2013

Language Translation for File Paths

Rowe, Neil C.; Schwamm, Riqui; Garfinkel, Simson L.

Monterey, California. Naval Postgraduate School

DFRWS

Language Translation for File Paths

Neil C. Rowe*, Riqui Schwamm, and Simson L. Garfinkel

U.S. Naval Postgraduate School, Monterey,California 93943, USA

Abstract

Forensic examiners are frequently confronted with content in languages that they do not understand, and they could benefit from machine translation into their native language. But automated translation of file paths is a difficult problem because of the minimal context for translation and the frequent mixing of multiple languages within a path. This work developed a prototype implementation of a file-path translator that first identifies the language for each directory segment of a path, and then translates to English those that are not already English nor artificial words. Brown's LA-Strings utility for language identification was tried, but its performance was found inadequate on short strings and it was supplemented with clues from dictionary lookup, Unicode character distributions for languages, country of origin, and language-related keywords. To provide better data for language inference, words used in each directory over a large corpus were aggregated for analysis. The resulting directory-language probabilities were combined with those for each path segment from dictionary lookup and character-type distributions to infer the segment's most likely language. Tests were done on a corpus of 50.1 million file paths looking for 35 different languages. Tests showed 90.4% accuracy on identifying languages of directories and 93.7% accuracy on identifying languages of directory/file segments of file paths, even after excluding 44.4% of the paths as obviously English or untranslatable. Two of seven proposed language clues were shown to impair directory-language identification. Experiments also compared three translation methods: the Systran translation tool, Google Translate, and word-for-word substitution using dictionaries. Google Translate usually performed the best, but all still made errors with European languages and a significant number of errors with Arabic and Chinese.
This paper appeared in the Proceedings of the 2013 DFRWS Conference.

Keywords: digital forensics; file paths; machine translation; dictionary; character distribution; Unicode; Naive Bayes inference

1. Introduction

Forensic examiners increasingly work with materials in unfamiliar human languages. Although some examiners have human linguists available to translate text, audio, and video into their native tongues, most do not. Human translation is expensive and not always timely (U.S. FBI, 2004). Automated translation of directories, file names, and other metadata could be a useful first step in an investigation. Computer users rely on named directories to organize their information, and they give their files descriptive names. Translation of them could enable recognizing similar activities or otherwise interesting behavior taking place in different linguistic parts of the world, and aid cross-language clustering of files.

This issue is important with our research data, the Real Drive Corpus. It currently contains more than 3000 drive images from 28 wide-ranging countries. It contains a wide range of languages, and not just the languages one would expect. For instance, drives from Israel contained significant amounts of Spanish and Chinese, while drives from United Arab Emirates contained significant amounts of French.

Machine translation of forensic file paths need not be perfect to be useful since many investigations focus on keyword lookup. In fact, merely identifying the languages of a file name without translating the words may help an investigation, as file contents are almost always in the same language as a file name, and file-name analysis could suggest what translators to call for the contents.

This work only addresses translating to English. This is the easiest target language since many operating-system file and directory names are in English. Nonetheless, there are many challenges with the remaining words.

1.1. Prior work

Prior work shows that drives can be characterized in many important ways by their metadata alone (Rowe and Garfinkel 2012), but file paths in an unintelligible language can still impede investigation.

Machine translation has a long history (Wilks 2009). The major approaches are case-based reasoning as in the Systran system, and statistical inference as in IBM's Candide system of the 1990s and its many descendants including Google Translate. Current systems are far from perfect; a figure of 50% accuracy on prose is often cited. But file paths use a limited language and translation success rates could be higher. Good success has been shown for instance for the constrained domain of news stories (Turchi et al 2012).

Language identification is a key subproblem. Most prior work on it has focused on N-grams as in (Mishra et al, 2010) and LA-Strings (Brown 2011), though word clues (Yang and Liang, 2010) and other mathematical techniques (Da Silva and Lopes, 2006) have been used. Language-detection products include Google's Compact Language Detector (McCandless 2011), and Shuyo's Language Detection Library for Java (Shuyo 2010).

Basis Technology has demonstrated the Odyssey Digital Forensics Search system (Basis 2013) which combines the company's multilingual named entity extraction technology with a search capability allowing a user to enter English words and search for their foreign-language equivalents, but it neither translates nor transliterates paths.

