# Translation-Based Steganography

Christian Grothoff Krista Grothoff
Ludmila Alkhutova Ryan Stutsman Mikhail Atallah

CERIAS, Purdue University `{christian,krista}@grothoff.org`,
`{lalkhuto,rstutsma}@purdue.edu,mja@cs.purdue.edu`

**Abstract.** This paper investigates the possibilities of steganographically embedding information in the "noise" created by automatic translation of natural language documents. Because the inherent redundancy of natural language creates plenty of room for variation in translation, machine translation is ideal for steganographic applications. Also, because there are frequent errors in legitimate automatic text translations, additional errors inserted by an information hiding mechanism are plausibly undetectable and would appear to be part of the normal noise associated with translation. Significantly, it should be extremely difficult for an adversary to determine if inaccuracies in the translation are caused by the use of steganography or by deficiencies of the translation software.

## 1 Introduction

This paper presents a new protocol for covert message transfer in natural language text, for which we have a proof-of-concept implementation. The key idea is to hide information in the noise that occurs invariably in natural language translation. When translating a non-trivial text between a pair of natural languages, there are typically many possible translations. Selecting one of these translations can be used to encode information. In order for an adversary to detect the hidden message transfer, the adversary would have to show that the generated translation containing the hidden message could not be plausibly generated by ordinary translation. Because natural language translation is particularly noisy, this is inherently difficult. For example, the existence of synonyms frequently allows for multiple correct translations of the same text. The possibility of erroneous translations increases the number of plausible variations and thus the opportunities for hiding information.

This paper evaluates the potential of covert message transfer in natural language translation that uses automatic machine translation (MT). In order to characterize which variations in machine translations are plausible, we have looked into the different kinds of errors that are generated by various MT systems. Some of the variations that were observed in the machine translations are also clearly plausible for manual translations by humans.

In addition to making it difficult for the adversary to detect the presence of a hidden message, translation-based steganography is also easier to use. The reason for this is that unlike previous text-, image- or sound-based steganographic systems, the substrate does not have to be secret. In translation-based steganography, the original text in the source language can be publically known, obtained from public sources, and, together with the translation, exchanged between the two parties in plain sight of the adversary. In traditional image steganography, the problem often occurs that the source image in which the message is subsequently hidden must be kept secret by the sender and used only once (as otherwise a “diff” attack would reveal the presence of a hidden message). This burdens the user with creating a new, secret substrate for each message.

Translation-based steganography does not suffer from this drawback, since the adversary cannot apply a differential analysis to a translation to detect the hidden message. The adversary may produce a translation of the original message, but the translation is likely to differ regardless of the use of steganography, making the differential analysis useless for detecting a hidden message.

To demonstrate this, we have implemented a steganographic encoder and decoder. The system hides messages by changing machine translations in ways that are similar to the variations and errors that were observed in the existing MT systems. An interactive version of the prototype is available on our webpage.[1]

The remainder of the paper is structured as follows. First, Section 2 reviews related work. In Section 3, the basic protocol of the steganographic exchange is described. In Section 4, we give a characterization of errors produced in existing machine translation systems. The implementation and some experimental results are sketched in Section 5. In Section 6, we discuss variations on the basic protocol, together with various attacks and possible defenses.

## 2 Related Work

The goal of both steganography and watermarking is to embed information into a digital object, also referred to as the substrate, in such a manner that the information becomes part of the object. It is understood that the embedding process should not significantly degrade the quality of the substrate. Steganographic and watermarking schemes are categorized by the type of data that the substrate belongs to, such as text, images or sound.

### 2.1 Steganography

In steganography, the very existence of the message must not be detectable. A successful attack consists of detecting the existence of the hidden message, even without removing it (or learning what it is). This can be done through, for example, sophisticated statistical analyses and comparisons of objects with and without hidden information.

Traditional linguistic steganography has used limited syntactically-correct text generation [29] (sometimes with the addition of so-called “style templates”) and semantically-equivalent word substitutions within an existing plaintext as a

---

[1] http://www.cs.purdue.edu/homes/rstutsma/stego/

medium in which to hide messages. Wayner [29,30] introduced the notion of using precomputed context-free grammars as a method of generating steganographic text without sacrificing syntactic and semantic correctness. Note that semantic correctness is only guaranteed if the manually constructed grammar enforces the production of semantically cohesive text. Chapman and Davida [6] improved on the simple generation of syntactically correct text by syntactically tagging large corpora of homogeneous data in order to generate grammatical "style templates"; these templates were used to generate text which not only had syntactic and lexical variation, but whose consistent register and "style" could potentially pass a casual reading by a human observer. Chapman et al [7], later developed a technique in which semantically equivalent substitutions were made in known plaintexts in order to encode messages. Semantically-driven information hiding is a relatively recent innovation, as described for watermarking schemes in Atallah et al [4]. Wayner [29,30] detailed text-based approaches that are strictly statistical in nature. However, in general, linguistic approaches to steganography have been relatively limited. Damage to language is relatively easy for a human to detect. It does not take much modification of a text to make it ungrammatical in a native speaker's judgement; furthermore, even syntactically correct texts can violate semantic constraints.

Non-linguistic approaches to steganography have sometimes used lower-order bits in images and sound encodings to hide the data, providing a certain amount of freedom in the encoding in which to hide information [30]. The problem with these approaches is that the information is easily destroyed (the encoding lacks robustness, which is a particular problem for watermarking), that the original data source (for example the original image) must not be disclosed to avoid easy detection, and that a statistical analysis can still often detect the use of steganography (see, e.g., [13,18,20,25,30], to mention a few).

### 2.2 Watermarking

The intended purpose of the watermark largely dictates the design goals for watermarking schemes. The possible uses of watermarking include inserting ownership information, inserting purchaser information, detecting modification, placing caption information and so on. One such decision is whether the watermark should be visible or indiscernible. For example, a copyright mark need not be hidden; in fact, a visible digital watermark can act as a deterrent to an attacker. Most of the literature has focused on indiscernible watermarks.

Watermarks are usually designed to withstand a wide range of attacks that aim at removing or modifying the watermark without significantly damaging the usefulness of the object. A *resilient* watermark is one that is hard to remove by an adversary without damaging the object to an unaceptable extent. However, it is sometimes the case that a *fragile* watermark is desirable, one that is destroyed by even a small alteration; this occurs when watermarking is used for the purpose of making the object tamper-evident (for integrity protection).

The case where the watermark has to be different for each copy of the digital object, is called *fingerprinting*. That is, fingerprinting embeds a unique message in each instance of the digital object (usually the message makes it possible to trace a pirated version back to the original culprit). Fingerprinting is easier to attack because two differently marked copies often make possible an attack that consists of comparing the two differently marked copies (the attacker's goal is then to create a usable copy that has neither one of the two marks).

Although watermarks can be embedded in any digital object, by far most of the published research on watermarking has dealt with media such as images, audio or video. There is some literature on watermarking other object types like software [9,10,11], databases [1,26], and natural language text [3,4].

### 2.3 Machine Translation

Most Machine Translation (MT) systems in use today are statistical MT systems based on models derived from a corpus, transfer systems that are based on linguistic rules for the translations, or hybrid systems that combine the two approaches. Other translation methodologies, such as semantic MT exist, but are not considered further as they are not commonly available at this time.

