MLADENIC D. (1998), "Turning Yahoo to Automatic Web-Page Classifier", in *Proc. European Conference on Artificial Intelligence*: 473-474.

RAYSON P. and GARSIDE R. (2000), "Comparing corpora using frequency profiling", in *Proc. of the Comparing Corpora Workshop at ACL 2000*,Hong Kong: 1-6.

REHM G. (2002), "Towards automatic web genre identification – a corpus-based approach in the domain of academia by example of the academic's personal homepage", in *Proc. of the Hawaii Internat. Conf. on System Sciences*.

SANTINI M. (2007), *Automatic Identification of Genre in Web Pages*, PhD thesis, University of Brighton.

SEBASTIANI F. (2002), "Machine learning in automated text categorization", in *ACM Computing Surveys*, n° 1, vol. 34.

SHAROFF S. (2006), "Creating general-purpose corpora using automated search engine queries", in Baroni M. and Bernardini S. (eds), *WaCky! Working papers on the Web as Corpus*, Gedit, Bologna, http://wackybook.sslmit.unibo.it.

SINCLAIR J. (1996), *Preliminary recommendations on corpus typology*, Expert Advisory Group on Language Engineering Standards document EAG–TCWG–CTYP/P, http://www.ilc.cnr.it/EAGLES96/corpustyp/corpustyp.html.

WITTEN I. and FRANK E. (2005), *Data Mining: Practical machine learning tools and techniques*, Morgan Kaufmann, San Francisco.

ZHAO Y. and KARYPIS G. (2004), "Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering", in *Machine Learning*, n° 3, vol. 55.

Identification of Languages and Encodings in a Multilingual Document

Anil Kumar Singh[1] and Jagadeesh Gorla[1]
Language Technologies Research Centre, IIIT, Hyderabad, India

Abstract

Text on the Web is available in numerous languages and encodings, often not according to any standards. The number of multilingual documents on the Web is also increasing. The problem of identifying the languages and encodings in a multilingual document and marking portions of a document with them has not been addressed so far. We present an exploration of this problem, the implied or required assumptions, and a solution. The problem can be divided into three parts: monolingual identification, enumeration of languages and identification of the language of every portion. For enumeration, we have been able to get a precision of 96.20%. We also experimented on language identification of each word. Given correct enumeration, we could obtain *type* precision of 90.91% and *token* precision of 86.80%. Finally, we show how precision is affected by language distance.

Keywords : multilingual, language identification, encoding, retrieval.

1. Introduction

One user one language and *one document one language* have been the assumptions on which much of the work on computers, the Internet and even Natural Language Processing (NLP) has been based. But as more and more people from around the world, especially from countries with many languages, have joined the community of computer and Internet users, the importance of accommodating bilingualism and multilingualism is gradually being realized.

Language identification becomes an important problem in the electronic world of many languages (Gordon 2005), even more so when multiple languages are mixed up in one document. Monolingual identification has been attempted by many researchers and it is now considered by many to be an almost solved problem. But multilingual identification has been rarely attempted. This is partly due to the fact that for a long time most of the documents on the Internet were monolingual. Multilingual documents are becoming more common now. Since it is very difficult to directly estimate the number of multilingual documents, we have used an indirect method as shown in Table 1.

[1] {anil,jagadeesh}@research.iiit.ac.in

	German	French	Spanish	Chinese (Traditional)	Chinese (Simplified)	Japanese
Slogan	1.30	1.33	1.06	0.19	0.01	0.10
Piece	1.23	1.52	1.32	1.22	1.15	1.48
Peace	1.53	1.65	1.44	1.23	1.15	1.85
Town	1.61	1.53	1.49	1.36	1.40	1.56
Trouser	0.10	0.06	0.76	0.02	0.02	0.60
Clutter	0.32	0.08	0.05	0.03	0.08	0.07
Down	4.48	1.67	1.44	1.28	1.5	1.64

Table 1. Multilingual Pages on the Web: These statistics indirectly indicate the number of bilingual pages on the Web. The numbers (in millions) are actually the number of results returned by Google when an English word was searched among pages of some other languages. The English words searched were deliberately selected to be of different origins: Latin, Celtic, Germanic etc. This was done to take care of the cognate words factor. The words are also diverse in terms of their frequency of occurrence in English.

In this paper, we will discuss the problem of multilingual language identification and consider different scenarios and the assumptions they imply or require. The solution to the problem will depend on these assumptions. We also show that the problem can be divided into three parts and these parts can be solved separately. The first part is monolingual identification. Many methods with very high precision are available for this part. The second part is **language enumeration**, *i.e.*, finding out what languages are present in the document. The third part is **segment identification**, *i.e.*, identifying the language of segments of text in the document. If the segments are assumed to be single words, we can further divide the problem into **word type identification** and **word token identification**. In this first work on formulating the problem of multilingual language identification and solving it in a systematic way, we propose a method to solve the language enumeration and segment identification problems under one of the most likely scenarios. We have evaluated these methods fairly extensively. The results achieved are highly encouraging. We also consider the relationship between precision (of identification) and the distance between language-encodings. Throughout this paper, *identification* means *language and encoding identification*, unless stated otherwise.

