Natural Language Watermarking

Foundations of individual Text-Watermarking
Overview

- Motivation
- Introduction: Natural Language Watermarking
- Examples of embedding methods
- Challenges in NLW
- References
What is NLW?

• Imagine the following scenario:

„Two prisoners (Alice and Bob) want to communicate and the only possibility they have, is sending eachother plaintext messages, which are delivered by the warden.

The prisoners cannot use encryption because the warden wants to read all outgoing messages, otherwise he will not deliver the message...“

This scenario is called:
„The Prisoners' Problem and the Subliminal Channel“
What is NLW?

Bob

Warden

Alice

We'll make the great escape at midnight!

Let me sing you a love song about what I feel in my heart... Butterflies can't find...

secret message

cover object

Note: the cover object remains the same!
What is NLW?

The Prisoners' Problem is the fundamental idea behind “Information Hiding”

And “Information Hiding” is the superordinate concept of NLW...
What is NLW?

NLW = “embedding of information by modifying original data in a discreet manner, such that the modifications are imperceptible when watermarked data is consumed…”

• The information to embed is called “watermark-message” which can be e.g.
  • Serial / ID number of the document
  • Customer licence data (Purchase date, IP Address,...)
  • Meaningful text (short messages)
  • etc.
Some characteristics of NLW

• Robust / Fragility
  - Difficult to remove for an attacker, who would like to destroy it
  - Should be resistant against simple format conversions, that occur often in a workaday life, e.g. HTML → TXT, DOC → PDF, XLS → XML, etc.
  - Removal should cause degradation in the quality of the data

• Readily Detectable
  - Data owner should easily detect it

• Unobtrusive
  - Invisible enough not to degrade data quality and prevent attacker from finding and deleting it
The link between NLP and NLW

• How does Natural Language Processing (NLP) relate to Natural Language Watermarking (NLW)?

• NLW combines many tools and methods provided by NLP in order to embed watermarks at a “natural language” level, e.g.:

  ✓ (Intelligent) Tokenization / Pattern Matching
  ✓ POS-Tagging
  ✓ Parsing
  ✓ Text simplification / text paraphrasing
  ✓ Word sense disambiguation
  ✓ Anaphora resolution
  ✓ ...and many more
Ownership Assertion

• Alice generates watermark and embeds it into her document
• Alice makes watermarked document publicly available
• Bob downloads document and claims he’s the owner of copyrighted content
• Alice produces the unmarked original and establishes the presence of her embedded watermark → Result: Bob is not the owner!

Fingerprinting

• Used to avoid unauthorized duplication / distribution
• A distinct watermark (fingerprint) is embedded in each copy of the data
• If unauthorized copies are found, the original copy can be traced back by retrieving the fingerprint
Content Protection

• Content owner wants to publish his content (for free), but in a “read-only” manner

• Content owner wants to publish individual copies of his content, (that means each copy is marked in a “unique” manner)

Content labeling

• Bits embedded in the data, comprise an annotation, giving some more information about the data
Embedding process

• Embedding a watermark-message into natural language text requires a systematic method for transforming (modifying) text...

• **Claim:** Transformations should *preserve grammaticality* of sentences

• **Ideally:** *differences* in sentence meaning caused by transformations should *not be noticeable*

• Generally there are three types of transformations:
  ➢ Synonym substitution
  ➢ Syntactic transformations
  ➢ Semantic transformations
Embedding process: Notation

• Following symbols will be used as abbreviations:

\( \mathcal{T} \)  The original document

\( \mathcal{T}' \)  The watermarked document

\( \mathcal{W} \)  The watermark message, e.g. “10001” \( \rightarrow \) formally: \( \mathcal{W} \in \bigcup_{i \in \mathbb{N}} \{0, 1\}^i \)

\( \mathcal{W}_{bit} \)  One single watermark bit, e.g.

