

Learning Rules to Improve a Machine Translation System

David Kauchak & Charles Elkan
Department of Computer Science
University of California, San Diego
La Jolla, CA 92037
{dkauchak, elkan}@cs.ucsd.edu¹

Abstract. In this paper we show how to learn rules to improve the performance of a machine translation system. Given a system consisting of two translation functions (one from language A to language B and one from B to A), training text is translated from A to B and back again to A. Using these two translations, differences in knowledge between the two translation functions are identified, and rules are learned to improve the functions. Context-independent rules are learned where the information suggests only a single possible translation for a word. When there are multiple alternate translations for a word, a likelihood ratio test is used to identify words that co-occur with each case significantly. These words are then used as context in context-dependent rules. Applied on the Pan American Health Organization corpus of 20,084 sentences, the learned rules improve the understandability of the translation produced by the SDL International engine on 78% of sentences, with high precision.

1 Introduction

Machine translation systems are now commonplace. For example, they can be found for free on a number of web sites. If we treat these systems as black box translation engines where text is input and the translation obtained, can we improve the translation performance automatically?

Most previous research in machine translation has focused on developing systems from the ground up. Modern systems generally employ statistical and/or learning methods ([Melamed, 2001] and [Yamada and Knight, 2001]). A number of translation systems are offered commercially not only to businesses, but also to anyone with web access ([FreeTranslation, 2002] and [Systran, 2002]). These systems are either stand-alone translation engines or integrated into a general information processing system ([Damianos *et al.*, 2002]). Although these systems typically do not employ state of the art translation methods, they are widely used. In this paper, we examine these publicly available systems. The methods we describe work well on this type of system, but can also be employed on other machine translation systems.

¹ This material is based on work done for ORINCON Information Assurance sponsored by the United States Airforce and supported by the Air Force Research Laboratory under Contract F30602-02-C-0046.

The most common machine translation systems do word level translation. Although word level methods are the simplest, they have proved surprisingly successful. In an investigatory study in [Koehn and Knight, 2001], they show that 90% of the words in a corpus can be translated using a straightforward word for word translation. In this paper, we examine learning word level correction rules to improve machine translation systems. Rule learning approaches have proved successful in other natural language problems because they leverage statistical techniques and also tend to produce understandable and interpretable rules ([Brill, 1995]).

Most machine translation systems can translate in both directions between a language pair. Such a system can be thought of as two different functions, one that translates in one direction and a second that translates in the opposite direction. These functions are usually developed semi-independently and often the lexicon used by each is independent. This results in a difference in the knowledge built into each function. In this paper, we propose a method for automatically detecting this knowledge discrepancy and, using this information, for improving the translation functions. Given a word list in language A , we translate those words to language B and back again to language A . In some sense, the original word list defines a ground truth for the final set of translated words. Deviations from this ground truth point to cases where the system can be improved.

Using this setup, we describe how rules can be learned to improve these translation functions. Context-independent rules are learned where there is no ambiguity about the translation of a word. For words with multiple possible translations, a corpus is used to identify candidate context words and the likelihood ratio test is used to identify which of these context words co-occur significantly. Using these significant words, context-dependent rules are learned that disambiguate between ambiguous cases.

Using our method, 7,971 context-independent rules and 1,444 context-dependent rules are learned. These rules improve the understandability of the translation of 24,235 words and 78% of the sentences in the Pan American Health Organization corpus of over half a million words and 20,084 sentences.

2 Setup and Terminology

Before we explain the method for improving machine translation systems, we first define some terminology and assumptions. A machine translation system is a pair of translation functions (f, f') where L_1 and L_2 are natural languages and where f translates from L_1 to L_2 and vice versa for f' . We assume that we have unlimited access to the translation functions of a machine translation system, but not to the details of how the functions operate. We also assume that we have a large amount of text available in the languages that the machine translation system translates between. Finally, instead of trying to learn correction rules that change an entire sentence at once, we only learn rules that change a single word at a time.

In many situations, doing multiple single word changes leads to results similar to full sentence correction. Solving the single-word correction problem involves

three different steps. The first step is to identify where a word is being translated incorrectly. Given this incorrectly translated word, the second step is to identify the correct translation for that word. Finally, given an incorrect translation and the appropriate correct translation, the third step is to generate rules capable of making corrections in new sentences.