In statistical MT [2,5], the system is trained using a bilingual parallel corpus to construct a *translation model*. The translation model gives the translator statistical information about likely word alignments. A word alignment [23,24] is a correspondence between words in the source sentence and the target sentence. For example, for English-French translations, the system "learns" that the English word "not" typically corresponds to the two French words "ne pas". The statistical MT systems are also trained with a uni-lingual corpus in the target language to construct a *language model* which is used to estimate what constructions are common in the target language. The translator then performs an approximate search in the space of all possible translations, trying to maximize the likelihood of the translation to score high in both the translation model and the language model. The selection of the training data for the construction of the models is crucial for the quality of the statistical MT system.

## 3 Protocol

The basic steganographic protocol for this paper works as follows. The sender first needs to obtain a substrate text in the source language. The substrate does not have to be secret and can be obtained from public sources - for example, a news website. The sender then translates the sentences in the source text into the target language using the steganographic encoder. The steganographic encoder essentially creates multiple translations for each sentence and selects one of these to encode bits from the hidden message. The translated text is then transmitted to the receiver, together with information that is sufficient to obtain the source text. This can either be the source text itself or a reference to the source. The receiver then also performs the translation of the source text using the same steganographic encoder configuration. By comparing the resulting sentences, the

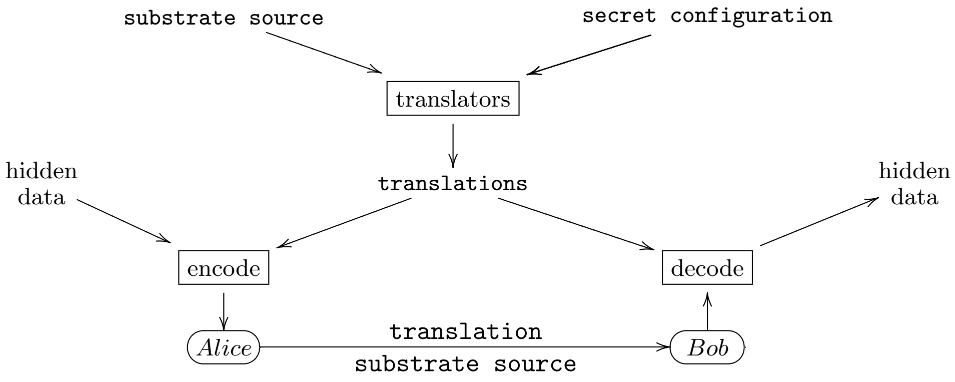

**Fig. 1.** Illustration of the basic protocol. The adversary can observe the public news and the message between Alice and Bob containing the selected translation and the (possibly public) substrate source.

receiver reconstructs the bitstream of the hidden message. Figure 1 illustrates the basic protocol.

The adversary is assumed to know about the existence of this basic protocol and is also able to obtain the source text and to perform translations. It is not practical for the adversary to flag all seemingly machine-translated messages which do not correspond exactly to translations generated from the cover source by well-known MT systems. There are two reasons for this. First, there are too many variants of MT software out there (frequently produced by "tweaking" existing ones), many of which are not advertised or made public. Second, even if there was a single universal MT software copy that everyone uses, there are still wildly differing behaviors for it depending on the corpus on which it is trained – there are too many such potential corpora to track, especially as users seek better translation quality by using a corpus particularly suited to their application domain (e.g., news stories about home construction costs and markets).

The adversary does not have access to the specific configuration of the steganographic encoder (which acts like a secret key). This configuration consists of everything that determines which translations are generated, such as the specific translation algorithms, the corpora used to train any user-generated translation systems which may be employed, rules, and dictionaries. It is assumed that the secret is transmitted using standard secret-sharing protocols and the specifics are not covered here. However, it should be noted that the size of the secret that is transmitted is flexible, based upon the user's choices; users can choose to simply share information about the settings of the encoder, or might choose to transmit entire corpora used to train a user-generated MT system. This varies based upon individual users' needs.

As with most steganographic systems, the hidden message itself can be encrypted with a secret key, making it harder for the adversary to perform guessing attacks on the secret configuration (as configurations of the steganographic system result in a random bitstream for the hidden message).

### 3.1 Producing translations

The first step for both sender and receiver after obtaining the source text is to produce multiple translations of the source text using the same algorithm. The goal of this step is to deterministically produce multiple different translations of the source text. The simplest approach to achieve this is to apply (a subset of) all available MT systems on each sentence in the source text. If the parties have full access to the code of a statistical MT system, they can generate multiple MT systems from the same codebase by training it with different corpora.

In addition to generating different sentences using multiple translation systems it is also possible to apply post-processing on the resulting translations to obtain additional variations. Such post-processing includes transformations that mimic the noise inherent in any (MT) translation. For example, post-processors could insert common translation mistakes (as discussed in Section 4).

As translation quality differs between different engines and also depends on which post-processors were applied to manipulate the result, the translation system uses a heuristic to assign a probability to each translation that describes its relative quality compared to the other translations. The heuristic can be based on both experience with the generators and algorithms that rank sentence quality based on language models [8]. The specific set of translation engines, training corpora and post-processing operations that are used to generate the translations and their ranking are part of the secret shared by the two parties that want to carry out the covert communication.

### 3.2 Selecting a translation

When selecting a translation to encode the hidden message, the encoder first builds a Huffman tree [17] of the available translations using the probabilities assigned by the generator algorithm. Then the algorithm selects the sentence that corresponds to the bit-sequence that is to be encoded.[2]

Using a Huffman tree to select sentences in accordance with their translation quality estimate ensures that sentences that are assumed to have a low translation quality are selected less often. Furthermore, the lower the quality of the selected translation, the higher the number of transmitted bits.

This reduces the total amount of substrate text required and thus the amount of text the adversary can analyze. The encoder can use a lower limit on the relative translation quality to eliminate sentences from consideration where the estimated translation quality is below a certain threshold, in which case that threshold becomes part of the shared secret between sender and receiver.

---

[2] Wayner [29,30] uses Huffman trees in a similar manner to generate statistically plausible substrate texts on a letter-by-letter basis.

### 3.3 Keeping the source text secret

The presented scheme can be adapted to be suitable for watermarking where it would be desirable to keep the source text secret. This can be achieved as follows. The encoder computes a (cryptographic) hash of each translated sentence. It then selects a sentence such that the last bit of the hash of the translated sentence corresponds to the next bit in the hidden message that is to be transmitted. The decoder then just computes the hash codes of the received sentences and concatenates the respective lowest bits to obtain the hidden message.

This scheme assumes that sentences are long enough to almost always have enough variation to obtain a hash with the desired lowest bit. Error-correcting codes must be used to correct errors whenever none of the sentences produces an acceptable hash code. Using this variation reduces the bitrate that can be achieved by the encoding. More details on this can be found in Section 6.

## 4 Lost in Translation

Modern MT systems produce a number of common errors in translations. This section characterizes some of these errors. While the errors we describe are not a comprehensive list of possible errors, they are representative of the types of errors we commonly observed in our sample translations. Most of these errors are caused by the reliance on statistical and syntactic text analysis by contemporary MT systems, resulting in a lack of semantic and contextual awareness. This produces an array of error types that we can use to plausibly alter text, generating further marking possibilities.

### 4.1 Functional Words

One class of errors that occurs rather frequently without destroying meaning is that of incorrectly-translated or omitted closed-class words such as articles, pronouns, and prepositions. Because these functional words are often strongly associated with some other word or phrase in the sentence, complex constructions often seem to lead to errors in the translation of such words. Furthermore, different languages handle these words very differently, leading to translation errors when using engines that do not handle these differences.