2. Making sense of mixed-language paths

Our approach is to obtain paths by first using SleuthKit to extract drive images, and then Fiwalk (now part of SleuthKit) to extract file metadata including file paths. SleuthKit's `tsk_fs_dir_walk` reads UCS-2 filenames stored in the FAT32 and NTFS directory entries, and recodes them as UTF-8 sequences. FAT12, FAT16 and FAT32 file systems use OEM character sets and Code Pages to store short filenames which SleuthKit uses to convert to UTF-8 Unicode. In a significant number of cases, SleuthKit did not produce either valid Unicode code points or the shortest possible encoding. We checked SleuthKit file names character-by-character and replaced invalid bytes with Python-style escape sequences. For example, the UCS-2 byte sequence "0xFF 0xFF" would be encoded as "\xFF\xFF", as U+FFFF is not a valid Unicode character.

Multiple tools for language handling are needed because the problem of translating file paths is difficult. Most translation tools are designed for large blocks of prose and take advantage of punctuation and syntactic rules. File paths use considerable but nontraditional punctuation, use little conventional syntax, and use frequent abbreviations and code words. Context is important in translation (Larson, 1984), so a big challenge is understanding the context of a word, but this is frequently unclear in paths. A path often contains different kinds of information in different places, and when multiple languages are used, they are rarely consistent through the path. 30.4% of the file paths in our corpus change language once in their sequence, and 23.1% change at least twice, ignoring untranslatable words. Consider these examples from our corpus:

- Documents and Settings/defaultuser/ Mes documents/Ma musique/Desktop.ini
- Mis Documentos/SalvadorJP/Excel/ GRUPOS.xls
- Documents and Settings/3742008/ Configuración local/Datos de programa/ Microsoft/Internet Explorer/.
- human/animation/weapon_pistol/ major_pain/몇 읪 킵읗궯^"A 鍁瞔/ pistol_pain_crawldeath.skc

2.1. Collecting word sequences for translation

The Systran and Google Translate software return input words unchanged if they cannot translate them. Thus our first attempt was to send Systran the entire path when languages are mixed, since most words in our corpus were English. This ran into trouble with words having different meanings in different languages. For instance, Systran translates "Temporary Internet Files" occurring in a Mexican file path into English as "Temporary Internet you case out" because "files" is the present subjunctive second person of the verb "filar" meaning "case out". Here the fact that these are all known English words should overrule an attempt to translate the phrase from Spanish. "Temporary Internet Files" appears frequently on our Mexican drives, and "files" is the most common English word in our corpus, so translating paths as a whole does not work well.

So it is important to handle each directory and file name in a path separately. We thus first segment directories, then segment at each punctuation mark or digit. The current Unicode specifications list 1216 punctuation marks (Unicode, 2013). The few words that contain punctuation (like "tutti-frutti") can be ignored without causing much trouble, except for apostrophe-s in English which is handled separately. The remaining characters can be translated and inserted back into the path to get a translated path.

2.2. Directory word aggregation

In our tests the LA-Strings language identifier demonstrated mediocre performance on short strings, apparently as a result of its inability to extract sufficient statistics. For instance, for the software terms "obj viewsspt viewssrc vs lk", LA-Strings thought the most likely language was Latvian, and for "cmap enutxt", Southern Dong. This problem could be reduced by aggregating the words of each

directory over the corpus. In this it was important to keep separate word lists by country of origin of each drive, since for example filenames under "My Documents" in China tended to be quite different than those in the same directory in Egypt. Word extraction ignored file extensions, which are untranslatable international codes. A single string was created for each directory, preserving order within paths but deleting duplicate segments. For instance, WINNT/Profiles/adrian/Menú Inicio/ Programas/Accesorios/Multimedia on Mexican drives contained “Control de volumen.lnk”, “Grabadora de sonidos.lnk”, “Reproductor de CD.lnk”, and “Reproductor de medios.lnk”, resulting in the string “Control de volumen Grabadora de sonidos Reproductor de CD Reproductor de medios”. This process is used for subdirectory names too.

78.2% of the words in the corpus were known English, so many paths contained only English words. The policy was that a directory could be excluded from translation if its word list did not contain at least one word not in English of at least three letters. This excluded 44.4% (853169/1920259) of the directories in our main corpus, including many directories with hexadecimal-character filenames.

2.3. Transliteration

Keyboard support for languages is inconsistent in the world. Thus we saw many instances of transliteration to a Roman alphabet, and in most of these cases, to ASCII characters. For instance, we saw “configuracion” rather than the proper Spanish “configuración” on many Mexican drives (both different from the English equivalent “configuration”). Thus transliterated dictionaries were needed. Although commercial transliteration systems exist (e.g. the Basis Rosette Language Platform), it is straightforward to create a transliteration table for European languages and map dictionaries with it. 18 languages of our 35 were chosen for transliteration. Appropriate languages are those with unambiguous mappings, like Spanish for which transliteration was especially useful. Arabic did not work well because some characters have multiple possible equivalents, and Chinese transliteration was too complex to implement.