The first two steps can be seen as data generation steps. These steps generate examples that can then be used to generate rules in the third step using some supervised learning method. The three steps can be written as follows:

1. Find mistakes: Find word s_i in sentence $\bar{s} \in L$ with input context $c_1(s_i)$ where s_i is translated incorrectly to t_i with output context $c_2(t_i)$.
2. Find corrections: Find the correct translation, r_i , for s_i in $\bar{s} \in L$ with input context $c_1(s_i)$, output context $c_2(t_i)$ and incorrect translation t_i .
3. Learn correction rules: Generate a correction function g such that $g(s_i, \bar{c}_1(s_i), t_i, \bar{c}_2(t_i)) = r_i$ for each data sample i . A rule fires when s_i is in the input sentence with context $c_1(s_i)$ and s_i is translated to t_i with context $c_2(t_i)$ by the original machine translation system. Firing changes t_i to r_i .

The contexts described above can be any representation of the context of a word in a corpus, but we will use the bag of words of the sentence containing the word. Although this loses positional information, it is a simple representation that minimizes the parameters required for learning. Our goal is to improve a machine translation system, given the assumptions stated above, by solving each of the three problems described. The key to our approach is that given a sentence \bar{s} in a language, we can learn information from $f(\bar{s})$ and $f'(f(\bar{s}))$.

3 Analysis of Cases for an Example MT System

We examine one particular system and the application of the ideas above to improve this system. There are a number of commercial systems publicly available including [Systran, 2002] and [FreeTranslation, 2002]. Although Systran's system is more widely used, FreeTranslation offers more relaxed requirements on the length of the text to be translated. Also, initial comparison showed that results on AltaVista, which uses Systran's translation software, were comparable to the results obtained from FreeTranslation. Given a machine translation system, (f, f') , we calculate translations $f(w)$ and $f'(f(w))$ for a set of words w in L_1 . In our case, we choose $L_1 = \text{English}$ and $L_2 = \text{Spanish}$.

Table 1 shows a summary of the data generated using Freetranslation.com in February 2003 from 45,192 English words ([Red Hat, 2002]). A partition (i.e. non-overlapping, exhaustive set) of the possible outcomes is shown. We examine each of these cases and explain how each case provides information for improving the translation system (f, f') . For many machine translation systems, the default when the translation for w is unknown is to translate the word as w (i.e. $f(w) = w$). Throughout these examples, we will assume that equality implies that the system could not translate the word. A message or flag issued by the system could be used instead, if available.

	Occurrences	Example $w, f(w), f'(f(w))$
$w = f'(f(w)) \neq f(w)$	9,330	dog, perro, dog
$w = f(w) \neq f'(f(w))$	278	metro, metro, meter
$w \neq f(w) = f'(f(w))$	8,785	scroll, rollo, rollo
$w = f(w) = f'(f(w))$	11,586	abstractness, abstractness, abstractness
$w \neq f(w) \neq f'(f(w))$	14,523	cupful, taza, cup

Table 1: The results from doing a three-way translation of approximately 45,192 English words to Spanish and back to English.

$$w = f'(f(w)) \neq f(w) :$$

In this case, the word w is translated to a different string, $f(w)$, in the second language. When $f(w)$ is translated back to the original language, it is translated back to the original word w . Generally, in this situation the machine translation system is translating these words correctly. Mistakes can still occur here if there are complementary mistakes in the lexicon in each direction. There is no information for the three problems described above in this case.

$$w = f(w) \neq f'(f(w)) :$$

In this case, the word w is translated to the same string in the second language; however, it is then translated to a different string when it is translated back to the original language. This happens when w is a word in both languages (possibly with different meanings), which the translation system is unable to translate to the second language (for example, $w = \text{arena}$, $f(w) = \text{arena}$, $f'(f(w)) = \text{sand}$). From these examples, we learn that the translation function f should translate $f'(f(w))$ to $f(w)$ (Problem 2). This information may or may not be useful. We can query f to see if this information is already known.

$$w \neq f(w) = f'(f(w)) :$$

In this case, the word w is translated from the original language to the second language; however, it is then translated as the same word when translated back to the original language. There are two cases where this happens.