For example, languages without articles, such as Russian, can produce article-omission errors when translating to a language which has articles, like English: “Behind sledge cheerfully to run” [12].

Even if articles are included, they often have the wrong sense of definiteness (“a” instead of “the”, and vice-versa). Finally, if both languages have articles these articles are sometimes omitted in translations where the constructions become complex enough to make the noun phrase the article is bound to unclear.

Many languages use articles in front of some nouns, but not others. This causes problems when translating from languages that *do* use articles in front of the latter set of nouns. For example, the French sentence “La vie est paralysée.”

translates to “Life is paralyzed.” in English. However, translation engines predictably translate this as “The life is paralyzed.”. “life” in the sense of “life in general” does not take an article in English. This is the same with many mass nouns like “water” and “money”, causing similar errors.

Furthermore, because articles are also used as pronouns in many languages, they are often mistranslated as such. Many of these languages also indicate gender with articles and pronouns, such that if “the armchair” is male, it might be referred to as “he” (in English) at the beginning of the next sentence, instead of “it”. But because no context is kept by todays MT engines, if there is a man being discussed in the previous sentence, he may also become an “it” in the next.

For example, the following two sentences were translated from a German article into English with Systran (The “Avineri” mentioned is a political scientist cited in the article): “Avineri ist nicht nur skeptisch. Er ist gleichzeitig auch optimistisch.” is translated as “Avineri is not only sceptical. It is at the same time also optimistic.” [22,28]. This lack of context makes correctly translating such words difficult.

Prepositions are also notoriously tricky; often, the correct choice of preposition depends entirely on the context of the sentence. For example, “J’habite *à* 100 mètres de lui” in French means “I live 100 meters from him” in English. However, [28] translates this as “I live *with* 100 meters of him”, and [12] translates it as “In live *in* 100 meters of him.” Both use a different translation of “à” (“with/in”) which is entirely inappropriate to the context.

“Il est mort à 92 ans” (“He died at 92 years”) is given by [28,12] as “He died in 92 years”. To say “He waits for me” in German, one generally says “Er wartet auf mich”. [28] chooses to omit the preposition (“auf” entirely, making the sentence incorrect (effectively, “He waits me.”) Similarly, “Bei der Hochzeit waren viele Freunde” (“Many friends were at the wedding”) yields “With the wedding were many friends.” In each of these cases, a demonstrably incorrect translation (in context) for the preposition occurs.

Another example is the following: in German, “nach Hause” and “zu Hause” both translate roughly into English as “home”. The difference between the two is that one means “towards home” and the other means “at home”. Because we can say in English “I’m going home” and “I’m staying home”, we don’t need to mention “towards” or “at”. When translating these two sentences to German without explicitly stating “*at* home” in the second sentence, however, the engines we examined produced incoherent sentences. [12] translated it as “Ich bleibe nach Hause” (“I’m staying to home”), and [28] rendered a completely nonsensical “Ich bleibe Haupt” (“I’m staying head”).

### 4.2 Grammar Errors

Sometimes, even more basic grammar fails. While this may simply be a measure of a sentence being so complicated that a verb’s subject cannot be found, it is still quite noticeable when, for example, the wrong conjugation of a verb is used. In the following translation, “It appeared concerned about the expressions of the presidency candidate the fact that *it do not fight the radical groups in the Gaza*

*Strip*" [22,28], the third-person singular subject appears directly before the verb, and still the wrong form of the verb is chosen.

### 4.3 Word-for-Word Translations

One phenomenon which occurs again and again is the use of partial or complete word-for-word translations of constructions which are not grammatically correct in the target language. At best, this only results in word-order issues: "Was aber erwartet Israel wirklich von den Palästinensern nach der Wahl am 9.1.?" ("But what does Israel really expect from the Palestians after the election on January 9?") is translated by [28] as "What however really expects Israel from the Palestinians after the choice on 9.1.?" In this case, the meaning is not hampered because the construction is fairly simple, and the words translate well between the two languages. However, in a language like Russian where possession is indicated by something being "at" the owner, translation for things like "I have the pencils" in Russian come out as "the pencils are at me" in a word-for-word English translation. Unnatural constructions based on word-for-word translations are by far the most noticeable flaw in many of the translations we looked at.

### 4.4 Blatant Word Choice Errors

Less frequently, a completely unrelated word or phrase is chosen in the translation. For example, "I'm staying home" and "I am staying home" are both translated into German by [28] as "Ich bleibe Haupt" ("I'm staying head") instead of "Ich bleibe zu Hause". These are different from semantic errors and reflect some sort of flaw in the actual engine or its dictionary, clearly impacting translation quality.

### 4.5 Context and Semantics

As mentioned previously, the fact that most translation systems do not keep context makes translation problematic. The *Bare Bones Guide to HTML* [31] is a document giving basic web page authoring information. When the simplified Chinese translation of this document's entry for an HTML "Menu List" is translated into English, however, the result is "The vegetable unitarily enumerates" [33,28]. While one can see that whatever the Chinese phrase for "Menu List" is might in fact have something to do with a vegetable, the context information should lead to a choice that does not have to do with food. Similarly, the German translation ([28]) of "I ran through the woods" gives a translation ("Ich lief durch das Holz") that implies running through the *substance* "wood", not the "forest" sense. Without having enough contextual information, either based on statistics or the preceding verb/preposition combination, the translator is unable to decide that a forest is more likely to be run through than lumber is, and chooses the wrong word.

### 4.6 Additional Errors

Several other interesting error types were encountered which, for space reasons, we will only describe briefly.

- In many cases, words that are not in the source dictionary simply go untranslated; for example, an English translation of the registration for a Dutch news site gives “These can contain no spaties or leestekens” for “Deze mag geen spaties of leestekens bevatten.”
- Many languages use reflexive verbs to describe certain actions which are not reflexive in other languages; the reflexive article is often translated regardless of whether it is needed in the second language (e.g. “Ich kaemme mich” becomes “I comb myself”).
- Proper names which also translate to common words are sometimes translated; “Linda es muy Linda” (“Linda is very beautiful”) is translated by [28] as “It is continguous is very pretty” and “Pretty it is very pretty” by [12]. Moving the name does not always stop it from being translated, even when capitalized.
- Verb tense is often inexact in translation, as there is often no direct mapping between verb tenses in different languages.

### 4.7 Translations between Typologically Dissimilar Languages

Typologically distant languages are languages whose formal structures differ radically from one another. These structural differences manifest themselves in many areas (e.g. syntax (phrase and sentence structure), semantics (meaning structure) and morphology (word structure)). Not surprisingly, because of these differences, translations between languages that are typologically distant (Chinese and English, English and Arabic, etc) are frequently so bad as to be incoherent or unreadable. We did not consider these languages for this work, since the translation quality is often so poor that exchange of the resulting translations would likely be implausible.

For example, when translating the *“Bare Bones Guide to HTML”* page from Japanese [32] to English, [28] gives “Chasing order, link to the *HTML* guide whom it explained and is superior *WWW Help Page* is reference.” *(Note that italicized portions were already in English on the Japanese page)* The original English from which the Japanese was manually translated reads: “If you’re looking for more detailed step-by-step information, see my WWW Help Page.” The original English sentence is provided only for general meaning here, but it is clear that what is translated into English by the MT system is incomprehensible.