3. Inference of the language of a directory

To identify the language of a directory, seven clues were evaluated, none sufficiently reliable individually. This was because path names are so short and because abbreviations, misspellings, nonstandard compound words, and specialized software code names are common. Our approach was to create an ensemble language identification technique specialized for pathnames. The seven clues were: the languages identified by the LA-Strings software, the number of matches to dictionary word lists for each of the 53 languages, the distribution of character types, the distribution of raw characters, the country of origin of the drive, the use of language-identifying keywords in the file path, and the inferred language possibilities for the directory above in the file hierarchy.

3.1. Using LA-Strings

We used a version of LA-Strings from September 2012 that claimed to identify 1032 languages and 3306 language/encoding pairs. LA-Strings tries to guess the language using lexical features such as character bigrams. As mentioned, we found it making many identifications of obscure languages because of insufficient data; we improved its performance by restricting it to the 100 most popular languages. Even then, it guessed many languages inconsistent with the origins of our corpus. For example, song titles on Indian drives that it was quite certain were the African language Hausa were confirmed to be transliterated Hindi.

LA-Strings returns language possibilities in decreasing order, and returns a percentage for the most-likely possibility which we interpreted as a likelihood. Since Zipf’s Law, $f(k) = C/k$, tends to be a good default model for frequency distributions sorted in decreasing order, we assumed the second language was half as likely as the first language, the third language was one third as likely, and so on. Five language possibilities seemed to suffice for nearly all cases. LA-Strings also failed to return output for some strings like "faxcn" and "ETUP", and returned only a percentage with no language for others like "162122"; we interpreted these as untranslatable text, our special language "un".

LA-Strings language names are not always consistent with the ISO-639 standard, so we had to figure mappings onto the two-letter codes of the standard. LA-Strings distinguishes several dialects, most notably for Chinese and Spanish. When Systran had only one translator for these dialects, we mapped them onto a single language name.

3.2. Using dictionary information

Another clue to the language of a directory is the appearance of its words in dictionaries. Our current dictionaries cover 1,105,778 distinct words. The English word list has 310,051 entries, using lists from our previous research plus sources at wordlist.sourceforge.net including the 1991 “Public Brand Software” list of 109,582 words. It excludes foreign-language words like “trattoria” from the English word list when the word is not primarily English. However, it does include many proper nouns such as person names, geographical names, names of businesses, names of commercial products (like “winzip” and “imac”), as well as specialized computer terms if they originated from English-speaking countries. It also includes a wide range of abbreviations and acronyms that occurred frequently in our corpus; these had to be manually confirmed. It does not include misspellings unless they have

become conventional. For borderline cases of place names of foreign origin used conventionally in English, the criterion was whether an English speaker would use it more than a foreign-language speaker; so “rome” was included in the English word list but not “roma” and “bahia” (which means “bay” in Portuguese although it is also the name of a city). All entries are stored in lower case because capitalization is inconsistent in cyberspace; older systems often have all uppercase, and it is rare to find English proper nouns correctly capitalized.

We also built word lists for 34 other languages and 18 transliterated languages from Wiktionary (Buchmeier 2013) and Google Translate. Altogether 75.2% of the 1.11 million words occurred in only one language after removing common borrowed English words like “Internet” from non-English lists, so most dictionary words are unambiguous as language indicators – it is the unknown software and hardware terms in our corpus that create challenges. Words that do not occur in any dictionary list and are not inferred compounds are assigned to language “un”.

The Wikitionary focuses on general-usage foreign-language words rather than terms that appear in computers and devices. For words more relevant to our task, we created word-translation pairs by applying Google Translate to the 32,015 English words occurring in our corpus path names at least 10 times. We did this for each language of our corpus, and excluded results equal to the input as well as other spurious results that indicated translation failure. This worked so well we also processed some other unknown words in the corpus with Google Translate. There were two kinds: 1.429,380 distinct words not in any word list, which were mostly untranslatable, and 953,839 words identified with a particular language by our inference methods but not listed in that language’s dictionary (e.g. plurals and verb forms). Unknown words that occurred more than 200 times in the corpus were sorted into files for each language based on their character distributions for the first kind of words (using methods to be described), and their identified language for the second kind of words. This sorting was reliable for Chinese and Arabic with their distinctive characters, and saved considerable time in editing even when imperfect.

Some dictionaries listed many one-letter and two-letter words. It was important to exclude these, except for important foreign words like “de” in French and “ab” in German, to prevent translation of hash-like file names originally containing intermixed numbers.

