Watermarking the Outputs of Structured Prediction with an application in Statistical Machine Translation.

Ashish Venugopal[1] **Jakob Uszkoreit**[1] **David Talbot**[1] **Franz J. Och**[1] **Juri Ganitkevitch**[2]

[1]Google, Inc.
1600 Amphitheatre Parkway
Mountain View, 94303, CA
{avenugopal,uszkoreit,talbot,och}@google.com

[2]Center for Language and Speech Processing
Johns Hopkins University
Baltimore, MD 21218, USA
juri@cs.jhu.edu

Abstract

We propose a general method to watermark and probabilistically identify the structured outputs of machine learning algorithms. Our method is robust to local editing operations and provides well defined trade-offs between the ability to identify algorithm outputs and the quality of the watermarked output. Unlike previous work in the field, our approach does not rely on controlling the inputs to the algorithm and provides probabilistic guarantees on the ability to identify collections of results from one's own algorithm. We present an application in statistical machine translation, where machine translated output is watermarked at minimal loss in translation quality and detected with high recall.

1 Motivation

Machine learning algorithms provide structured results to input queries by simulating human behavior. Examples include automatic machine translation (Brown et al., 1993) or automatic text and rich media summarization (Goldstein et al., 1999). These algorithms often estimate some portion of their models from publicly available human generated data. As new services that output structured results are made available to the public and the results disseminated on the web, we face a daunting new challenge: Machine generated structured results contaminate the pool of naturally generated human data. For example, machine translated output and human generated translations are currently both found extensively on the web, with no automatic way of distinguishing between them. Algorithms that mine data from the web (Uszkoreit et al., 2010), with the goal of learning to simulate human behavior, will now learn models from this contaminated and potentially self-generated data, reinforcing the errors committed by earlier versions of the algorithm.

It is beneficial to be able to identify a set of encountered structured results as having been generated by one's own algorithm, with the purpose of filtering such results when building new models.

Problem Statement: We define a structured result of a query q as $r = \{z_1 \cdots z_L\}$ where the order and identity of elements z_i are important to the quality of the result r. The structural aspect of the result implies the existence of alternative results (across both the order of elements and the elements themselves) that might vary in their quality.

Given a collection of N results, $\mathcal{C}_N = r_1 \cdots r_N$, where each result r_i has k ranked alternatives $D_k(q_i)$ of relatively similar quality and queries $q_1 \cdots q_N$ are arbitrary and not controlled by the watermarking algorithm, we define the watermarking task as:

Task. *Replace r_i with $r'_i \in D_k(q_i)$ for some subset of results in $\mathcal{C}_N$ to produce a watermarked collection $\mathcal{C}'_N$*
such that:

- $\mathcal{C}'_N$ is probabilistically identifiable as having been generated by one's own algorithm.

Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 1363–1372,
Edinburgh, Scotland, UK, July 27–31, 2011. ©2011 Association for Computational Linguistics

- the degradation in quality from $\mathcal{C}_N$ to the watermarked $\mathcal{C}'_N$ should be analytically controllable, trading quality for detection performance.
- $\mathcal{C}'_N$ should not be detectable as watermarked content without access to the generating algorithms.
- the detection of $\mathcal{C}'_N$ should be robust to simple edit operations performed on individual results $r \in \mathcal{C}'_N$.

2 Impact on Statistical Machine Translation

Recent work(Resnik and Smith, 2003; Munteanu and Marcu, 2005; Uszkoreit et al., 2010) has shown that multilingual parallel documents can be efficiently identified on the web and used as training data to improve the quality of statistical machine translation.

The availability of free translation services (Google Translate, Bing Translate) and tools (Moses, Joshua), increase the risk that the content found by parallel data mining is in fact generated by a machine, rather than by humans. In this work, we focus on statistical machine translation as an application for watermarking, with the goal of discarding documents from training if they have been generated by one's own algorithms.

To estimate the magnitude of the problem, we used parallel document mining (Uszkoreit et al., 2010) to generate a collection of bilingual document pairs across several languages. For each document, we inspected the page content for source code that indicates the use of translation modules/plug-ins that translate and publish the translated content.