1. The most likely situation is that there is a problem with the translation system from the second language to the original language (i.e. in f') since the default behavior for translating an unknown word is to leave the word untranslated. In this case, two pieces of information are learned. First, if $f(w)$ is seen on the input and is translated to $f'(f(w))$ then a mistake has occurred (Problem 1). We can also suggest the correct translation. Given a sentence \bar{s} , if word s_i is translated to s_i and $s_i = f'(f(w))$, then s_i was incorrectly translated and the correct translation for s_i is w (Problem 2).
2. The second case, which is less likely, is that $f(w)$ is a word that, when translated back to the original language, is the same string (this is similar to case 2 below of $w = f(w) = f'(f(w))$). For example, $w = \text{abase}$, $f(w) = \text{de}$

grade (present subjunctive form of degradar, to degrade), $f'(f(w)) = \text{degrade}$. We can learn that $f(w)$ is an ambiguous word that can be translated as either w or $f'(f(w))$.

$w = f(w) = f'(f(w))$:

In this case, all the words are the same. There are two common situations.

1. If the word for w in the second language is actually w then the translation is correct. This is common with proper names (for example, $w = \text{Madrid}$, $f(w) = \text{Madrid}$, $f'(f(w)) = \text{Madrid}$). In this case, no information is gained to solve the problems listed above.
2. If the system is unable to translate w , then $w = f(w)$. If this is the case, then it is unlikely that w will actually be a valid word in the second language (as shown above, this does happen 278 out of 45,192 times, where the $f(w)$ is translated to something different by f') and so the word again gets translated as w in the second translation step (for example, $w = \text{matriarchal}$, $f(w) = \text{matriarchal}$, $f'(f(w)) = \text{matriarchal}$). In this case, the translation function f makes a mistake on word w (Problem 1).

$w \neq f(w) \neq f'(f(w)) \neq w$:

There are two situations that may cause this to happen. w may be a synonym for $f'(f(w))$ or there may be at least one error in the translation. If we assume that the knowledge in the translation systems is accurate, then both w and $f'(f(w))$ are appropriate translations for $f(w)$. These two cases can be disambiguated using contextual information.

One last piece of information can be obtained when $f(w) \neq f'(f(w))$. In these cases, some translation was done by f' . We can assume that if $f'(f(w))$ actually is a word in the original language. Using this, we can extend the word list in the original language.

4 Rule Learning

Using the framework described in Section 3, we can learn rules that improve the output of a translation system. We learn two different types of rules: context-independent and context-dependent. If there is no ambiguity about the translation of a word, then context is not required to disambiguate and a context-independent rule can be learned. If, on the other hand, there are multiple possible translations, then context is required to decide between the different possible translations. Figure 1 outlines the algorithm for generating the data and for learning both types of rules.

For preprocessing, the word lists is translated from the starting language to the alternate language and back to the original language. Table 2 summarizes the information that is used for generating rules from these translations. The input words are L_j words. The current translations are the words expected to be seen in the output of the translation system. Finally, the correct translations indicate which word the output word should be changed to.

By examining the input words involved in the cases in Table 2, non-ambiguous words can be identified where an input word only has one learned correct translation. Notice that many of the entries in Table 2 are inherently ambiguous, such

Preprocessing steps

- Translate L_1 word list from L_1 to L_2 and back to L_1
- Translate L_2 word list from L_2 to L_1 and back to L_2
- Generate input word (L_2), current translation (L_1) and correct translation (L_2) triplets using rules in Table 2
- For all words, w , in corpus, generate frequency counts, $count(w)$
- Translate corpus from L_1 to L_2 to use for learning contexts

Generate context-independent rules for non-ambiguous words

- Identify non-ambiguous words by finding all “input words” with only a single suggested correct translation
- Generate context-independent rules of the form:
$$g(\text{input word}, [], \text{current translation}, []) \rightarrow \text{correct translation}$$