Our corpus contained many compounds, some standard like “fairytale” and “trackball”, but most not like “brightideas”, “bidcenter”, and “wifedonation”. A Spanish example was “escuelamusica” (“music school”). Non-dictionary compounds were inferred by splitting unrecognized words in two parts and checking whether the pieces were known words in the same language. The error rate could be kept below 1% by requiring pieces of at least four characters each. Also sought were splits involving a single initial or final letter (e.g. “ktextcolor”), and splits involving some known plural endings. Analysis found around 50,000 proposed compounds, mostly English, that had to be manually checked over several weeks. Examples of errors were “personales” = “person” + “ales” and ”animados” = “anima” + “dos”, both proposed English compounds that are better interpreted as Spanish words. Abbreviations were frequently seen in compounds, e.g. “hpmcpap” and “ndfapi”, and many of these were found automatically due to our collection of standard cyberspace abbreviations. All abbreviations had to be predefined for each language because their automated inference is unreliable due their ambiguity and human creativity in abbreviating. Splitting of foreign-language compounds into known pieces enables their translation, and the results can be added to the dictionary.

The likelihood of a language for a set of words can be proportional to the number of words that match in that language’s word list. Counts should be adjusted however based on the size of the word list since these varied from 3,642 for Malay to 75,934 for Arabic. A reasonable assumption can be made that the lists represent the most common words in each language. Again we assume Zipf’s Law, so a word list of length N will cover a fraction proportional to

$$\sum_{k=1}^{N}(1/k) \approx 0.5772 + \ln(N) + (1/2N)$$

of words in a language. Each match to a word list can be weighted by the inverse of this.

Latin and Chinese posed interesting problems. Most Latin is legal phrases, biological names, and literary terms for which translation is not useful, so we do not translate it. Chinese does not use much punctuation, making recognition of words difficult. We use the method of splitting Chinese character strings in two until parts are recognized, then inserting spaces; the split with the longest word is preferred. Chinese punctuation is inconsistent – for instance, we found a Braille dot used as a period – but we segment words at any of its punctuation that we can find.

3.3. Using character distribution for each language

Another language clue is the distribution of Unicode characters in the words. For instance, words solely of ASCII characters are generally not Chinese, although LA-Strings guesses that they are on occasion. The association between script and language was exploited by grouping characters into categories based on their Unicode code-point values. The groups used were Nonalphabet, ASCII, Latin-1, Latin-A, Latin-B, Phonetic, Greek, Cyrillic, Hebrew, Arabic, Devanagari, Bengali, Indic, Thai, SE Asian, Hangul, Latin Extended Additional, Japanese, Chinese, and Unusual (everything else). Distributions were computed of these character types for each

language using our dictionary information. The inner product of these distributions was made with the distribution of character types in a directory, and maximum of this and 0 was used as the probability of a language for that directory.

Similar analysis was done for distribution of the characters themselves. The dictionaries were scanned to get statistics of the fraction of the time a character occurred in words from each language. Conditional probabilities of a language for each of the first 70,000 Unicode code points were computed assuming all languages and all words were equally likely. For a set of words to be identified, we calculated the likelihood of a given language L from a normalized Naive Bayes model based on the conditional probabilities of the characters in the language $p_{i,L}$:

$$p_L = \exp[(1/M)\sum_{i=1}^{M} \ln(\max(p_{i,L}, c_{i,L})]$$

Here $c_{i,L}$ is a lower-bound probability of a character when there are no occurrences of it in the dictionary, something not zero because our dictionaries can be incomplete and users of every language occasionally employ foreign words. The probability is determined by the character group, thus any character in the Cyrillic Unicode range (U+0400 to U+04FF) is considered likely to be Russian even if it never occurred in a Russian dictionary entry. The "private area" characters in Unicode were ignored since they are intended to be application-specific; they frequently appeared in word-processing software as Chinese characters, but were consistently untranslatable by our translation systems.

3.4. Using country of origin

Another clue to the language of a path is the country of origin of the drive (we do not have any more specific information about origin in our corpus). For instance, nearly all the Turkish in the corpus came from drives purchased in Turkey. Site www.infoplease.com provided country-specific estimates of language frequency of use. However, these weights refer to speaking preferences of natives, and the weights for cyberspace will give a higher weight to English. From inspection of the corpus it was estimated that 80% of the words in paths will be English or untranslatable as a minimum, and more for countries that use English more.

3.5. Using keyword clues in the path

An obvious clue to a language is a language name in the file path. For instance, "Program Files/Any Video Converter/lang/fr" has files for a French-language version. To narrow the cases, only quite specific keywords were considered: language codes in the several ISO standards, names of languages in both English and the language itself, and names of countries in both English and the language itself. We strip off common prefixes such as "lang" from the directory name. This clue applied to 4.7% of the directories in our corpus.