2. Assumptions

The solution to the multilingual identification problem completely depends on the assumptions we make. One or two of these assumptions may be unavoidable to make the problem tractable. Some of these assumptions have been given in below.

1. **Diversity Assumption**: The accuracy of a language identifier depends on the number of languages from which the identifier has to select one. This reflects the coverage of the identifier in terms of linguistic diversity, which implies an assumption about linguistic diversity. There are two kinds of *diversity assumptions*, both of which can be applicable at the same time.

   (a) **Global Diversity Assumption**: This is about how many languages are assumed to be in the world. In practical terms, this is reflected in the number of languages for which the system has been trained.

   (b) **Local Diversity Assumption**: For a particular user or for a particular context, the number of possible and **relevant** languages may be less than the number for which the system has been trained. For example, a user may only be interested in the documents in European languages, even though the system has been trained for languages from around the world. In such a case, a *local diversity assumption* is likely to increase the accuracy and speed of the identifier.

2. **Limited Ambiguity Assumption**: Multilingual documents have text in more than one language, but if we do not assume a small limit to this number, the problem may not be tractable, unless we assume large segment sizes. All the algorithms for monolingual identification work well only when the test data size is sufficient, *e.g.*, 100 characters. Thus, to make the problem solvable, we will make the *limited ambiguity assumption*, viz., that the number of languages to be disambiguated *for a segment* is a very small number. In our experiments, we have assumed this number to be either two or three, which means that the multilingual documents can be either bilingual or trilingual. Unlike the *diversity assumption*, this assumption is about the possible languages in a document, not in the world. Therefore, it applies only to a multilingual identifier, not to a monolingual identifier.

3. **Language Switching Assumption**: Another assumption that applies only to a multilingual identifier is the *language switching assumption*. This specifies how frequently or where a shift from one language to another can occur in a document. There are two such assumptions, only one of which can apply at a time.

   (a) **Long Sequence Assumption**: This assumption says that the minimum segment size in any language is large enough for a monolingual identifier to identify its language accurately. If we make this assumption, the problem of segment identification actually becomes a problem of identifying where language shift occurs and from which language to which language. This is, of course, a less realistic assumption.

   (b) **Isolated Word Assumption**: The more realistic assumption is that every word in the document can be in a different language, subject to the *limited ambiguity assumption*, *i.e.*, language switch can occur at any word boundary. Our experiments have been conducted under this assumption. The problem in such a case is to identify the language of every word, as every word is a segment. In one sense, this is a simpler problem because we do not need to identify the boundaries of the segments. However, since the segment size can be as small as a word of one character, the precision is likely to be low.

4. **Available Information Assumption**: Language and encoding identification can be very simple under the ideal conditions. If we assume that all the documents are in HTML or XML etc., that the languages and encodings have been specified by using the relevant codes everywhere and that only those languages and encodings are possible that can be specified using these standard codes, then we merely need to write a program to use this information. The solution can also depend on the availability of other sources of information, *e.g.*, the list of function words or of characteristic unique words for every language encoding pair.

3. Languages and Encodings

Support for many languages with speakers numbering more than 10 million, *e.g.* the languages of the Indian sub-continent, has been mostly non-existent on computers. The 'encodings' used for these languages (except nowadays Unicode) are not recognized by the operating systems or Web authoring tools or even by the HTML standards. And they cannot be, to the satisfaction of all, because there is no exhaustive commonly agreed upon list of languages of India, let alone that of encodings. What is a language for some is just a dialect for others. And for every language (or dialect), there are numerous very different encodings. This, in simple words, means that the Web pages written in Indian languages almost always contain 'wrong' encoding in the meta-tags, simply because there is no HTML code for the encoding used by the author of the Web page.

A problem closely associated with language identification is that of encoding identification. That this is also a research problem may not be evident at first if we think only in terms of standard encodings like ASCII, ISO-8859-1, UTF-8 etc. An examination of multilingual documents will show that, from the language processing point of view, we have to understand the term 'encoding' a bit differently. It does not just mean the encodings recognised by operating systems or by standards organizations. 'Encoding' is a scheme used to **encode** the text of a particular language. It is a mapping from the letters (or other units) of a script to numbers in a range (say, 0 to 255). Thus defined, the term includes unrecognised encodings, which are quite common in many parts of the world. These encodings may be transliteration schemes, but not always.