\( \mathcal{T} =_{sem} \mathcal{T}' \)  Semantic equality (both documents have the same sense)

\( \mathcal{T} =_{syn} \mathcal{T}' \)  Syntactical equality (both documents have „somehow“ same syntactical structure)
Embedding process: Steps

1) Annotate $\mathcal{T}$ with POS-Tags for each word

2) Apply regular expressions to extract patterns to match the rules of the embedding methods

3) Apply methods to embedd $\mathcal{W}$ into $\mathcal{T}$

4) If a method $\mathbf{A}$ couldn’t embedd $\mathcal{W}$ by itself, some other methods $\mathbf{B}, \mathbf{C}, \mathbf{D}, \ldots$ will try to help (collaborative NLW)

5) During the process of applying transformations, embedding-methods must “ensure” that the sense of $\mathcal{T}$ is preserved

Note: A pattern equals to one $\mathcal{W}_{bit}$

To handle this some methods call a semantic query module, which tries to lookup in a giant corpus for phrases that are somehow similar to the transformations
Embedding process: Conclusion

• What do we know so far?

• We know **what** Natural Language Watermarking is
• We know for **which purpose** we use this disciplin
• We know (some) **characteristics** of NLW
• We know (a part) of the commonly used **notation**

• Guess what’s missing?
Embedding process: Conclusion

• What do we know so far?
  
• We know **what** Natural Language Watermarking is
• We know for **which purpose** we use this disciplin
• We know (some) **characteristics** of NLW
• We know (a part) of the commonly used **notation**

• Guess what’s missing?

The **methods** to embed the watermarks (“the core”)
Embedding methods: Overview

- Conjunction Modulation
- Enumeration Modulation
- Center Permutation
- Hyponym & Troponym Truncation
- Synonym Substitution
- Hyphen Split
- Subordinate Clause Swap
- Prefix Elimination

Note: since the implemented framework was focused on a local language, the following examples are in German...

Focus in this presentation...
Embedding methods: Overview

• Conjunction Modulation

Idea: Swap open-class words, which are connected by conjunctions...

\[ T_1 = \ldots \text{bei den Wahlmännern und Wahlfrauen der Union...} \]
\[ T_2 = \ldots \text{hätte die Linke Gauck oder Wulff gewählt...} \]
\[ T_3 = \ldots \text{es war weder gelb noch grün...} \]

\[ T'_1 = \ldots \text{bei den Wahlfrauen und Wahlmännern der Union...} \]
\[ T'_2 = \ldots \text{hätte die Linke Wulff oder Gauck gewählt...} \]
\[ T'_3 = \ldots \text{es war weder grün noch gelb...} \]
Embedding methods: Overview

• Conjunction Modulation: What’s behind it?
**Embedding methods: Overview**

- **Conjunction Modulation:** What’s behind it?

<table>
<thead>
<tr>
<th>lassen</th>
<th>sich</th>
<th>Manipulationen</th>
<th>und</th>
<th>Systemausfälle</th>
<th>verhindern</th>
</tr>
</thead>
<tbody>
<tr>
<td>VVFIN</td>
<td>PRF</td>
<td>NN</td>
<td>KON</td>
<td>NN</td>
<td>VVINF</td>
</tr>
</tbody>
</table>

- Both nouns (NN) are connected through the “und” conjunction (KON)

- They’re both independent from all the other words in the sentence, (more formally, they are *constituents* !)

- **Grammar rule:** “*Only for constituents it is permitted to be swapped closed inside a sentence...*”

- How does a machine recognizes constituents? → Chunker (e.g. Stanford Parser)
Embedding methods: Overview

• Enumeration Modulation

**Idea:** slightly different than the Conjunction Modulation method
Swap open-class words, which are separated by comma...

\[ T_1 = "...Fertigprodukte wie Pizza, Cornflakes oder Limonade..." \]

\[ T_2 = "...der teuren, serviceaufwändigen und kunden-unfreundlichen Technik..." \]

\[ T_1' = "...Fertigprodukte wie Cornflakes, Pizza oder Limonade..." \]

\[ T_2' = "...der serviceaufwändigen, teuren und kunden-unfreundlichen Technik..." \]
Embedding methods: Overview