3.6. Using inheritance of probabilities from the superdirectory

Another clue to the language is that of the directory above the one under consideration, which is helpful when the directory under consideration has only a few words. This is easy to implement with a hash lookup for the language probabilities of the directory above.

3.7. Combining the clues

Care must be taken in combining clues since they are not independent and reasonable default values in the case of missing evidence (such as words not in any dictionary) are hard to estimate. The safest approach is to treat the clues as disjunctive rather than conjunctive as with Naive Bayes. This suggests evidence combination by adding the weighted likelihoods. We must experimentally determine the best weights on the factors (see section 5).

4. Translation of paths in the identified language

4.1. Determining the translation languages for a path

Our ultimate goal is to translate file paths. Knowing the predominant language of a directory helps, since for instance "pie.jpg" could be a picture of pastry in English or a foot in Spanish. But some directories do not have a predominant language. For instance, the words of one directory mixed French and English:

> *preset add noise grain de photo faible moyen preset add noise grain de photo élevé preset add noise réglages usine preset aged newspaper réglages usine preset airbrush lumière*

Additional clues are thus sometimes needed to resolve the language of a directory segment in an individual path. Two obvious clues are the dictionary data on the particular words and their character distribution. To combine the three, multiplication is appropriate using:

$$p_L = p_{dict,L}\, p_{chars,L} \sqrt{p_{dir,L}}$$

A conjunctive rather than disjunctive approach is suggested here because each clue must be strong for the correct language. However, the square root (a weighting by half in the logarithmic domain) was important to downweight the directory clue, since the language predominant in a directory can be overridden by its individual files, though it is a good way to break near-ties on the other factors as between Spanish and Portuguese. The highest-rated language, if it is not English or "untranslatable", is then used for translation of the words.

As an example, the path “Documents and Settings/defaultuser/Mes documents/Ma musique/.” splits into four directories and one file name. All the characters are ASCII in the frequencies of English. The words “documents”, “and”, “settings”, ”defaultuser” and “ma” are recognized as English words, the fourth as an inferred compound; “mes”, “documents”, “ma”, and “musique” are recognized as French words. Thus the preponderant evidence is that the first two directories are English and the second two are French. We send the second two separately to the translator with a directive to translate from French.

4.2. Translation

We tested three translation methods: the Systran service (www.systransoft.com), the Google Translate service (translate.google.com), and our own dictionary-substitution method. Systran has existed in some form for fifty years (Hutchins and Somers, 1992, chapter 10). (Wilks, 2009) estimated that it had a 60% success rate which was the best of any translation system then. It currently handles 17 non-English languages and provides (with some difficulty) an applications programming interface for batch processing. Google Translate is more recent and covers 65 languages. But unlike most of our toolkit, Systran and Google Translate are proprietary. To gauge their worth, we wrote a simple translator as a comparison. Since we already have dictionary information (including some multiword entries) for the languages we encountered, we can use it to produce a word-for-word translation, which we hypothesized would suffice for the many single-word file and directory names, and might provide a good approximation for the multiword ones sufficient for keyword lookup. Word-for-word translation also filled gaps in translator coverage, as for Hausa.

If a translation is found, we replace the original words in the file path with the translation, inserting the original punctuation and digits and setting case analogously. If the translation and the original have the same number of words, punctuation marks are inserted one-to-one. If the translation has fewer words, we use as many punctuation marks in order as we can; if the translation has more words, we repeat the last mark as many times as necessary. For instance, “menu_buscar_cambios_v26[1]” becomes “menu_to_search_for_changes_v26[1]”. This heuristic works well for most directories since usually one punctuation mark is used consistently as a delimiter within one directory or file name. Punctuation marks include artificial marks for separating case changes as in “callEdit” and between components of compounds that have been split to aid translation. Since many directory and file names occur multiple times in cyberspace, it is important to cache translations. It also enables a translator to learn from experience, since the translations found can be added to the dictionary.

5. Experimental results

Programs were implemented in Python 3.2. They were tested on a main corpus of 50.1 million file paths, including 26.3 million distinct ones, which were in 1.46 million distinct directories; and then on a secondary corpus of 29.4 million files not used for development. Identification was attempted for 35 different languages. Testing had three phases: identification of the languages of directories, identification of the languages of path pieces, and translation comparison.

5.1. Testing of directory language identification

The first step excluded directories whose words, after removal of punctuation and digits, were all English or untranslatable. 200 examples of the directories excluded were examined and all were found to be correctly excluded. An example was a Macromedia directory which with words “effectivemeasure net embed redtube com local settings”, all in our English word list thanks to compound analysis.