Generate context-independent rules for k -dominant words

- Find sets of “input words” that have the same suggested correction translation. These words represent possible translation options. Identify k -dominant words where
 $count(option_i) > k$ and $count(option_j) = 0$ for all $j \neq i$
- Generate context-independent rules of the form:
$$g(\text{input word}, [], \text{current translation}, []) \rightarrow option_i$$

Generate context-dependent rules for ambiguous words

- Get the possible context words t_j for each $option_i$ for the remaining ambiguous words
 - In the L_1 corpus, find sentences where $option_i$ appears and the corresponding ambiguous word is in the translated sentence in L_2
 - Get all possible context words t_j as the words surrounding $option_i$
- For each $option_i$, generate the context, $c(option_i)$, as all t_j that pass the significance level α threshold for the likelihood ratio test
- Learn context-dependent rules of the form:
$$g(\text{input word}, [], \text{current translation}, c(option_i)) \rightarrow option_i$$

Figure 1: Outline of algorithm to learn rules to improve L_2 to L_1 translation. The preprocessing steps generate the initial data for use in learning the rules. The following three sets of steps describe the algorithms for learning the context-independent and context-dependent rules.

as when $w \neq f(w) \neq f'(f(w))$. Almost all non-ambiguous words are generated from the case when $w \neq f(w) = f'(f(w))$, where the system knows how to translate $f(w)$ from English but does not know how to translate it back to English.

For those words where there is only one known translation and therefore no ambiguity, a context-independent rule of the form $g(s, [], t, []) = r$ can be learned, where s = input word, t = current translation and r = correct translation. Using this methodology, 7,155 context-independent rules are learned from the list of 45,192 words and the FreeTranslation engine.

4.1 Dealing With Ambiguous Words

For the remaining input word, current translation and correct translation triplets, there are at least two correct translations for the same input word. We must decide between these possible correct translations. We suggest two methods that both leverage a corpus in the target language, in this case English, to distinguish

Case	Input word	Current translation	Correct translation
Eng Sp Eng $w \neq f(w) = f'(f(w))$ $f(w)$ is not an English word	$f(w)$	$f'(f(w))$	w
Eng Sp Eng $w \neq f(w) = f'(f(w))$ $f(w)$ is an English word	$f(w)$ $f(w)$	$f'(f(w))$ $f'(f(w))$	w $f'(f(w))$
Eng Sp Eng $w \neq f(w) \neq f'(f(w))$	$f(w)$ $f(w)$	$f'(f(w))$ $f'(f(w))$	w $f'(f(w))$
Sp Eng Sp $w = f(w) \neq f'(f(w))$ $f'(f(w)) = f(f'(f(w)))$	$f'(f(w))$	$f(f'(f(w)))$	$f(w)$
Sp Eng Sp $w = f(w) \neq f'(f(w))$ $f'(f(w)) \neq f(f'(f(w))) \neq f(w)$	$f'(f(w))$ $f'(f(w))$	$f(f'(f(w)))$ $f(f'(f(w)))$	$f(w)$ $f(f'(f(w)))$

Table 2: Patterns for generating rules for Spanish to English improvement

between translation options.

We would like to identify as many non-ambiguous words in the data as possible, since these rules are simpler. To do this, we can use the English corpus available. For our purposes, we use the Pan American Health Organization corpus ([PAHO, 2002]) that consists of over half a million words. Counting the occurrences of the possible translations (i.e. correct translation entries) can give some indication about which translation options are more likely. We define an input word as being k -dominant if one translation option occurs at least k times in the text and all other options do not appear at all. When a word is k -dominant, it is reasonable to assume that the input word should always be translated as the dominant option. We can learn a context-independent rule that states exactly this. Using this method, all of the k -dominant words with $k = 5$ are learned resulting in an additional 816 context-independent rules.

For all the input words where there are multiple possible translations and no one option is k -dominant, context can be used to disambiguate between the possible translations. The rules being learned have the possibility of both an input context and an output context. In practice only context in the input *or* output language is necessary. In our case, for Spanish to English improvement, English text is more readily available, so only the output contexts will be learned.