We computed the proportion of the content within our corpus that uses these modules. We find that a significant proportion of the mined parallel data for some language pairs is generated via one of these translation modules. The top 3 languages pairs, each with parallel translations into English, are Tagalog (50.6%), Hindi (44.5%) and Galician (41.9%). While these proportions do not reflect impact on each language's monolingual web, they are certainly high enough to affect machine translations systems that train on mined parallel data. In this work, we develop a general approach to watermark structured outputs and apply it to the outputs of a statistical machine translation system with the goal of identifying these same outputs on the web. In the context of the watermarking task defined above, we output selecting alternative translations for input source sentences. These translations often undergo simple edit and formatting operations such as case changes, sentence and word deletion or post editing, prior to publishing on the web. We want to ensure that we can still detect watermarked translations despite these edit operations. Given the rapid pace of development within machine translation, it is also important that the watermark be robust to improvements in underlying translation quality. Results from several iterations of the system within a single collection of documents should be identifiable under probabilistic bounds.

While we present evaluation results for statistical machine translation, our proposed approach and associated requirements are applicable to any algorithm that produces structured results with several plausible alternatives. The alternative results can arise as a result of inherent task ambiguity (for example, there are multiple correct translations for a given input source sentence) or modeling uncertainty (for example, a model assigning equal probability to two competing results).

3 Watermark Structured Results

Selecting an alternative r' from the space of alternatives $D_k(q)$ can be stated as:

$$r' = \arg\max_{r \in D_k(q)} w(r, D_k(q), h) \qquad (1)$$

where w ranks $r \in D_k(q)$ based on r's presentation of a watermarking signal computed by a hashing operation h. In this approach, w and its component operation h are the only *secrets* held by the watermarker. This selection criterion is applied to all system outputs, ensuring that watermarked and non-watermarked version of a collection will never be available for comparison.

A specific implementation of w within our watermarking approach can be evaluated by the following metrics:

- False Positive Rate: how often non-watermarked collections are *falsely* identified as watermarked.
- Recall Rate: how often watermarked collections are *correctly* identified as watermarked.
- Quality Degradation: how significantly does $\mathcal{C}'_N$ differ from $\mathcal{C}_N$ when evaluated by task specific quality metrics.

While identification is performed at the collection level, we can scale these metrics based on the size of each collection to provide more task sensitive metrics. For example, in machine translation, we count the number of words in the collection towards the false positive and recall rates.

In Section 3.1, we define a random hashing operation h and a task independent implementation of the selector function w. Section 3.2 describes how to classify a collection of watermarked results. Section 3.3 and 3.4 describes refinements to the selection and classification criteria that mitigate quality degradation. Following a comparison to related work in Section 4, we present experimental results for several languages in Section 5.

3.1 Watermarking: $\mathcal{C}_N \rightarrow \mathcal{C}'_N$

We define a random hashing operation h that is applied to result r. It consists of two components:

- A hash function applied to a structured result r to generate a bit sequence of a fixed length.
- An optional mapping that maps a single candidate result r to a set of sub-results. Each sub-result is then hashed to generate a concatenated bit sequence for r.

A good hash function produces outputs whose bits are independent. This implies that we can treat the bits for any input structured results as having been generated by a binomial distribution with equal probability of generating 1s vs 0s. This condition also holds when accumulating the bit sequences over a collection of results as long as its elements are selected uniformly from the space of possible results. Therefore, *the bits generated from a collection of unwatermarked results will follow a binomial distribution with parameter* $p = 0.5$. This result provides a null hypothesis for a statistical test on a given bit sequence, testing whether it is likely to have been generated from a binomial distribution binomial(n, p) where $p = 0.5$ and n is the length of the bit sequence.

For a collection $\mathcal{C}_N = r_1 \cdots r_N$, we can define a watermark ranking function w to systematically select alternatives $r'_i \in D_k(q)$, such that the resulting $\mathcal{C}'_N$ is *unlikely* to produce bit sequences that follow the $p = 0.5$ binomial distribution. A straightforward biasing criteria would be to select the candidate whose bit sequence exhibits the highest ratio of 1s. w can be defined as:

$$w(r, D_k(q), h) = \frac{\#(1, h(r))}{|h(r)|} \qquad (2)$$

where $h(r)$ returns the randomized bit sequence for result r, and $\#(x, \vec{y})$ counts the number of occurrences of x in sequence $\vec{y}$. Selecting alternatives results to exhibit this bias will result in watermarked collections that exhibit this same bias.