To test language identification of the remaining directories, a 1000-item directory test set was created in part from a random sample of directory segments with at least one translatable word, manually excluding a few directories with multiple languages. Random English and untranslatable segments were included at a lower frequency to boost the occurrence rates of foreign languages to about 50% of the total. Most languages in the test set were easy to identify, but a few required testing in Google Translate.

Table 1 shows the testing of the language clues on this test set. Two metrics are given. Basic accuracy is the fraction of exact matches on language; modified accuracy does not count as errors the confusion of English with untranslatable directories (neither are sent for translation) nor confusion of languages with their transliterated forms (both are designated for the same translation language). Modified accuracy also weights at one third the incorrect identification of English or untranslatable words as a different language, since a translator will usually just echo such words with little harm.

All seven clues appear useful in isolation, but their correlation meant that a better test was to eliminate each in turn from the combination of all seven and see how it affected performance. Removal of character-type clues improved performance on both metrics, and removal of inheritance clues in addition boosted performance further, so we concluded that these clues can be eliminated. LA-Strings was kept despite its apparent mixed benefits in the hope its broad coverage would provide robustness on new languages. Experiments showed best performance with a weight of 0.22 on LA-Strings, 0.11 on dictionary lookup, 0.34 on characters, 0.012 on country, and 0.07 on keywords. We conclude, surprisingly, that character N-grams do not help language identification of directories. We suspect that character-type clues were too similar to character clues to help, and that inheritance misleads with frequently changing languages in paths.

Table 1: Accuracy of the seven clues to directory language on the directory test set.

All 7 clues	**Just LA-Strings**	**Just dictionary lookup**	**Just character types**	**Just characters**	**Just country**	**Just path keywords**	**Just inheritance**
.721, .904	.547, .680	.649, .857	.551, .775	.359, .743	.214, .338	.469, .684	.446, .675
All without character types and inheritance	**All without LA-Strings**	**All without dictionary lookup**	**All without character types**	**All without characters**	**All without country**	**All without path keywords**	**All without inheritance**
.798, .934	.694, .904	.662, .836	.775, .929	.703, .898	.722, .886	.793, .897	.765, .883

5.2. Testing of individual-path language identification

Language identification of pieces of individual file paths that serve as the input to Systran was also tested. 5,403,058 pieces were processed from the directories passed from the first phase, and 194,114 (3.6%) were judged worthy of translation. Considering the directory pieces excluded in the first phrase, we estimate that 2.0% of all distinct names in cyberspace should be considered for translation.

A different test set of 3518 phrases was used, a random sample of all the paths sent to Systran for translation with a lower sampling frequency for English and untranslatable phrases to limit them to 50% total. Results are shown in Table 2 with "tl-" denoting a transliterated language. Here ar = Arabic, cs = Czech, de = German, en = English, es = Spanish, fr = French, he = Hebrew, it = Italian, ja = Japanese, ko = Korean, nl = Dutch, pt = Portuguese, ru = Russian, sv = Swedish, tr = Turkish, zh = Chinese, and un = untranslatable; "other" is Czech, Farsi, Greek, Hausa, Polish, and their transliterations. A phrase is assigned occasionally to different languages in different contexts; when this happens, we tabulated only the language inferred for the first occurrence.

Table 2: Confusion matrix of languages on individual-path segment language identification.

	ar	de	en	es	fr	he	it	ja	ko	nl	ru	tr	zh	tl-de	tl-es	tl-fr	tl-he	tl-hi	tl-it	other	un
ar	799		2																	8	3
de		14	3											36							10
en			301											2	4	1			1	3	273
es			4	107										2	370					11	85
fr			2		25											9					2
he						179															1
it			1				9								1						
ja								6													1
ko			1						17												1
nl			1							3											1
ru											2										
tr												17								8	6
zh			1										67								1
tl-ar			2																	1	8
tl-de														2							
tl-es				1											80						1
tl-fr																8					1
tl-he																					1
tl-hi			1											1						8	3

tl-it			1												1				3		
other	7		1									2								9	5
un			7												3	2	2			8	952

93.7% modified accuracy resulted with all factors, 88.4% with the directory factor deleted, 88.5% with the dictionary-lookup factor deleted, and 90.0% with the characters factor deleted. So it appears that all three factors help.

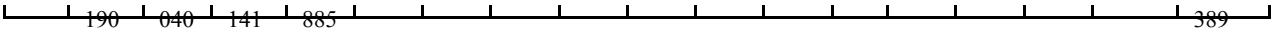
We also tested 29.4 million files of recently purchased drive images not used for development, which had more French and less Spanish and Arabic. A new test set of 1000 items was created following the previous methods. Basic accuracy on the test set was 89.9% and modified accuracy was 93.5%, so performance was not substantially different from that on the original corpus. It appears that we have covered most of the important words of cyberspace in most major languages.