Given an ambiguous input word, a , that has $option_1, \dots, option_n$ as possible correct translations, the goal is to learn a context for each possible translation, $option_i$, that disambiguates it from the other translations. We do this by gathering words from the English corpus that occur in the same sentences as each of the possible translation options $option_1, \dots, option_n$. We can use the machine translation system to verify that $option_i$ actually gets translated to a (and correspondingly that a gets translated to $option_i$) in that context.

4.2 Determining Significant Context Words

The problem described above is the problem of collocation: finding words that are strongly associated. Many methods have been proposed for discovering col-

locations such as frequency counts, mean and variance tests, t -test, χ^2 test and likelihood ratio test ([Manning and Schütze, 1999]). The likelihood ratio test has been suggested as the most appropriate for this problem since it does not assume a normal distribution like the t -test nor does it make assumptions about the minimum frequency counts like the χ^2 test ([Dunning, 1993]).

For this problem, we have two different sets of sentences that we are interested in: the set S_i of sentences that contain the translation option t_i , and the set \bar{S}_i of sentences that don't contain the translation option. The decision is for each context word w_j in the sentences belonging to S_i , whether or not that word is significantly associated with the translation option t_i or not.

The likelihood ratio test tests an alternate hypothesis against a null hypothesis. In this case, the null hypothesis is that the two groups of sentences (sentences with and without t_i) come from the same distribution with respect to the occurrence of w_j in the sentence. The alternate hypothesis is that the two groups of sentences are different with respect to the occurrence of w_j . We will also impose the further constraint that w_j must be *more* likely to occur in sentences of S_i .

For each set of sentences, the occurrence of w_j can be modeled using the binomial distribution. The assumption is that there is some probability that w_j occurs in a sentence. For both hypotheses, the likelihood equation is $\ell = p(S_i; \theta_1) p(\bar{S}_i; \theta_2)$. For the null hypothesis, the sentences come from the same distribution and therefore $\theta_1 = \theta_2 = \theta$. In all these situations, the maximum likelihood estimate of the parameter is used (the frequencies seen in the training data, in this case the English corpus). Using these parameter estimations, the likelihood ratio can be calculated in a similar fashion to [Dunning, 1993]. We compare this value with a significance level, α , to make a decision about whether the collocation is significant or not. We do this for all words in sentences that contain t_i , then construct context-dependent rules that contain all words that pass the significance test in the context. For our experiments, $\alpha = 0.001$ is an appropriate significance level. Intuitively, this means that there is a one in a thousand chance of a candidate word being misclassified as significant.

To improve the generality of the contexts learned, we perform the test on stemmed versions of the words and generate context-dependent rules using these stemmed words. The Porter stemmer ([Porter, 1980]) is used to stem the words. For the remainder of the paper, the results provided are for the stemmed versions.

5 Results

In this section, we examine the success of the learned rules in a real domain. We examine the Pan American Health Organization (PAHO) Conferences and General Services Division parallel texts. This data set consists of 180 pairs of documents in English and Spanish ([PAHO, 2002]).

The 180 Spanish documents consist of 20,084 sentences, identified by periods (minus a number of Spanish abbreviations), and 616,109 words, identified by surrounding white space. The sentences are translated using FreeTranslation.com to get the initial translation. Then, the rules learned using the algorithms in Section 4 are applied to change the sentences. For the context-independent rules, the rule fires anytime the appropriate words are seen in the original sentence and translated sentence. The context-dependent rules add the additional restriction

Spanish:

El contenido de **alquitrán** en los cigarrillos de tabaco negro sin filtro es mayor que en los restantes tipos de cigarrillos y son aquellos precisamente los de mayor consumo en la población, lo que aumenta la potencialidad del tabaquismo como factor de riesgo.

Original translation:

The content of **alquitrán** in the black cigarettes of tobacco without filter is greater than in the remaining types of cigarettes and are those precise the of greater consumption in the population, what enlarges the potencialidad of the tabaquismo as factor of risk.

Improved translation:

The content of **tar** in the black cigarettes of tobacco without filter is greater than in the remaining types of cigarettes and are those precise the of greater consumption in the population, what enlarges the potencialidad of the tabaquismo as factor of risk.