3.2 Detecting the Watermark

To classify a collection $\mathcal{C}_N$ as watermarked or non-watermarked, we apply the hashing operation h on each element in $\mathcal{C}_N$ and concatenate the sequences. This sequence is tested against the null hypothesis that it was generated by a binomial distribution with parameter $p = 0.5$. We can apply a Fisherian test of statistical significance to determine whether the observed distribution of bits is unlikely to have occurred by chance under the null hypothesis (binomial with $p = 0.5$).

We consider a collection of results that *rejects* the null hypothesis to be watermarked results generated by our own algorithms. The p-value under the null hypothesis is efficiently computed by:

$$p - value \;=\; P_n(X \geq x) \quad (3)$$

$$= \sum_{i=x}^{n} \binom{n}{i} p^i (1-p)^{n-i} \quad (4)$$

where x is the number of 1s observed in the collection, and n is the total number of bits in the sequence. Comparing this p-value against a desired significance level α, we reject the null hypothesis for collections that have $P_n(X \geq x) < \alpha$, thus deciding that such collections were generated by our own system.

This classification criteria has a fixed false positive rate. Setting $\alpha = 0.05$, we know that 5% of *non-watermarked* bit sequences will be *falsely* labeled as watermarked. This parameter α can be controlled on an application specific basis. By biasing the selection of candidate results to produce more 1s than 0s, we have defined a watermarking approach that exhibits a fixed false positive rate, a probabilistically bounded detection rate and a task independent hashing and selection criteria. In the next sections, we will deal with the question of robustness to edit operations and quality degradation.

3.3 Robustness and Inherent Bias

We would like the ability to identify watermarked collections to be robust to simple edit operations. Even slight modifications to the elements within an item r would yield (by construction of the hash function), completely different bit sequences that no longer preserve the biases introduced by the watermark selection function.

To ensure that the distributional biases introduced by the watermark selector are preserved, we can optionally map individual results into a set of sub-results, each one representing some local structure of r. h is then applied to each subresult and the results concatenated to represent r. This mapping is defined as a component of the h operation.

While a particular edit operation might affect a small number of sub-results, the majority of the bits in the concatenated bit sequence for r would remain untouched, thereby limiting the damage to the biases selected during watermarking. This is of course no defense to edit operations that are applied globally across the result; our expectation is that such edits would either significantly degrade the quality of the result or be straightforward to identify directly.

For example, a sequence of words $r = z_1 \cdots z_L$ can be mapped into a set of consecutive n-gram sequences. Operations to edit a word z_i in r will only affect events that consider the word z_i. To account for the fact that alternatives in $D_k(q)$ might now result in bit sequences of different lengths, we can generalize the biasing criteria to directly reflect the expected contribution to the watermark by defining:

$$w(r, D_k(q), h) = P_n(X \geq \#(1, h(r))) \quad (5)$$

where P_n gives probabilities from binomial$(n = |h(r)|, p = 0.5)$.

Inherent collection level biases: Our null hypothesis is based on the assumption that collections of results draw uniformly from the space of possible results. This assumption might not always hold and depends on the type of the results and collection. For example, considering a text document as a collection of sentences, we can expect that some sentences might repeat more frequently than others.

This scenario is even more likely when applying a mapping into sub-results. n-gram sequences follow long-tailed or Zipfian distributions, with a small number of n-grams contributing heavily toward the total number of n-grams in a document.

A random hash function guarantees that inputs are distributed uniformly at random over the output range. However, the same input will be assigned the same output deterministically. Therefore, if the distribution of inputs is heavily skewed to certain elements of the input space, the output distribution will not be uniformly distributed. The bit sequences resulting from the high frequency sub-results have the potential to generate inherently biased distributions when accumulated at the collection level. We want to choose a mapping that tends towards generating uniformly from the space of sub-results. We can empirically measure the quality of a sub-result mapping for a specific task by computing the false positive rate on non-watermarked collections. For a given significance level α, an ideal mapping would result in false positive rates close to α as well.

Figure 1 shows false positive rates from 4 alternative mappings, computed on a large corpus of French documents (see Table 1 for statistics). Classification decisions are made at the collection level (documents) but the contribution to the false positive rate is based on the number of words in the classified document. We consider mappings from a result (sentence) into its 1-grams, $1-5$-grams and $3-5$ grams as well as the non-mapping case, where the full result is hashed.