Because many translation systems were originally designed as a rough “first pass” for human translators who know both languages, it may well be that knowing the original language makes it possible to understand what is meant in the translation; in some sense, translators using such a tool would have to consciously or unconsciously be aware of the error types generated by the translation tool in order to produce accurate translations from it. While we did not explore these error types for this paper, an area for future improvement would be to look into the error types in various language pairs by asking bilinguals about the translations.

## 5 Implementation

This section describes some of the aspects of the implementation with focus on the different techniques that are used to obtain variations in the generated translations.

### 5.1 Translation Engines

The current implementation uses different translation services that are available on the Internet to obtain an initial translation. The current implementation supports three different services, and we plan on adding more in the future. Adding a new service only requires writing a function that translates a given sentence from a source language to the target language. Which subset of the available MT services should be used is up to the user to decide, but at least one engine must be selected.

A possible problem with selecting multiple different translation engines is that they might have distinct error characteristics (for example, one engine might not translate words with contractions). An adversary that is aware of such problems with a specific machine translation system might find out that half of all sentences have errors that match those characteristics. Since a normal user is unlikely to alternate between different translation engines, this would reveal the presence of a hidden message.

A better alternative is to use the same machine translation software but train it with different corpora. The specific corpora become part of the secret key used by the steganographic encoder; this use of a corpus as a key was previously discussed in another context [4]. As such, the adversary could no longer detect differences that are the result of a different machine translation algorithm. One problem with this approach is that acquiring good corpora is expensive. Furthermore, dividing a single corpus to generate multiple smaller corpora will result in worse translations, which can again lead to suspicious texts. That said, having full control over the translation engine may also allow for minor variations in the translation algorithm itself. For example, the GIZA++ system offers multiple algorithms for computing translations [14]. These algorithms mostly differ in how translation “candidate outcomes” are generated. Changing these options can also help to generate multiple translations.

After obtaining one or more translations from the translation engines, the tool produces additional variations using various post-processing algorithms. Problems with using multiple engines can be avoided by just using one high-quality translation engine and relying on the post-processing to generate alternative translations.

### 5.2 Semantic Substitution

Semantic substitution is one highly effective post-pass and has been used in previous approaches to hide information [4,7]. One key difference from previous work is that errors arising from semantic substitution are more plausible in translations compared to semantic substitutions in an ordinary text.

A typical problem with traditional semantic substitution is the need for substitution lists. A substitution list is a list of tuples consisting of words that are semantically close enough that subtituting one word for another in an arbitrary sentence is possible. For traditional semantic substitution, these lists are generated by hand. An example of a pair of words in a semantic substitution list would be `comfortable` and `convenient`. Not only is constructing substitution lists by hand tedious, but the lists must also be conservative in what they contain. For example, general substitution lists cannot contain word pairs such as `bright` and `light` since `light` could have been used in a different sense (meaning `effortless`, `unexacting` or even used as a noun).

Semantic substitution on translations does not have this problem. Using the original sentence, it is possible to automatically generate semantic substitutions that can even contain some of the cases mentioned above (which could not be added to a general monolingual substitution list). The basic idea is to translate back and forth between two languages to find semantically similar words. Assuming that the translation is accurate, the word in the source language can help provide the necessary contextual information to limit the substitutions to words that are semantically close in the current context.

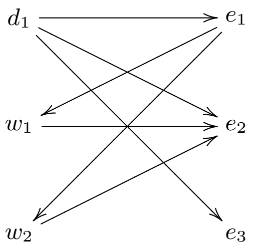

**Fig. 2.** Example for a translation graph produced by the semantic substitution discovery algorithm. Here two witnesses ($w_1$ and $w_2$) and the original word $d_1$ confirm the semantic proximity of $e_1$ and $e_2$. There is no witness for $e_3$, making $e_3$ an unlikely candidate for semantic substitution.

Suppose the source language is German (d) and the target language of the translation is English (e). The original sentence contains a German word $d_1$ and the translation contains a word $e_1$ which is a translation of $d_1$. The basic algorithm is the following:

- Find all other translations of $d_1$, call this set $E_{d_1}$. $E_{d_1}$ is the set of candidates for semantic substitution. Naturally $e_1 \in E_{d_1}$.
- Find all translations of $e_1$, call this set $D_{e_1}$. This set is called the set of witnesses.
- For each word $e \in E_{d_1} - \{e_1\}$ find all translations $D_e$ and count the number of elements in $D_e \cap D_{e_1}$. If that number is above a given threshold $t$, add $e$ to the list of possible semantic substitutes for $e_1$.

A witness is a word in the source language that also translates to both words in the target language, thereby confirming the semantic proximity of the two words. The witness threshold $t$ can be used to trade-off more possible substitutions against a higher potential for inappropriate substitutions.

The threshold does not have to be fixed. A heuristic can be used to increase the threshold if the number of possible substitutions for a word or in a sentence is extraordinarily high. Since the number of bits that can be encoded only increases with $\log_2 n$ for $n$ possible substitutions we suggest to increase $t$ whenever $n$ is larger than 8.

**Examples:** Given the German word “fein” and the English translation “nice”, the association algorithm run on the LEO (http://dict.leo.org/) dictionary gives the following semantic substitutions: for three witnesses, only “pretty” is generated; for two witnesses, “fine” is added; for just one witness, the list grows by “acute”, “capillary”, “dignified” and “keen”. Without witnesses (direct translations), the dictionary adds “smooth” and “subtle”. The word-pair “leicht” and ”light” gives “slight” (for three witnesses). However, “licht” and “light” gives “bright” and “clear”. In both cases the given substitutions match the semantics of the specific German word.

### 5.3 Adding plausible mistakes

Another possible post-pass adds mistakes that are commonly made by MT systems to the translations. The transformations that our implementation can use are based on the study of MT mistakes from section 4. The current system supports changing articles and prepositions using hand-crafted, language specific substitutions that attempt to mimic the likely errors observed.

### 5.4 Results from the Prototype

Different configurations of the system produce translations of varying quality, but even quality degradation is not predictable. Sometimes the generated modifications actually (by coincidence) improve the quality of the translation. For example, a good translation of the original French sentence “Dans toute la région, la vie est paralysée.” into English would be “In the entire region, life is paralysed.” Google’s translation is “In all the area, the life is paralysed.” wheras LinguaTec returns “In all of the region the life is crippled.”. Applying article substitution here can actually improve the translation: one of the choices generated by our implementation is “In all of the region, life is crippled.” Even aggressive settings are still somewhat meaningful: “In all **an** area, **a** life is paralysed.”

The following German text is the first part of a paragraph from a review about a Moroccan film called “Windhorse” [27]:

*Der marokkanische Film ”Windhorse” erzählt die Geschichte zweier, unterschiedlichen Generationen angehörender Männer, die durch Marokko reisen. Auf dem Weg suchen sie nach dem Einzigen, was ihnen wichtig ist: dem Sinn des Lebens.*

Our prototype system gives the following translation:

*The Moroccan film ”Windhorse” tells story from men belonging by two, different generations who travel through Morocco. They are looking for the only one which is important to them on the way: the sense of a life.*

For comparison, the source engine translations are also given:

Google: *The Moroccan film ”Windhorse” tells the history of two, different generations of belonging men, who travel by Morocco. On the way they look for the none one, which is important to them: the sense of the life.*

LinguaTec: *The Moroccan film ”Windhorse” tells the story of men belonging to two, different generations who travel through Morocco. They are looking for the only one which is important to them on the way: the meaning of the life.*

The Babelfish translation is identical to the Google translation except that “the none one” is replaced by “the only one”. LinguaTec provides some different syntactic structures and lexical choices, but looks quite similar.