Figure 2: Example of an improvement in translation. The first sentence is the original Spanish sentence to be translated. The second sentence is the translation made by FreeTranslation.com. The final sentence is the translation after a learned improvement rule has been applied. The change is in bold.

that the translated sentence must also contain one of the words in the learned output context of the rule to fire.

Over 9,000 rules are learned. Table 3 shows the results from applying these rules to the sentences. The rules change 22,206 words in the PAHO data set and 14,952 or 74% of the sentences. Figure 2 shows an example firing of a context-independent rule that changes “alquitrán” to “tar”.

Rule type	Rules learned	Avg. # words in context	Rules used	Words changed
Context independent	6,783	NA	701	5,022
Context independent, dominant $k = 5$	809	NA	191	4,768
Context dependent, $\alpha = .001$	1,355	5	301	12,416

Table 3: Summary of results for rules generated from a word list with 45,192 entries.

6 Using Extended Word Lists

The methods in this paper are based on having a word list in language L_1 . In this section, we present two methods for extending this word list. One of the advantages of the rule learning method described above is that it is robust to erroneous words in the word list. If the system does not recognize a word in the word list then it will not get translated, as is the case where $w = f(w) = f'(f(w))$. No learning is done in this case, so erroneous words are filtered out by the machine translation system. Since a high quality word list is not required, the word lists can be constructed at least two different ways.

When translating w to $f(w)$ and back to the original language as $f'(f(w))$, if

$f(w) \neq f'(f(w))$ then some translation was done between $f(w)$ and $f'(f(w))$. Given the robustness of the learning system, we can assume that if the machine translation system translates $f(w)$ to $f'(f(w))$, then $f'(f(w))$ is a word in the original language. Using this method, 419 additional words not in the original word list are learned.

In many circumstances, translation systems are to be used in a specific domain (for example medicine, politics, public health, etc.). The PAHO data set mentioned earlier contains documents in the public health domain. To improve the recall of the machine translation we can incorporate more rules that contain terminology that is relevant to this particular domain. We can do this by examining words in a corpus of a similar domain to add to the word list. In our case, since the PAHO data set contains the parallel text in English, we can use this text. The English version of the PAHO data set contains 5,215 new words that are not in the original word list.

Table 4 shows the results of learning rules with the original 45,192 words plus the 419 learned words and the 5,215 domain specific words. The additional words add 468 new rules. Although these new rules only constitute a small fraction of the total rules (~5%) they account for over 8% of the changes. In particular, the new, domain specific context-independent rules fire over four times more often than the rules learned from a generic word list. Because these additional rules are learned using domain specific words, they are much more likely to apply for translating text in that particular domain. With the addition of these new rules, 78% of the sentences are changed.

Rule type	Rules learned	Avg. # words in context	Rules used	Words changed
Context independent	7,155	NA	903	6,526
Context independent, dominant $k = 5$	816	NA	200	5,038
Context dependent, $\alpha = .001$	1,444	5	327	12,671

Table 4: Summary of results for rules generated using a general word list with 45,000 entries plus 419 learned words and 5,215 domain specific words.

7 Discussion

In this paper, we have examined a technique for improving a machine translation system using only plain text. One of the advantages of this approach is that the resources required to learn the rules are easier to obtain than traditional approaches that require aligned bitext ([Macklovitch and Hannan, 1996]). Also, our method makes no assumptions about the workings of the translation system.

By translating words from the original language to the second language and back to the original language, differences in information between the two translation functions are isolated. Using this information, we show how correction rules can be learned. Context-independent rules are learned when the system only suggests a single possible translation. When there is ambiguity about what the correct translation of a word is, the likelihood ratio is used to identify words that co-occur significantly with each translation option.

Using these rules, almost 25,000 words are changed on a corpus of over half a million words. On a sample of 600 changes, the context-independent rules have a

precision of 99% and the context-dependent rules have a precision of 79%. One of the open questions for machine translation research is how to evaluate a translation. A few automated methods have been suggested such as BLEU ([Papineni *et al.*, 2001]), which is based on n -gram occurrence in a reference text. Although these methods have merit, for the particular rules learned by our system, an n -gram metric would almost always see improvement since changing a Spanish word in English text to an English word will generally be better. For this reason, we instead chose to evaluate the results by hand.