Figure 1 shows that the 1-grams and $1-5$-gram generate sub-results that result in heavily biased false positive rates. The $3-5$ gram mapping yields false positive rates close to their theoretically expected values. [1] Small deviations are expected since documents make different contributions to the false positive rate as a function of the number of words that they represent. For the remainder of this work, we use the 3-5 gram mapping and the full sentence mapping, since the alternatives generate inherently distributions with very high false positive rates.

3.4 Considering Quality

The watermarking described in Equation 3 chooses alternative results on a per result basis, with the goal of influencing collection level bit sequences. The selection criteria as described will choose the most biased candidates available in $D_k(q)$. The parameter k determines the extent to which lesser quality alternatives can be chosen. If all the alternatives in each $D_k(q)$ are of relatively similar quality, we expect minimal degradation due to watermarking.

Specific tasks however can be particularly sensitive to choosing alternative results. Discriminative approaches that optimize for arg max selection like (Och, 2003; Liang et al., 2006; Chiang et al., 2009) train model parameters such that the top-ranked result is well *separated* from its competing alternatives. Different queries also differ in the inherent ambiguity expected from their results; sometimes there really is just one correct result for a query, while for other queries, several alternatives might be equally good.

By generalizing the definition of the w function to interpolate the estimated loss in quality and the gain in the watermarking signal, we can trade-off the ability to identify the watermarked collections against quality degradation:

$$\begin{aligned} w(r, D_k(q), f_w) = \lambda * \text{gain}(r, D_k(q), f_w) \\ -(1-\lambda) * \text{loss}(r, D_k(q)) \end{aligned} \tag{6}$$

Loss: The $\text{loss}(r, D_k(q))$ function reflects the quality degradation that results from selecting alternative r as opposed to the best ranked candidate in $D_k(q)$). We will experiment with two variants:

$$loss_{rank}(r, D_k(q)) = (\text{rank}(r) - k)/k$$

$$loss_{cost}(r, D_k(q)) = (\text{cost}(r) - \text{cost}(r^1)) / \text{cost}(r^1)$$

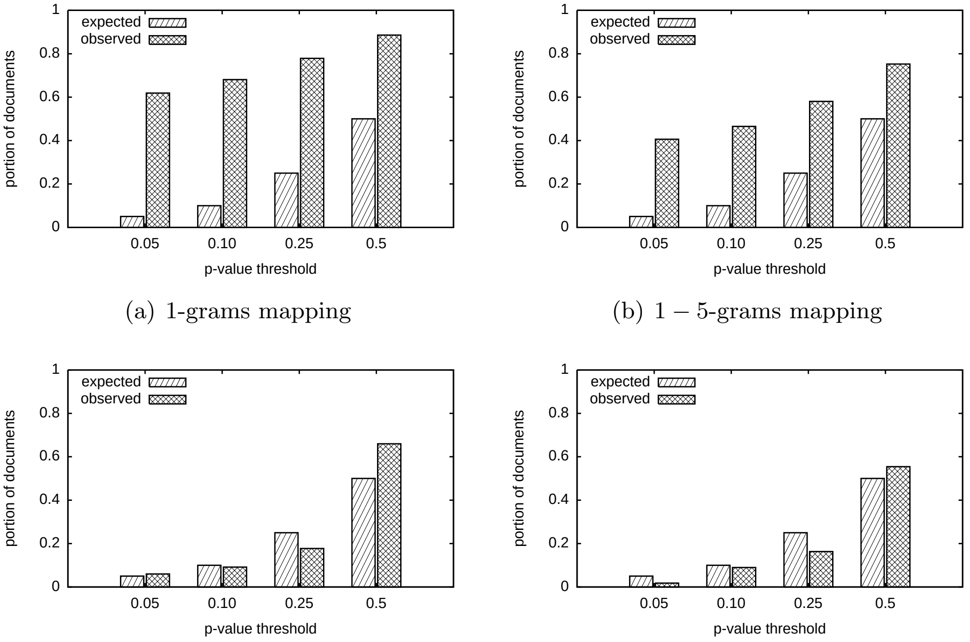
where:

- $\text{rank}(r)$: returns the rank of r within $D_k(q)$.
- $\text{cost}(r)$: a weighted sum of features (not normalized over the search space) in a log-linear model such as those mentioned in (Och, 2003).
- r^1: the highest ranked alternative in $D_k(q)$.

loss_{rank} provides a generally applicable criteria to select alternatives, penalizing selection from deep within $D_k(q)$. This estimate of the quality degradation does not reflect the generating model's opinion on relative quality. loss_{cost} considers the relative increase in the generating model's cost assigned to the alternative translation.