Clearly the addition of more engines would lead to more variety in the LiT version. Sometimes substitutions lead to quality degradation (“belonging by” vs. “belonging to”), and sometimes not (“sense of the life” vs. “sense of a life”). Sometimes the encoding makes the engine choose the better version of a section of text to modify: “They are looking for the only one” vs. “they look for the none one”.

The original quality of the translations is not perfect. Furthermore, our version contains many of the same “differences” when compared to the source engines as the source engines have amongst themselves. Many of those differences are introduced by us (“story from men” vs. “story of men”) as opposed to coming directly from the source engines. While none of the texts are particularly readable, our goal is to plausibly imitate machine-translated text, not to solve the problem of perfect translation.

The example has most of prototype’s transformations enabled in order to achieve a higher bitrate. In general, this results in more degradation of the translation; decreasing the number of transformations might improve the quality, but would also decrease the bitrate by offering fewer variations. More transformations and source engines may make the resulting text potentially more likely to be flagged as suspicious by an adversary. For this example, we achieve a bitrate of 0.0164 uncompressed and 0.0224 compressed (9.33 bits per sentence); different hidden texts would, due to the encoding scheme used, achieve different bitrates. In general, we have found that the prototype gives us average bitrates of between 0.00265 and 0.00641 (uncompressed), and 0.00731 and 0.01671 (compressed), depending upon settings.

**Bitrates and system configuration** Figure 3 lists the different configurations and bitrates that are achieved by our prototype. The data is only intended to give a rough idea of the bitrates that can be achieved. An improved implementation using more rules or more translation engines can likely achieve higher bitrates. Also, it is impossible for us to give a precise metric for the quality of the generated translations. Still, the Figure can be used to give an impression for the bitrates that can be achieved with translation-based steganography. In order to allow for a fair comparison with other steganographic systems that use binary data, such as images, the bitrate is given for both uncompressed and compressed text.

| Id | Languages | Engines | SS-W | error-passes | Quality-Limit | bitrate | |
|---|---|---|---|---|---|---|---|
| | | | | | | ASCII-text | compressed |
| 1 | DE-EN | 1,2 | $\infty$ | - | 0.50 | 0.00226 | 0.00621 |
| 2 | DE-EN | 1,2 | 4 | - | 0.05 | 0.00266 | 0.00731 |
| 3 | DE-EN | 1,2 | 2 | - | 0.05 | 0.00178 | 0.00492 |
| 4 | DE-EN | 1,2 | 1 | - | 0.05 | 0.00281 | 0.00776 |
| 5 | DE-EN | 1,2 | 0 | - | 0.05 | 0.00488 | 0.01306 |
| 6 | DE-EN | 1,2 | $\infty$ | (1) | 0.05 | 0.00593 | 0.01585 |
| 7 | DE-EN | 1,2 | $\infty$ | (2) | 0.05 | 0.00247 | 0.00687 |
| 8 | DE-EN | 1,2 | 2 | (2) | 0.05 | 0.00283 | 0.00779 |
| 9 | DE-EN | 1,2 | 1 | (1)(2) | 0.00 | 0.00632 | 0.01671 |
| 10 | DE-EN | 1,2 | 0 | (1)(2) | 0.00 | 0.00721 | 0.01907 |
| 11 | FR-EN | 1,2 | $\infty$ | - | 0.50 | 0.00246 | 0.00670 |
| 12 | FR-EN | 1,2 | 4 | - | 0.05 | 0.00496 | 0.01344 |
| 13 | FR-EN | 1,2 | 2 | - | 0.05 | 0.00535 | 0.01429 |
| 14 | FR-EN | 1,2 | 1 | - | 0.05 | 0.00695 | 0.01834 |
| 15 | FR-EN | 1,2 | 0 | - | 0.05 | 0.00696 | 0.01834 |
| 16 | FR-EN | 1,2 | $\infty$ | (1) | 0.05 | 0.00551 | 0.01486 |
| 17 | FR-EN | 1,2 | $\infty$ | (2) | 0.05 | 0.00264 | 0.00721 |
| 18 | FR-EN | 1,2 | 2 | (2) | 0.05 | 0.00521 | 0.01401 |
| 19 | FR-EN | 1,2 | 1 | (1)(2) | 0.00 | 0.00818 | 0.02158 |

**Fig. 3.** Bitrates for the different configurations. Engine 1 is Google [16], Engine 2 is Linguatec [19]. SS-W lists the threshold for the number of witnesses in semantic substitution ($\infty$ for no semantic substitutions). The error-passes are (1) articles and (2) prepositions. The quality limit is the lower limit for the relative estimated translation quality (see Section 6.3). The BR columns give the bitrate for plaintext and compressed text, counting only the size of the generated translation (excluding the text in the source language).

In order to give an idea of the generated translations for the different settings (see Figure 3) we give translations for a German sentence (translated to English) and a French sentence (also translated to English). The original German sentences were “Gleich in den ersten Tagen nach der Katastrophe wies Unicef darauf hin, dass die Kinder unter den Opfern des Seebebens am schwersten betroffen sind. Wir sind heute in einem Maß von einer funktionierenden Infrastruktur abhängig, wie es nie zuvor der Fall war.", which in English would be "Already in the first days after the disaster, Unicef pointed out that children were hit worst among the victims of the seaquake. Today, the extent of our dependency on a working infrastructure is larger than ever.".

Google [16] translates this sentence as follows: "Directly in the first days after the disaster Unicef pointed out that the children among the victims of the sea-quake are most heavily concerned. We depend today in a measure on a functioning infrastructure, as it was the case never before.". The Linguatec engine returns "Is Unicef pointed out after the catastrophe within the first days that the children are affected most heavily under the victims of the seaquake. We are dependent in a measure of an operating infrastructure today how it the case never was before."

If we add errors with the article substitution (1), we could translations such as "Directly in the first days after the disaster Unicef pointed out that the children among the victims of **an** sea-quake are most heavily concerned. We depend today in a measure on a functioning infrastructure, as it was **an** case never before." For prepositions, a possible result is "Directly in the first days **behind** the disaster Unicef pointed out that the children among the victims of the sea-quake are most heavily concerned. We depend today in a measure **above** a functioning infrastructure, as it was the case never before."

## 6 Discussion

This section discusses various attacks on the steganographic encoding and possible defences against these attacks. The discussion is informal, as the system is based on MT imperfections that are hard to analyze formally (which is one of the reasons why MT is such a hard topic).

### 6.1 Future Machine Translation Systems

A possible problem that the presented steganographic encoding might face in the future is significant progress in machine translation. If machine translation were to become substantially more accurate, the possible margin of plausible mistakes might get smaller. However, one large category of machine translation errors today results from the lack of context that the machine translator takes into consideration.

In order to significantly improve existing machine translation systems one necessary feature would therefore be the preservation of context information from one sentence to the next. Only with that information will it be possible to eliminate certain errors. But introducing this context into the machine translation system also brings new opportunities for hiding messages in translations. Once machine translation software starts to keep context, it would be possible for the two parties that use the steganographic protocol to use this context as a secret key. By seeding their respective translation engines with $k$-bits of context they can make deviations in the translations plausible, forcing the adversary to potentially try $2^k$ possible contextual inputs in order to even establish the possibility that the mechanism was used. This is similar to the idea of splitting the corpus based on a secret key, with the difference that the overall quality of the per-sentence translations would not be affected.