Language and encoding are closely interconnected in such a way that if we could identify the language, we would most probably have also identified the encoding, or vice-versa. It could be argued that they can be identified separately and that identification of one may help the other. Previous work and our own experience, however, indicates that considering language-encoding pair as one unit might be a reasonable way to solve the problem. Under certain situations we could, of course, have prior information about one of these. This can be used by the identification system to make a better prediction.

For our experiments, we have not used any prior information about language or encoding, even if it could have been obtained just by looking at the byte format, *e.g.*, in the case of Unicode encodings like UTF-8, UTF-16, etc. We did this to evaluate our system only for the cases where such prior information is not available. For identifying language-encoding pairs in all the cases, we have used only a statistical method, as explained later. The synthetically created test data (as it was too expensive to 'annotate' real life multilingual data) for evaluation is adequate for the assumptions we have made, but for different assumptions we will need to create such data differently.

4. Related Work

There is a long history of work on language identification. In fact, it was one of the first language processing problems for which a statistical approach was used. Ingle (1976) used a list of short characteristic words in various languages and matched the words in the test data with this list. Such unique strings based methods were meant for human translators.

The earliest approaches used for automatic language identification were based on that same idea of unique strings. They were called 'translator approaches'. Beesley's (1988) automatic language identifier for online texts used mathematical language models originally developed for breaking ciphers. These models basically relied on orthographic features like characteristic letter sequences and frequencies for each language.

Some of the methods were similar to n-gram based text categorization (Cavnar and Trenkle 1994) which calculates and compares profiles of n-gram frequencies. Cavnar also proposed that the top 300 or so n-grams are almost always highly correlated with the language, while the lower ranked n-grams give more specific indication about the text, namely the topic. Many approaches similar to Cavnar's have been tried, the main difference being in the distance measure used. The similarity measures tried for language identification include mutual information or relative entropy, also called Kullback-Leibler distance (Sibun and Reynar 1996), cross entropy (Teahan and Harper 2001), and mutual or symmetric cross entropy (Singh 2006).

Giguet (1995) relied upon grammatically correct words instead of the most common words. He used the knowledge about the alphabet and the word morphology via *syllabation*. He tried this method for tagging sentences in a document with the language name, *i.e.*, dealing with multilingual documents.

Johnson's method (Stephen 1993) was based on characteristic 'common words' of each language. This method assumes unique words for each language. In practice, the test string might not contain any unique words.

Cavnar's method, combined with some heuristics, was used by Kikui (1996) to identify languages as well as encodings for a multilingual text. He relied on known mappings between languages and encodings and treated East Asian languages differently from West European languages.

There has been no comparable systematic work on multilingual text documents, although there has been some work based on an Optical Character Recognition (OCR) system, such as by Tan *et al.* (1999). One attempt at multilingual identification was by Prager (1999). His Linguini system uses a vector space based monolingual identifier to also find out the component languages of a document and the relative proportions of each. Artemenko *et al.* (2006) tried a method for identifying the languages in a document and have reported an accuracy of 97% for this task. But neither of them identified languages of segments.

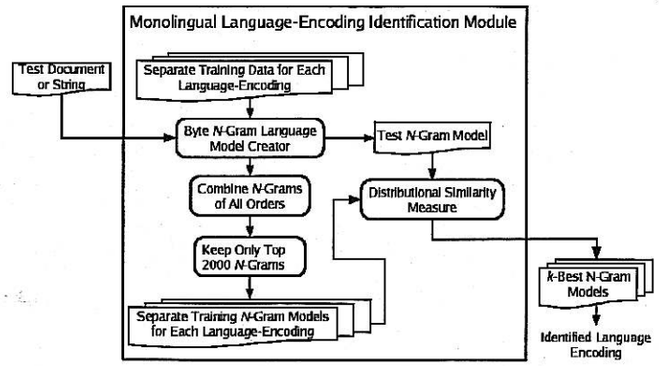

Figure 1. Monolingual Language Identification

5. Monolingual Identification

We use a monolingual identifier as a black box for multilingual identification. The method used by us for monolingual identification is based on Singh's work (Singh 2006), using symmetric cross entropy as the similarity measure. Such a monolingual identifier effectively calculates a distributional similarity score between two n-gram models. The system is trained by preparing byte based n-gram models from the training data. Then n-grams of all orders are combined and sorted by rank. Only the top N n-grams (where $N = 2000$) are retained because they are the characteristic n-grams for a language (Cavnar and Trenkle 1994).

For the given test data or string, we prepare a similar n-gram model and combine the n-grams of all orders. However, unlike for training models, we keep all the n-grams. This is because the test strings will be usually small: in our case as small as a word. This n-gram test model is then compared with all the training (or reference) n-gram models and similarity scores are calculated using symmetric cross entropy:

$$(1) \quad sim(p,q) = \sum_{x=y} (p(x) * log\ q(y) + q(y) * log\ p(x))$$

where p and q are the two distributions, or in the present case, n-gram models; x and y are the variables (n-grams) in the two distributions or n-gram models, respectively.