Gain: The $\text{gain}(r, D_k(q), f_w)$ function reflects the gain in the watermarking signal by selecting candidate r. We simply define the gain as the $P_n(X \geq \#(1, h(r)))$ from Equation 5.

[1] In the final version of this paper we will perform sampling to create a more reliable estimate of the false positive rate that is not overly influenced by document length distributions.



(a) 1-grams mapping

(b) $1-5$-grams mapping

(c) $3-5$-grams mapping

(d) Full result hashing

Figure 1: Comparison of expected false positive rates against observed false positive rates for different sub-result mappings.

4 Related Work

Using watermarks with the goal of transmitting a hidden message within images, video, audio and monolingual text media is common. For structured text content, linguistic approaches like (Chapman et al., 2001; Gupta et al., 2006) use language specific linguistic and semantic expansions to introduce hidden watermarks. These expansions provide alternative candidates within which messages can be encoded. Recent publications have extended this idea to machine translation, using multiple systems and expansions to generate alternative translations. (Stutsman et al., 2006) uses a hashing function to select alternatives that encode the hidden message in the lower order bits of the translation. In each of these approaches, the watermarker has control over the collection of results into which the watermark is to be embedded.

These approaches seek to embed a hidden message into a collection of results that is *selected* by the watermarker. In contrast, we address the condition where the input queries are not in the watermarker's control.

The goal is therefore to introduce the watermark into all generated results, with the goal of probabilistically identifying such outputs. Our approach is also task independent, avoiding the need for templates to generate additional alternatives. By addressing the problem directly within the search space of a dynamic programming algorithm, we have access to high quality alternatives with well defined models of quality loss. Finally, our approach is robust to local word editing. By using a sub-result mapping, we increase the level of editing required to obscure the watermark signal; at high levels of editing, the quality of the results themselves would be significantly degraded.

5 Experiments

We evaluate our watermarking approach applied to the outputs of statistical machine translation under the following experimental setup.

A repository of parallel (aligned source and target language) web documents is sampled to produce a large corpus on which to evaluate the watermarking *classification* performance. The corpora represent translations into 4 diverse target languages, using English as the source language. Each document in this corpus can be considered a collection of un-watermarked structured results, where source sentences are queries and each target sentence represents a structured result.

Using a state-of-the-art phrase-based statistical machine translation system (Och and Ney, 2004) trained on parallel documents identified by (Uszkoreit et al., 2010), we generate a set of 100 alternative translations for each source sentence. We apply the proposed watermarking approach, along with the proposed refinements that address task specific loss (Section 3.4) and robustness to edit operations (Section 3.3) to generate watermarked corpora.

Each method is controlled via a single parameter (like k or λ) which is varied to generate alternative watermarked collections. For each parameter value, we evaluate the Recall Rate and Quality Degradation with the goal of finding a setting that yields a high recall rate, minimal quality degradation. False positive rates are evaluated based on a fixed classification significance level of $\alpha = 0.05$. The false positive and recall rates are evaluated on the word level; a document that is misclassified or correctly identified contributes its length in words towards the error calculation. In this work, we use $\alpha = 0.05$ during classification corresponding to an expected 5% false positive rate. The false positive rate is a function of h and the significance level α and therefore constant across the parameter values k and λ.

We evaluate quality degradation on human translated test corpora that are more typical for machine translation evaluation. Each test corpus consists of 5000 source sentences randomly selected from the web and translated into each respective language.

We chose to evaluate quality on test corpora to ensure that degradations are not hidden by imperfectly matched web corpora and are consistent with the kind of results often reported for machine translation systems. As with the classification corpora, we create watermarked versions at each parameter value. For a given parameter value, we measure false positive and recall rates on the classification corpora and quality degradation on the evaluation corpora.