### 6.2 Repeated Sentence Problem

A general problem with any approach to hiding messages in the translation is that if the text in the source language contains the same sentence twice it might be translated into two different sentences depending on the value of the bit that was hidden. Since machine translation systems (that do not keep context) would always produce the same sentence this would allow an attacker to suspect the use of steganography. The solution to this problem is to not use repeated sentences in the source text to hide data, and always output the translation that was used for the first occurence of the sentence.

This attack is similar to an attack in image steganography. If an image is digitally altered, variations in the colors in certain implausible areas of the picture might reveal the existence of a hidden message. Solving the problem is easier for text steganography since it is easier to detect that two sentences are identical than to detect that a series of pixels in an image belong to the same digitally constructed shape and thus must have the same color.

### 6.3 Statistical Attacks

Statistical attacks have been extremely successful at defeating steganography of images, audio and video (see, e.g., [13,20,25]). An adversary may have a statistical model (e.g. a language model) that translations from all available MT systems obey. For example, Zipf's law [21] states that the frequency of a word is inversely proportional to its rank in the sorted-by-frequency list of all words. Zipf's law holds for English, and in fact holds even within individual categories such as nouns, verbs, adjectives, etc.

Assuming that all plausible translation engines generally obey such a statistical model, the steganographic encoder must be careful not to cause telltale deviations from such distributions. Naturally, this is an arms race. Once such a statistical law is known, it is actually easy to modify the steganographic encoder to eliminate translations that deviate significantly from the required distributions. For example, Golle and Farahat [15] point out (in the different context of encryption) that it is possible to extensively modify a natural language text without straying noticeably from Zipf's law. In other words, this is a very manageable difficulty, as long as the steganographic system is made "Zipf-aware".

We cannot preclude the existence of yet-undiscovered language models for translations that might be violated by our existing implementation. However, we expect that discovering and validating such a model is a non-trivial task for the adversary. On the other hand, given such a model (as we pointed out above) it is easy to modify the steganographic system so as to eliminate deviations by avoiding sentences that would be flagged. Section 7 sketches various statistical models for attacks that might be useful against the existing prototype implementation.

### 6.4 Use for Watermarking

The technique of this paper can be used for watermarking, in a manner that does not require the original text (or any reference translation) for reading the mark. The idea for not requiring the original in order to recover the message, which was mentioned in Section 3.3, is now sketched in more detail.

We begin with a fragile version of the scheme. Let the bits of the mark be denoted by $b_1, \ldots, b_n$. Let $k \in \mathbb{N}$ be a parameter that will be determined later. The technique consists of using a (secret) random seed $s$ as key for determining those places where the $n$ bits of the mark will be embedded. Let the random sequence generated by the seed consist of numbers $r_1, \ldots, r_{k \cdot n}$ and let the corresponding places in the text where the bits of the mark will be embedded be $p_1, \ldots, p_{k \cdot n}$ (with $p_i$ denoting the spot for the $i$-th bit). Of course $p_i$ is determined by $r_i$.

The $p_i$'s are partitioned into groups of size $k$ each. Let the resulting groups be $C_1, \ldots, C_n$ ($C_1$ consists of $p_1, \ldots, p_k$). In what follows $P_j$ will denote the concatenation of the contents of the $k$ positions $p_i$ that are in group $C_j$ (so $P_j$ changes as the algorithm modifies those $k$ positions – e.g., when the algorithm replaces “cat” by “feline” that replacement is reflected within $P_j$). Each $C_j$ is associated with $s_j$ which is defined to be the least significant bit of $H_s(P_j)$ where $H_s$ is a keyed cryptographic one-way hash function having $s$ as key (recall that $s$ is the secret seed that determined the $r_i$).

As a result, $s_j$ changes with 50% probability as $P_j$ is modified. In order to embed $b_j$ in $C_j$ the algorithm “tortures $C_j$ until it confesses”: $C_j$ is modified until its $s_j$ equals $b_j$. Every one of the $k$ possible changes made within $C_j$ has a 50% change of producing an $s_j$ that equals the target $b_j$, and the probability that we fail $e$ times is $2^{-e}$. A large choice for $k$ will give the algorithm more room for modifications and thus ensure that the embedding will fail with reasonably low probability. It is possible to choose a small $k$ and use an error-correcting code in order to correct bits that could not be embedded properly.

The advantage of the scheme is that the receiver can receive all of the $s_j$ from the seed $s$ without needing the original text or any reference baseline translation of it: the received message and the seed are all that is required to retrieve the mark.

More robust versions of the scheme can be obtained by using the techniques described in [4], which include the use of *markers* (a marker is a sentence that merely indicates that the group of contiguous sentences that immediately follow it are watermark-carrying, so the marker is not itself watermark-carrying). One of the ways of determining markers is by a secret (because keyed) ordering of the sentences, the markers being the sentences that are lowest in that secret ordering – see [4] for details, and for an analysis that quantifies the scheme's resilience against different kinds of attacks.

### 6.5 Other applications

While we have explored the possibility of using the inherent noise of natural language translation to hide data, we suspect that there may be other areas where transformation spaces exist which exhibit a similar lack of rigidity. For example, compilers doing source translation have a variety of possible output possibilities that still preserve semantics. Finding a way to hide information with these possibilities while still mimicking the properties of various optimization and transformation styles is a possibility for future work.

## 7 Statistical Attack Vectors

Statistical attacks fall into two categories. The first attacks the translation. The translation models look at both the source and the target text. The second category, language models, only look at the generated translations and try to find inconsistencies within that text, without reference to the original text.

### 7.1 Translation Models

**Word Count** The average word count may yield a different ratio between the source and target languages when steganography is in use. The rationale is that the steganographic engine will discard words more readily than inserting them (though, insertion of more words can occur in the current engine through preposition substitution). An unmodified translation would then yield a higher ratio of average number of words in the source language to the average number of words in the target language compared to a text that had information hidden in it.

**Engine Errors** Certain engines are given to certain types of errors. If a text is translated by an engine and has no information hidden in it it should make the errors characteristic to that engine 100% of the time. Taking the ratio of the number of errors characteristic to one engine over the number of errors characteristic to another should always yield a 0 or 1 (tolerance can be allowed in case the user hand corrected some errors). If the ratio yields something outside of that tolerance it can be seen that there are a mixture of error types, and therefore a mixture of engines being used. Automatic classification of engine-specific errors is likely to be rather difficult. Also, if the translation engines used by the encoder are unknown to the attacker this attack cannot be used.

### 7.2 Language Model

**Character Count** The idea here is quite similar to the previous model. An untouched translation will yield a lower ratio of average number of characters in the source language to the average number in the target language. The rationale here is when substitution occurs the substituted word is more likely to be more complex and longer than the original translation.

**Nouns Without Articles** Since the current steganographic engine discards articles for nouns on occasion and never inserts new ones (since it does not have enough semantic knowledge to detect nouns) the total percentage of nouns without articles should be higher for a text containing hidden information.

**Witness Count** When a semantic substitution is made by our system the word that is chosen for replacement is likely to be more specific or complex than the word that it is replacing. With that in mind the number of witnesses for the word selected to hide information should ordinarily be lower than the word that it is replacing. A normal translation should thus yield a higher ratio between semantic substitution witnesses from the target to the source language than a translation that contains a hidden message.