A majority of the context-independent rules represent changes where the original system did not know any possible translation, so it is not surprising that the precision is high. The context-dependent rules have lower precision even though a significance level of 0.001 was used. The main reason for this lower precision is that the likelihood ratio can suggest collocations that are significant, but that are not useful for ambiguity resolution. This is attenuated when the counts are very small or when the ambiguous translation is common and the counts are therefore high. In these cases, common words such as “is”, “that”, “it”, “have”, etc. are identified as significant.

Another problem is the particular rule representation chosen. The context-dependent rules define the context as a bag of words. Unfortunately, a bag of words does not model many relationships, such as word order, syntax or semantics, which can be useful for discriminating significant collocations. For example, when deciding between “another” and “other” in the sentence fragment “Another important coordination type...”, the location of “type” and the fact that it is singular suggests “another” as the correct translation.

One final problem is that stemming can cause undesired side effects in the contexts learned. As seen in the sentence fragment above, plurality is important, particularly when deciding between two translations that only differ by plurality. Unfortunately, stemming, in attempting to improve generality, removes the plurality of a word. The combination of these problems leads to a lower precision for the context-dependent rules. Future research should be directed towards employing alternate rule representations and alternate collocation techniques such as in [Krenn, 2000].

The techniques that we used in this paper are just the beginning of a wide range of improvements and applications that use existing machine translation systems as a resource. As new applications that use translation systems arise, particularly those in time and information critical fields, such as [Damianos *et al.*, 2002], the importance of accurate automated translation systems becomes critical.

References

- [Brill, 1995] Brill, E. 1995. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. *Computational Linguistics* 21(4), p. 543-565.
- [Damianos *et al.*, 2002] Damianos, L., Ponte, J., Wohlever, S., Reeder, F., Day, D., Wilson, G. and Hirschman, L. 2002. MITAP for Biosecurity: A Case Study. In *AI Magazine*, Winter 2002, p. 13-29.
- [Dunning, 1993] Dunning, T. 1993. Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics* 19(1), p. 61-74.

- [FreeTranslation.com, 2002] <http://www.freetranslation.com> powered by SDL International's *Enterprise Translation Server*.
- [Koehn and Knight, 2001] Koehn, P. and Knight, K. 2001. Knowledge Sources for Word-Level Translation Models. In *Empirical Methods in Natural Language Processing conference*.
- [Krenn, 2000] Krenn, B. 2000. Collocation Mining: Exploiting Corpora for Collocation Identification and Representation. In *Proceedings of The Ninth EURALEX International Congress*.
- [Melamed, 2001] Melamed, D. 2001. *Empirical Methods for Exploiting Parallel Texts*. The MIT Press.
- [Macklovitch and Hannan, 1996] Macklovitch, E. and Hannan, M. 1996. Line'Em Up: Advances In Alignment Technology And Their Impact on Translation Support Tools. In *Proceedings of the Second Conference of the Association for Machine Translation in the Americas*, p. 41-57.
- [Manning and Schütze , 1999] Manning, C. and Shütze, H. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.
- [PAHO, 2002] Pan American Health Organization documents: <http://crl.nmsu.edu/cgi-bin/Tools/CLR/clrcat#H8>
- [Papineni *et al.*, 2001] Papineni, K.A., Roukos, S., Ward, T. and Zhu, W.J. 2001. Bleu: A Method for Automatic Evaluation of Machine Translation. IBM Research Report, RC22176.
- [Porter, 1998] Porter, M. 1980. An Algorithm for Suffix Stripping. *Program (Automated Library and Information Systems)*, 14(3), p. 130-137.
- [Red Hat, 2002] Linux, Red Hat 7.2, English word list /usr/dict/words.
- [Systran, 2002] Systran Corporation (portals: Google, AOL, AltaVista, CompuServe, Lycos, OracleMobile.com).
- [Yamada and Knight, 2001] Yamada, K. and Knight, K. 2001. A Syntax-based Statistical Translation Model. In *Proceedings of the Association for Computational Linguistics*, p. 523-530.
- [Yarowsky, 1994] Yarowsky, D. 1994. Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish And French. In *Proceedings of the Association for Computational Linguistics*, p. 77-95.