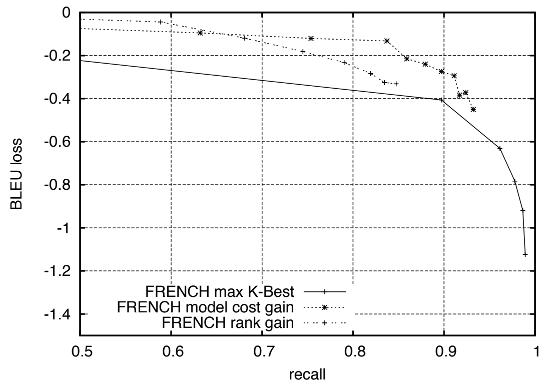


Figure 2: BLEU loss against recall of watermarked content for the baseline approach (max K-best), rank and cost interpolation.

Table 1 shows corpus statistics for the classification and test corpora and non-watermarked BLEU scores for each target language. All source texts are in English.

5.1 Loss Interpolated Experiments

Our first set of experiments demonstrates baseline performance using the watermarking criteria in Equation 5 versus the refinements suggested in Section 3.4 to mitigate quality degradation. The h function is computed on the full sentence result r with no sub-event mapping. The following methods are evaluated in Figure 2.

- Baseline method (labeled “max K-best”): selects r' purely based on gain in watermarking signal (Equation 5) and is parameterized by k: the number of alternatives considered for each result.
- Rank interpolation: incorporates rank into w, varying the interpolation parameter λ.
- Cost interpolation: incorporates cost into w, varying the interpolation parameter λ.

The observed false positive rate on the French classification corpora is 1.9%.

	Classification			Quality		
Target	**# words**	**# sentences**	**# documents**	**# words**	**# sentences**	**BLEU %**
Arabic	200107	15820	896	73592	5503	12.29
French	209540	18024	600	73592	5503	26.45
Hindi	183676	13244	1300	73409	5489	20.57
Turkish	171671	17155	1697	73347	5486	13.67

Table 1: Content statistics for classification and quality degradation corpora. Non-watermarked BLEU scores are reported for the quality corpora.

We consider 0.2% BLEU loss as a threshold for acceptable quality degradation. Each method is judged by its ability to achieve high recall below this quality degradation threshold.

Applying cost interpolation yields the best results in Figure 2, achieving a recall of 85% at 0.2% BLEU loss, while rank interpolation achieves a recall of 76%. The baseline approach of selecting the highest gain candidate within a depth of k candidates does not provide sufficient parameterization to yield low quality degradation. At $k = 2$, this method yields almost 90% recall, but with approximately 0.4% BLEU loss.

5.2 Robustness Experiments

In Section 5.2, we proposed mapping results into sub-events or features. We considered alternative feature mappings in Figure 1, finding that mapping sentence results into a collection of 3-5 grams yields acceptable false positive rates at varied levels of α.

Figure 3 presents results that compare moving from the result level hashing to the 3-5 gram sub-result mapping. We show the impact of the mapping on the baseline max K-best method as well as for cost interpolation. There are substantial reductions in recall rate at the 0.2% BLEU loss level when applying sub-result mappings in cases. The cost interpolation method recall drops from 85% to 77% when using the 3-5 grams event mapping. The observed false positive rate of the 3-5 gram mapping is 4.7%.

By using the 3-5 gram mapping, we expect to increase robustness against local word edit operations, but we have sacrificed recall rate due to the inherent distributional bias discussed in Section 3.3.

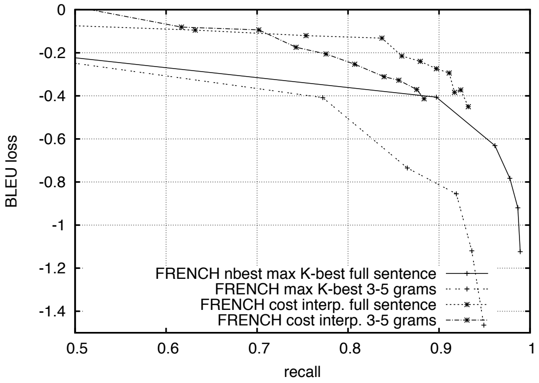


Figure 3: BLEU loss against recall of watermarked content for the baseline and cost interpolation methods using both result level and 3-5 gram mapped events.