## 8 Conclusion

This paper introduced a new steganographic encoding scheme based on hiding messages in the noise that is inherent to natural language translation. The steganographic message is hidden in the translation by selecting between multiple translations which are generated by either modifying the translation process or by post-processing the translated sentences. In order to defeat the system, an adversary has to demonstrate that the resulting translation is unlikely to have been generated by any automatic machine translation system. A study of common mistakes in machine translation was used to come up with plausible modifications that could be made to the translations. It was demonstrated that the variations produced by the steganographic encoding are similar to those of various unmodified machine translation systems, demonstrating that it would be impractical for an adversary to establish the existence of a hidden message. The highest bitrate that our prototype could achieve with this new steganographic encoding is about 0.01671.

### Acknowledgements

Portions of this work were supported by Grants IIS-0325345, IIS-0219560, IIS-0312357, and IIS-0242421 from the National Science Foundation, Contract N00014-02-1-0364 from the Office of Naval Research, by sponsors of the Center for Education and Research in Information Assurance and Security, and by Purdue Discovery Park's e-enterprise Center.

## References

1. R. Agrawal, P. Haas, and J. Kiernan. Watermarking relational data: Framework, algorithms and analysis. *The VLDB Journal*, 12(2):157–169, 2003.
2. Y. Al-Onaizan, J. Curin, M. Jahr, K. Knight, J. Lafferty, I. D. Melamed, F. J. Och, D. Purdy, N. A. Smith, and D. Yarowsky. Statistical machine translation, final report, JHU workshop, 1999. `http://www.clsp.jhu.edu/ws99/projects/mt/final_report/mt-final-report.ps`.

3. M. Atallah, V. Raskin, M. Crogan, C. Hempelmann, F. Kerschbaum, D. Mohamed, and S. Naik. Natural language watermarking: Design, analysis, and a proof-of-concept implementation. In *Proceedings of the 4th International Information Hiding Workshop 2001*, 2001.
4. M. Atallah, V. Raskin, C. Hempelmann, M. Karahan, R. Sion, and K. Triezenberg. Natural language watermarking and tamperproofing. In *Proceedings of the 5th International Information Hiding Workshop 2002*, 2002.
5. P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1993.
6. M. Chapman and G. Davida. Hiding the hidden: A software system for concealing ciphertext in innocuous text. In *Information and Communications Security — First International Conference*, volume Lecture Notes in Computer Science 1334, Beijing, China, 11–14 1997.
7. M. Chapman, G. Davida, and M. Rennhard. A practical and effective approach to large-scale automated linguistic steganography. In *Proceedings of the Information Security Conference (ISC '01)*, pages 156–165, Malaga, Spain, 2001.
8. P. R. Clarkson and R. Rosenfeld. Statistical language modeling using the cmu-cambridge toolkit. In *Proceedings of ESCA Eurospeech*, 1997.
9. C. Collberg and C. Thomborson. On the limits of software watermarking. Technical Report 164, Department of Computer Science, The University of Auckland, Private Bag 92019, Auckland, New Zealand, Aug. 1998.
10. C. Collberg and C. Thomborson. Software watermarking: Models and dynamic embeddings. In *ACM Symp. on Principles of Programming Languages (POPL)*, pages 311–324, 1999.
11. C. Collberg and C. Thomborson. Software watermarking: models and dynamic embeddings. In *ACM SIGPLAN–SIGACT POPL'99*, San Antonio, Texas, USA, Jan. 1999.
12. Smart Link Corporation. Promt-online. http://translation2.paralink.com/.
13. J. Fridrich, M. Goljan, and D. Soukal. Higher-Order Statistical Steganalysis of Palette. In *Proceedings of the SPIE International Conference on Security and Watermarking of Multimedia Contents*, volume 5020, pages 178–190, San Jose, CA, 21 – 24 January 2003.
14. U. Germann, M. Jahr, D. Marcu, and K. Yamada. Fast decoding and optimal decoding for machine translation. In *Proceedings of the 39th Annual Conference of the Association for Computational Linguistics (ACL-01)*, 2001.
15. P. Golle and A. Farahat. Defending email communication against profiling attacks. In *Proceedings of the 2004 ACM workshop on Privacy in the electronic society (WPES 04)*, pages 39–40, 2004.
16. Google. Google translation. http://www.google.com/language_tools.
17. D. Huffman. A method for the construction of minimum redundancy codes. *Proceedings of the Institute of Radio Engineers*, 40:1098–1101, 1951.
18. N. F. Johnson and S. Jajodia. Steganalysis of images created using current steganography software. In *IHW'98 - Proceedings of the International Information hiding Workshop*, April 1998.
19. Linguatec. Linguatec translation. http://www.linguatec.de/.
20. S. Lyu and H. Farid. Detecting Hidden Messages using Higher-Order Statistics and Support Vector Machines. In *Proceedings of the Fifth Information Hiding Workshop*, volume LNCS, 2578, Noordwijkerhout, The Netherlands, October, 2002. Springer-Verlag.

21. C. D. Manning and H. Schuetze. *Review of Foundations of Statistical Natural Language Processing.* MIT Press, Cambridge, MA, 1999.
22. B. Marx. Friedensverhandlungen brauchen ruhe. *Deutsche Welle Online*, Jan 2005.
23. F. J. Och and H. Ney. A comparison of alignment models for statistical machine translation. In *COLING00*, pages 1086–1090, Saarbrücken, Germany, August 2000.
24. F. J. Och and H. Ney. Improved statistical alignment models. In *ACL00*, pages 440–447, Hongkong, China, October 2000.
25. A. Pfitzmann and A. Westfeld. Attacks on steganographic systems. In *Third Information Hiding Workshop*, volume LNCS, 1768, pages 61–76, Dresden, Germany, 1999. Springer-Verlag.
26. R. Sion, M.J. Atallah, and S. Prabhakar. Rights protection for relational data. *IEEE Trans. Knowl. Data Eng.*, 16(12):1509–1525, 2004.
27. S. Suren. Neue bilder der arabischen jugend. *Deutsche Welle Online*, March 2005.
28. Systran Language Translation Technologies. Systran. http://systransoft.com/.
29. P. Wayner. Mimic functions. *Cryptologia*, XVI(3):193–214, 1992.
30. P. Wayner. *Disappearing Cryptography: Information Hiding: Steganography and Watermarking.* Morgan Kaufmann, 2nd edition edition, 2002.
31. Kevin Werbach. The bare bones guide to html. http://werbach.com/barebones/download.html, 1999.
32. Kevin Werbach and Hisashi Nishimura. The bare bones guide to html (japanese translation). http://werbach.com/barebones/jp/barebone-j.html.
33. Kevin Werbach and Iap Sin-Guan. The bare bones guide to html (simplified chinese translation). http://werbach.com/barebones/barebone_cn.html.

## A Extended Example

This section gives an extended example for running the tool on the first part of the Communist Manifesto, translating from German to English with preposition substitution and semantics substitution with two witnesses. The output text has the text “Hail, hail” embedded yeilding a bitrate of 0.00262 (0.00656 compressed).

### A.1 Source Text

Die Geschichte aller bisherigen Gesellschaft ist die Geschichte von Klassenkämpfen.