An interesting question is whether user-generated file paths need translation more than others. We extracted from our corpus all paths with extensions indicating Web pages, documents, presentations, spread sheets, email, camera images, audio, video, and program source code using the classification scheme described in (Rowe and Garfinkel, 2012) which provided 925 such extensions. This reduced the paths to 24.8% of the original number, while reducing the number of distinct translatable path pieces to 29.0% of the original. So these file paths do not appear to need particularly more translation. We conclude that operating-system and software-related files often retain the language of their origin.

Table 3 breaks down by country the language identified for every distinct directory piece with at least one translatable word. Counts include both the language and its transliteration, and we only show the countries for which we have substantial data. Numbers greater than 1000 are given on two lines. Ae = United Arab Emirates, bd = Bangladesh, ca = Canda, cn = China, de = Germany, eg = Egypt, gh = Ghana, il = Israel, in = India, ma = Morocco, mx = Mexico, pk = Pakistan, ps = Palestine, sg = Singapore, tr = Turkey, and ? = unknown origin. This reveals interesting business connections; who knew there was so much Chinese in Israel or Korean in Bangladesh? So there is an overwhelming amount of English in world cyberspace regardless of country. It is surprising how little Chinese (zh) and Hindi (hi) was present considering that they are among the world's most-spoken languages and we have many drives from China and India. This may reflect the difficulty of acquiring and using non-English keyboards.

Table 3: Country of origin (rows) versus inferred language of directories (columns).

	ar	de	en	es	fr	he	it	ja	ko	nl	pt	ru	sv	tr	zh	other	un
ae	28	105	40,560	69	5,879		20		16	5	22		24	38	353	88	144,218
bd		2	7,300	1	5		2		1,080		2		13	2	32	29	44,376
ca		3	7,389		6						1		25	4	70	6	38,134
cn		143	89,935	86	75		57	972	7	34	29	12	105	14	5.350	172	257,991
de		3,460	10,545	82	13	2	28		4	1	21		19	17	21	98	56,689
eg	655	19	3,687	17	18		5				2		13	2		61	28,001
gh		5	25,569	6	9	2	6		577		2		8	2	13	33	63,416
il	5	146	117,441	838	160	23,873	59		2	23	71		137	25	4,841	277	657,282
in	44	777	210,764	1,192	707	7	642	2		432	316	2	180	806	52	1,029	703,589
ma	1	5	1,226	7	258		8						7	1	1	7	16,578
mx		58	44,889	97,831	78	4	129		3	6	127		30	10	197	181	330,763
pk		1	5,730		3		14		90		2		1	1	6	17	31,091
ps	649	13	63,136	36	31	12	20			3	6		41	10		73	210,933
sg	0	89	82,766	42	14	9	10	13			9		10	6	59	40	204,875
tr	0	40	18,478	20	17	2	4			4	7		8	4,290		78	55,753
?	126,	5,	432,	65,	474	50	511	146	34	209	892	27	603	563	906	4,649	1,013,



5.3. Testing of translation

Testing compared results from our own word-for-word translation, Systran, and Google Translate. All made a variety of errors. Table 4 shows some examples that caused difficulty.

To better quantify performance, we examined 200 randomly selected results each for Spanish, French, and Japanese, three languages for which the authors had expertise (Table 5). Ties for "best" counted in both categories. Google Translate performed the best overall, and our simple word-for-word substitution worked surprisingly well, although to be fair, the 18.3% single-word names and 30.7% two-word names were easy for all translators. Since Google Translate provided translations for 46% of our dictionary words, word-for-word translations were similar to those of Google Translate in many cases although the words were not in the same order.

Something to translate was found in 3.7% of the paths in our main corpus. As examples of full path translations using Systran, Applications/Microsoft Office X/Office/Assistenten-Vorlagen/Kataloge/ Kapsel was translated to Applications/Microsoft Office X/Office/Assistants-Were-present/Catalogs/ Cap, and top.com/السلسلة المعلوماتية/تصميماتي.jpg was translated to top.com/My designs/The computer-based series.jpg.

6. Conclusions

It appears that automatic translation of multilingual file paths to English can be done reasonably successfully. The task is trickier than it might seem and the best methods differ from those previously successful for prose paragraphs. Paths must be partitioned by directory and each piece analyzed separately, and knowledge of the preponderant language of a directory is valuable. However, we still obtained some erroneous and even nonsensical translations from all translators, especially for Arabic and Chinese. This suggests that automated path translation currently can only help the initial phases of forensic investigation.