• Center Permutation

Idea: Swap independent constituents in the middle field of a sentence...

\[ T_1 = \ldots \text{Opa Rainer hat} \text{seinem Enkel das Buch gestern geschenkt...} \]

\[ T'_1 = \ldots \text{Opa Rainer hat} \text{das Buch seinem Enkel gestern geschenkt...} \]

= \ldots \text{Opa Rainer hat} \text{gestern das Buch seinem Enkel geschenkt...} \]

= \ldots \text{Opa Rainer hat} \text{gestern seinem Enkel das Buch geschenkt...} \]
• **Center Permutation:** How does it work?

- The three constituents are all *independent* from each other (and from the rest of the sentence).
- The *grammar rule:* “*only for constituents it is permitted*...” applies also here!
• **Center Permutation:** How does it work?

- In case that there is a temporal constituent in the middle field (temporal adverb) it is also allowed to declare the constituent as the “new prefield”, such that the “old prefield” is moved directly after the finite verb (“hat”)}
Embedding methods: Overview

• Hyponym & Troponym Truncation

**Idea:** Replace noun/verb by it’s direct Hyponym...

\[ T_1 = "...gestern wurde unser Nachbar von einem Bullterrier gebissen..." \]

\[ T_2 = "...Zum Öffnen des Gehäuses wird ein Kreuzschraubenzieher benötigt..." \]

\[ T'_1 = "...gestern wurde unser Nachbar von einem Terrier gebissen..." \]

\[ T'_2 = "...Zum Öffnen des Gehäuses wird ein Schraubenzieher benötigt..." \]
Embedding methods: Overview

- Hyponym & Troponym Truncation

  How does it work?

- Pickup a word from a sentence, e.g. “Perserkatze”

- Build a hyponym-chain (at least 3 hyponyms)
• Hyponym & Troponym Truncation

How does it work?

• Pickup a word from a sentence, e.g. “Perserkatze”

• Build a hyponym-chain (at least 3 hyponyms)
Embedding methods: Overview

- Hyponym & Troponym Truncation

  How does it work?

- Pickup a word from a sentence, e.g. “Perserkatze”

- Build a hyponym-chain (at least 3 hyponyms)

- Truncate chain at the hyponym next to last and replace old word by it’s direct hyponym...

```
Perserkatze
Katze
Raubtier
Säugetier
Wirbeltier
Tier
Lebewesen
```

```
T ≈ T'
```
• **Hyponym & Troponym Truncation**

How does it work?

• Pickup a word from a sentence, e.g. “Perserkatze”

• Build a hyponym-chain (at least 3 hyponyms)

• Truncate chain at the hyponym next to last and replace old word by it’s direct hyponym...

• May loose information on details level... BUT sense will always remain the same!

• **Note:** good results, if hyponym appears as a substring of the old word (**Katze** / **Perserkatze**)
Embedding methods: Overview

- **Synonym Substitution**

  **Idea:** Replace a word (mostly adverbs/adjectives) with a similar synonym...

  \[ T_1 = "...könnte nach BP-Angaben deutlich mehr Öl austreten, als *bislang* angenommen..." \]

  \[ T_2 = "...im zweiten Wahlgang ist für Komorowski daher *optimistischer* als..." \]

  \[ T'_1 = "...könnte nach BP-Angaben deutlich mehr Öl austreten, als *bisher* angenommen..." \]

  \[ T'_2 = "...im zweiten Wahlgang ist für Komorowski daher *zuversichtlicher* als..." \]
**Embedding methods: Overview**

- **Synonym Substitution:** How does it work?