Freier und Sklave, Patrizier und Plebejer, Baron und Leibeigener, Zunftbürger und Gesell, kurz, Unterdrücker und Unterdrückte standen in stetem Gegensatz zueinander, führten einen ununterbrochenen, bald versteckten, bald offenen Kampf, einen Kampf, der jedesmal mit einer revolutionären Umgestaltung der ganzen Gesellschaft endete oder mit dem gemeinsamen Untergang der kämpfenden Klassen.

In den früheren Epochen der Geschichte finden wir fast überall eine vollständige Gliederung der Gesellschaft in verschiedene Stände, eine mannigfaltige Abstufung der gesellschaftlichen Stellungen. Im alten Rom haben wir Patrizier, Ritter, Plebejer, Sklaven; im Mittelalter Feudalherren, Vasallen, Zunftbürger, Gesellen, Leibeigene, und noch dazu in fast jeder dieser Klassen besondere Abstufungen.

Die aus dem Untergang der feudalen Gesellschaft hervorgegangene moderne bürgerliche Gesellschaft hat die Klassengegensätze nicht aufgehoben. Sie hat nur neue Klassen, neue Bedingungen der Unterdrückung, neue Gestaltungen des Kampfes an die Stelle der alten gesetzt.

Unsere Epoche, die Epoche der Bourgeoisie, zeichnet sich jedoch dadurch aus, daß sie die Klassengegensätze vereinfacht hat. Die ganze Gesellschaft spaltet sich mehr und mehr in zwei große feindliche Lager, in zwei große, einander direkt gegenüberstehende Klassen: Bourgeoisie und Proletariat.

Aus den Leibeigenen des Mittelalters gingen die Pfahlbürger der ersten Städte hervor; aus dieser Pfahlbürgerschaft entwickelten sich die ersten Elemente der Bourgeoisie.

Die Entdeckung Amerikas, die Umschiffung Afrikas schufen der aufkommenden Bourgeoisie ein neues Terrain. Der ostindische und chinesische Markt, die Kolonisierung von Amerika, der Austausch mit den Kolonien, die Vermehrung der Tauschmittel und der Waren überhaupt gaben dem Handel, der Schiffahrt, der Industrie einen nie gekannten Aufschwung und damit dem revolutionären Element in der zerfallenden feudalen Gesellschaft eine rasche Entwicklung.

Die bisherige feudale oder zünftige Betriebsweise der Industrie reichte nicht mehr aus für den mit neuen Märkten anwachsenden Bedarf. Die Manufaktur trat an ihre Stelle. Die Zunftmeister wurden verdrängt durch den industriellen Mittelstand; die Teilung der Arbeit zwischen den verschiedenen Korporationen verschwand vor der Teilung der Arbeit in der einzelnen Werkstatt selbst.

Aber immer wuchsen die Märkte, immer stieg der Bedarf. Auch die Manufaktur reichte nicht mehr aus. Da revolutionierte der Dampf und die Maschinerie die industrielle Produktion. An die Stelle der Manufaktur trat die moderne große Industrie, an die Stelle des industriellen Mittelstandes traten die industriellen Millionäre, die Chefs ganzer industrieller Armeen, die modernen Bourgeois.

Die große Industrie hat den Weltmarkt hergestellt, den die Entdeckung Amerikas vorbereitete. Der Weltmarkt hat dem Handel, der Schiffahrt, den Landkommunikationen eine unermeßliche Entwicklung gegeben. Diese hat wieder auf die Ausdehnung der Industrie zurückgewirkt, und in demselben Maße, worin Industrie, Handel, Schiffahrt, Eisenbahnen sich ausdehnten, in demselben Maße entwickelte sich die Bourgeoisie, vermehrte sie ihre Kapitalien, drängte sie alle vom Mittelalter her überlieferten Klassen in den Hintergrund.

Wir sehen also, wie die moderne Bourgeoisie selbst das Produkt eines langen Entwicklungsganges, einer Reihe von Umwälzungen in der Produktions- und Verkehrsweise ist.

Jede dieser Entwicklungsstufen der Bourgeoisie war begleitet von einem entsprechenden politischen Fortschritt . Unterdrückter Stand unter der Herrschaft der Feudalherren, bewaffnete und sich selbst verwaltende Assoziation in der Kommune (3), hier unabhängige städtische Republik , dort dritter steuerpflichtiger Stand der Monarchie , dann zur Zeit der Manufaktur Gegengewicht gegen den Adel in der ständischen oder in der absoluten Monarchie , Hauptgrundlage der großen Monarchien überhaupt, erkämpfte sie sich endlich seit der Herstellung der großen Industrie und des Weltmarktes im modernen Repräsentativstaat die ausschließliche politische Herrschaft. Die moderne Staatsgewalt ist nur ein Ausschuß, der die gemeinschaftlichen Geschäfte der ganzen Bourgeoisklasse verwaltet.

## A.2 Output Text

The history of all past society is the history of class warfares.

Suitor and slave, Patrizier and Plebejer, Baron and body-own, Zunftbuerger and join, Briefly, Eliminator and suppressed stood in constant contrast to each other, Led a continuous, Soon hid, Soon open fight, A fight, a revolutionary transformation to the whole society each time ended or the common fall of the fighting classes.

In the earlier epochs of history we find nearly everywhere a complete arrangement of the society into different conditions, A diverse gradation of the social positions. In old Rome we have Patrizier, Knight, Plebejer, Slaves; In the Middle Ages feudal sirs, Vasallen, Zunftbuerger, Skilled workers, Body-own, And still to it in nearly of these classes special gradations.

The modern civil society come out from the fall of the feudalen society did not waive the class contrasts. It has only new classes, New conditions of the oppression, New organizations of the fight to the place of the old set.

Our epoch, The epoch of the bourgeoisie, Stands out, however, due to it, That it simplified the class contrasts. The whole society splits in two large hostile camps more and more, Into two great ones, Each other directly facing classes: Bourgeoisie and proletariat.

From the body-own of the Middle Ages the stake citizens of the first cities followed; From stake citizenry the first elements of the Bourgeoisie developed.

The discovery America, A new land created the sailing around of Africa for the paying bourgeoisie. The East Indian and Chinese market, Colonizing of America, The exchange with the colonies, The increase of the mediums of exchange and the goods gave the trade at all, Shipping, The industry an upswing and thus, never known, the revolutionary element in the feudalen society disintegrating a rapid development.

The past feudale or zuenftige mode of operation of the industry was not enough any longer out for the need increasing with new markets. The manufactory took its job. The guild masters were replaced by the industrial middle classes; The division of the work between the different Korporationen disappeared before the division of the work in the individual workshop.

But the markets always grew, The need always rose. Also the manufaktur was not sufficient. The steam and the machinery revolutionized the industrial production there. The modern large industry took the place of the manufactory, To the place of the industriellen of middle class the industriellen millionaires stepped, The bosses of whole industrieller armies, The modern Bourgeois.

The large industry manufactured the world market, The discovery of America prepared this one. The world market has the trade, Shipping, An immense development given to the country communications. This has reacted again upon the extension of the industry, And in the same measure, Into what industry, Trade, Shipping, Railways expanded, In the same himself developed mass for the bourgeoisie, Increased it its capitals, She pressed all classes handed down here of the Middle Ages to the background.

So we see, Like the modern Bourgeoisie themselves the product of a long development course, A set of circulations in production and traffic way is. Each of these entwicklungsstufen of the Bourgeoisie was accompanied of appropriate political progress.