Table 4: Example comparative translation results.

Original	**Word-for-word translation**	**Systran translation**	**Google Translate translation**
Spanish: entren ser lider	come be head	they enter to be leader	come to be leader
Spanish: linda ronstandt la cigarra canciones de mi padre	cute ronstandt the cicada songs mine cool	the cicada is contiguous ronstandt songs of my father	linda ronstandt cicada songs my father
French: premierbaiser pps	first kiss pps	premierbaiser pps	premierbaiser pps
French: tetes de vainqueurs pps	heads of winners pps	suck winners ps	heads of winners pps
German: tipps fürs surfen	tips for surf	tipps for that surf	tips for surfing
German: prüfungszeugnis speditionskauffrau	examination certificate forwarding clerk	certificate of examination shipping company clerk	audit certificate forwarding clerk
Dutch: slonzige randen	slovenly edges	sloppy being affected by marginal blight	sloppy edges
Polish: magazyn kratownica	repository truss	warehouse grate	storage grid
Arabic: مشكلة سقوط السارية	problem downfall applicable	Shaper of falling contagious	Problem of the fall of the applicable
Arabic: عادل جواد دنيا ليل	fair jawad world night	Horse life of night equated	Adel Jawad night minimum
Japanese: デスク	desktop	Indication of	Show

Korean: 대처광 이준영btv	display I love ahn btv	desktop Secret examination and Lee Jun-young btv	Desktop I love Ahn, btv
Chinese: 陆 行 鸟 饲 å x 手 x e x c	陆行鸟饲 å x hand x e x c	Goes by land the bird to raise å x x e x c	The land line Torikai å x hand x e x c

Acknowledgements

Thanks to Systran for important technical assistance, and also to Albert Wong.

Table 5: Comparative translation testing.

Language /Measure	Spanish	French	Japanese
Word-for-word OK	.72	.74	.57
Systran OK	.65	.61	.75
Google Translate OK	.81	.80	.92
None OK	.07	.03	.04
Word-for-word best	.55	.65	.48
Systran best	.52	.55	.48
Google Translate best	.78	.75	.85

References

1. Brown, R. Finding and identifying text in 900+ languages. Digital Investigation, Vol. 9, pp. 34-43, 2012.
2. Basis Technology. Odyssey digital forensics search. www.basistech.com/datasheets/Odyssey-Digital-Forensics-Search-EN.pdf, accessed April 2013.
3. Buchmeier M. Bilingual Dictionaries for offline use. en.wiktionary.org/wiki/User:Matthias_Buchmeier, accessed February 2013.
4. Da Silva J, Lopes G. Identification of document language is not yet a completely solved problem. Proc. Intl. Conf. on Intelligent Agents, Web Technologies, and Internet Commerce, 2006.
5. Hutchins W., Somers H. An introduction to machine translation. London, UK: Academic Press, 1992.
6. Larson M. Meaning-based translation: a guide to cross-language equivalences. Boston: University Press of America, 1984.
7. McCandless M. Language detection with Google's Compact Language Detection, 2011. blog.mikemccandless.com/2011/10/language-detection-with-googles-compact.html, accessed April 2013.
8. Mishra G, Nitharwal, S, Kaur, S. Language identification using fuzzy-SVM technique. Proc. 2nd Intl. Conf. on Computing, Communication, and Networking Technologies, 2010, p. 1-5.
9. Rowe N, Garfinkel S. Finding suspicious activity on computer systems. Proc. 11th European Conf. on Information Warfare and Security, Laval, France, July 2012.
10. Shuyo N. Language detection library for Java. http://code.google.com/p/language-detection, 2010.
11. Turchi M, Atkinson M, Wilcox A, Crawley B, Bucci S, Steinberger R, and Van der Goot E. ONTS: 'Optima' news translation system. Proc. Conf. European Chapter of the Assoc. for Computational Linguistics, Avignon, France, April 2012.
12. Unicode. Unicode 6.2.0. Unicode, Inc. 2013. http://www.unicode.org, April 2013.
13. U.S. FBI. The Federal Bureau of Investigation's foreign language program -- translation of counterterrorism and counterintelligence foreign language material. Audit Report 04-25, www.fas.org/irp/agency/doj/oig/translation.pdf, accessed April 2013.
14. Wilks Y. Machine translation: its scope and limits. New York: Springer, 2009.
15. Yang X., Liang W. An n-gram and Wikipedia joint approach to natural language identification. Proc. 4th Intl. Universal Communication Symposium, Dalian, CN, October 2010, p. 332-339.

---

* Email address: ncrowe at nps.edu