5.3 Multilingual Experiments

The watermarking approach proposed here introduces no language specific watermarking operations and it is thus broadly applicable to translating into all languages. In Figure 4, we report results for the baseline and cost interpolation methods, considering both the result level and 3-5 gram mapping. We set $\alpha = 0.05$ and measure recall at 0.2% BLEU degradation for translation from English into Arabic, French, Hindi and Turkish. The observed false positive rates for full sentence hashing are: Arabic: 2.4%, French: 1.8%, Hindi: 5.6% and Turkish: 5.5%, while for the 3-5 gram mapping, they are: Arabic: 5.8%, French: 7.5%, Hindi:3.5% and Turkish: 6.2%. Underlying translation quality plays an important role in translation quality degradation when watermarking. Without a sub-result mapping, French (BLEU: 26.45%)

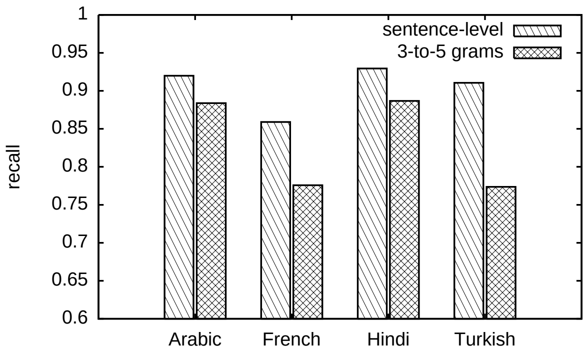


Figure 4: Loss of recall when using 3-5 gram mapping vs sentence level mapping for Arabic, French, Hindi and Turkish translations.

achieves recall of 85% at 0.2% BLEU loss, while the other languages achieve over 90% recall at the same BLEU loss threshold. Using a subresult mapping degrades quality for each language pair, but changes the relative performance. Turkish experiences the highest relative drop in recall, unlike French and Arabic, where results are relatively more robust to using sub-sentence mappings. This is likely a result of differences in n-gram distributions across these languages. The languages considered here all use space separated words. For languages that do not, like Chinese or Thai, our approach can be applied at the character level.

6 Conclusions

In this work we proposed a general method to watermark and probabilistically identify the structured outputs of machine learning algorithms. Our method provides probabilistic bounds on detection ability, analytic control on quality degradation and is robust to local editing operations. Our method is applicable to any task where structured outputs are generated with ambiguities or ties in the results. We applied this method to the outputs of statistical machine translation, evaluating each refinement to our approach with false positive and recall rates against BLEU score quality degradation.

Our results show that it is possible, across several language pairs, to achieve high recall rates (over 80%) with low false positive rates (between 5 and 8%) at minimal quality degradation (0.2% BLEU), while still allowing for local edit operations on the translated output. In future work we will continue to investigate methods to mitigate quality loss.

References

Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. 2007. Minimum error rate training in statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning.*

Peter F. Brown, Vincent J.Della Pietra, Stephen A. Della Pietra, and Robert. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19:263–311.

Mark Chapman, George Davida, and Marc Rennhardway. 2001. A practical and effective approach to large-scale automated linguistic steganography. In *Proceedings of the Information Security Conference.*

David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT).*

Jade Goldstein, Mark Kantrowitz, Vibhu Mittal, and Jaime Carbonell. 1999. Summarizing text documents: Sentence selection and evaluation metrics. In *Research and Development in Information Retrieval*, pages 121–128.

Gaurav Gupta, Josef Pieprzyk, and Hua Xiong Wang. 2006. An attack-localizing watermarking scheme for natural language documents. In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, ASIACCS '06, pages 157–165, New York, NY, USA. ACM.

Percy Liang, Alexandre Bouchard-Cote, Dan Klein, and Ben Taskar. 2006. An end-to-end discriminative approach to machine translation. In *Proceedings of the Joint International Conference on Computational Linguistics and Association of Computational Linguistics (COLING/ACL*, pages 761–768.

Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics.*

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 2003 Meeting of the Asssociation of Computational Linguistics*.

Philip Resnik and Noah A. Smith. 2003. The web as a parallel corpus. computational linguistics. *Computational Linguistics*.

Ryan Stutsman, Mikhail Atallah, Christian Grothoff, and Krista Grothoff. 2006. Lost in just the translation. In *Proceedings of the 2006 ACM Symposium on Applied Computing*.

Jakob Uszkoreit, Jay Ponte, Ashok Popat, and Moshe Dubiner. 2010. Large scale parallel document mining for machine translation. In *Proceedings of the 2010 COLING*.