  Before words can be replaced, it’s necessary to have a synonym database

<table>
<thead>
<tr>
<th>einig</th>
<th>zugehörig</th>
<th>angegliedert</th>
<th>verbunden</th>
</tr>
</thead>
<tbody>
<tr>
<td>wüstig</td>
<td>dick</td>
<td>bauschig</td>
<td>gebauscht</td>
</tr>
<tr>
<td>nervös</td>
<td>gespannt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>beschäftigt</td>
<td>tätig</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nahe</td>
<td>anliegend</td>
<td>daneben</td>
<td>bei</td>
</tr>
<tr>
<td>streitlustig</td>
<td>offensiv</td>
<td>aggressiv</td>
<td></td>
</tr>
<tr>
<td>mutlos</td>
<td>feige</td>
<td>schüchtern</td>
<td></td>
</tr>
<tr>
<td>gemütlich</td>
<td>lauschig</td>
<td></td>
<td></td>
</tr>
<tr>
<td>rätselhaft</td>
<td>enigmatisch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>aus Anlass</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>anbei</td>
<td>beiliegend</td>
<td>beigefügt</td>
<td>beigelegt</td>
</tr>
<tr>
<td>breitspurig</td>
<td>hochmütig</td>
<td>vermessen</td>
<td>stolz</td>
</tr>
<tr>
<td>so gut wie</td>
<td>gerade noch</td>
<td>beinahe</td>
<td>nahezu</td>
</tr>
<tr>
<td>in Ordnung</td>
<td>ganz recht</td>
<td>gut so</td>
<td></td>
</tr>
<tr>
<td>alljährlich</td>
<td>per annum</td>
<td>jährlich</td>
<td>pro Jahr</td>
</tr>
<tr>
<td>auf ein Jahr gerechnet</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>namenlos</td>
<td>unennbar</td>
<td>ungenannt</td>
<td></td>
</tr>
<tr>
<td>augenscheinlich</td>
<td>scheinbar</td>
<td></td>
<td></td>
</tr>
<tr>
<td>danach</td>
<td>nachfolgend</td>
<td>anknüpfend</td>
<td>nachkommend</td>
</tr>
<tr>
<td>motivierend</td>
<td>anmütigend</td>
<td>aufputschend</td>
<td>antreibend</td>
</tr>
<tr>
<td>sicherlich</td>
<td>gern</td>
<td>gerne</td>
<td>freudig</td>
</tr>
</tbody>
</table>

Database can be either **local** or **external** (e.g. Uni-Leipzig knowledge base)
• **Synonym Substitution**: How does it work?

• Once a synonym is chosen, it must be sure that it's somehow replaceable with the original word (preserve meaning...)

• Idea: Lookup in a concordancer for occurrence of original and marked word, use a context-window with a minimum of two surrounding neighbours...

...als *bislang* angenommen...

...als *bisher* angenommen...

...als *seither* angenommen...

...als *früher* angenommen...

If a specific threshold can be reached, the word can be replaced by the synonym with the highest frequency...
Embedding methods: Overview

• Synonym Substitution: *alternative approach*...

  No matter which tricks you try to apply – this method is a challenge!

  But there are quite other alternative approaches, e.g. Phrase Substitution

\[ T_1 = "Ich begreife nicht was Sie meinen ..." \]

\[ T'_1 = "Ich kann Ihnen nicht folgen..." \]

Problem: requires (several) phrase corpora (e.g. Wiki-Phrases)
Challenges in NLW

• Nor of the presented methods is 100% “bulletproof” 😞

• Most expensive part of NLW is: evaluation of the result...

Only humans are able to judge if a transformation is 100% justified!

• Embedding usefull watermarking-messages (> 32bit) requires larger texts, ≈ 10KBytes is a quite good start...

• Embedding methods usually work together, but sometimes they can block eachother, such that the whole embedding process fails...
Challenges in NLW

- Example

\[ T \sim T' \]
Questions?
Thanks for your attention...
References

- “Foundations of individual Text-Watermarking”,

- “The Prisoners' Problem and the Subliminal Channel”,