased the cat (often)`
- Clefting (e.g., of mandatory subject):
  `it was the dog that chased the cat`
- Adjunct insertion:
  `it seems that / generally speaking / basically the dog chased the cat`
- Combinations of all these:
  `it seems that it was the cat that was often chased by the dog`

Watermarking Algorithm Overview

- Split text into sentences $s_1,\ldots,s_n$
- Find tree representation $T_1,\ldots,T_n$ of each sentence
- Map each tree into a bit string $B_1,\ldots,B_n$ according to a secret prime p
- Choose subset $t_1,\ldots,t_\alpha$ of sentences according to secret prime p
- Transform subset, such that β bits of each $B_{t_1},\ldots B_{t_\alpha}$ corresponds to the watermark W

Mapping of Trees to Bit Strings

- Assign each node of *T* a number *i* according to pre-order traversal
- Replace every number *i* with a bit:
  1 if *i* + *H(p)* is a quadratic residue modulo *p*,
  else with *0*
- List the bits in post-order traversal

Choosing Watermark Sentences

- Compute $B'_i = H(B_i)$ for each sentences
- Bitwise B'_i compare to $H(p)$
- Rank sentences decreasingly to number of matches
- Choose α top ranked sentences as markers

Watermark Insertion

- Use sentences after markers for watermark
- Apply transformations until bit strings match watermark
- If all transformations fail ask user for manual changes or insert new sentences

Probabilities

- Meaning-preserving transformation: *0*
- Meaning-modifying transformation: $\leq 3\alpha/n$
- Insertion of a sentence: $\leq 2\alpha/n$
- Moving a block of sentences: $\leq 3\alpha/n$
- All of the above are upper bounds !

Properties

- Watermark extraction with key only
- Probability of false positives 2^{-w}
- On average $2^{\beta-1}$ transformations are performed
- Different watermarks for the same cover text will reveal watermark placement (random modifications of a number of sentences will make this more difficult)

Current Prototype

- Proof-of-Concept implementation
- Uses syntax instead of TMR
- Uses Link parser from CMU
- Uses limited set of transformations

Example

"The functions of these instruments are discussed in the Appendix."

"In the Appendix the functions of these instruments are discussed."

"In the Appendix the functions of these tools are discussed."

Experiences with Prototype

- http://www.cerias.purdue.edu/homes/wmnlt/
- One “watermark-bit” per sentence: $\beta = 1$
- Number of transformations: *3*
- Approximate bandwidth: *n/3*

Planned Extensions

- Build TMR-based version
- Adjust ontology, parser, analyzer and generator
- Increase number of transformations
- Sentence insertion