Now we can select the most likely language-encoding pair(s) based on this n-gram model similarity score. We kept the order of byte n-grams as 6 (instead of the usual 4 or 5) in our experiments because our method for multilingual identification depends on identifying even the small words correctly.

The process for monolingual identification has been shown in figure 1.

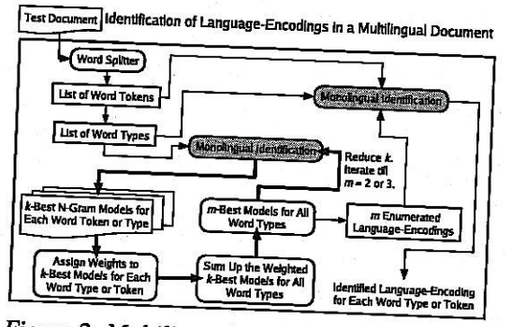

Figure 2. Multilingual Language Identification

6. Multilingual Language-Encoding Identification

There are two aspects of generalized multilingual language-encoding identification, one of which makes the problem much harder than monolingual identification. This is the fact that it is hardly possible to get representative training data for multilingual documents due to the very nature of the problem. However, the second aspect makes the problem solvable even without multilingual training data. This is the *limited ambiguity assumption* mentioned earlier: there are likely to be only two or three languages in a document in most cases.

Language-encoding identification, monolingual or multilingual, can be seen as a classification task and it can be argued that we should use some sophisticated pattern classification technique like maximum entropy for solving this task. However, the previous work and our own experience suggests that a simpler n-gram model similarity based method is more suitable for this purpose. Also, techniques like maximum entropy would require annotated multilingual training and testing data which is not easy to prepare. The advantage is that only a small amount of training data (2500-10000 words) per language-encoding is enough and we do not need any specially selected features. The data need not even be very clean. A small amount of test data (5-15 words) is enough for accurate identification even if fairly high level of diversity (60 varied language encodings pairs) is assumed.

As indicated earlier, we have divided the problem into three parts. These are monolingual identification, language enumeration and segment identification. We make the *limited ambiguity assumption* (only two languages per document), assume a high level of diversity, and also assume that the language can shift at every word, which is a very realistic assumption. The last assumption means that we have to identify the language of every word in the document.

The outline of our method is shown in Figure 2. We first train a monolingual identifier as described in the previous section. Then, given a test document, we split it into words and form lists of word types as well as word tokens (instances of word types). Then, through a number of iterations which could be two or more depending on the level of diversity, number of confusable pairs, etc., we enumerate or identify the languages present in the document.

Category	Count	Examples
Related	51	Assamese-UTF8::Oriya-UTF8 Danish-ISO-8859-1::Norwegian-ISO-8859-1 etc.
Less Related	41	Catalan-ISO-8859-1::Russian-Windows-1251 Punjabi-UTF8::Telugu-UTF8 etc.
Unrelated	91	Dutch-ISO-8859-1::Marathi::Saamanaa Hindi-Typewriter::Tagalog-ISO-8859-1 etc.

Table 2. Language-Encodings Tested on for Evaluation

6.1. Language Enumeration

Two algorithms are being used at this stage. One is *monoKBest* (Algorithm 2). The other is *languageEnumerator* (Algorithm 1). The *monoKBest* algorithm returns K-best n-gram models or language-encodings for a given word type or token. The *languageEnumerator* algorithm returns the m-best language-encodings for a given document.

6.1.1. Notations Used in Algorithms

The following are the notations used in the algorithms:

1. d is a multilingual document.
2. $L = (l_1, \ldots, l_j, \ldots, l_J)$ is the list of possible language-encodings for any of the words in a given document. Note that J represents the global diversity assumption.
3. m is the number of languages after enumeration. It represents the local diversity assumption.
4. K is the number of best possible language-encodings in each iteration.
5. $MLS(x_i)$ is a vector of language-encodings and their corresponding scores for the word x_i:
   $MLS(x_i) = ((l_1^i, s_1^i), (l_2^i, s_2^i), \ldots, (l_K^i, s_K^i))$

6. $\overrightarrow{W} = (w_1, w_2, \ldots, w_r, \ldots, w_K)$ is a weight vector, where w_r is the weight assigned to a language-encoding (l), if l is the r^{th} best language-encoding of the word x_i. The values in this vector are manually assigned constants decreasing exponentially with the rank (p). These weights can be used to tune the algorithm.

Algorithm 1 languageEnumerator(d, K, L, W, m)

1: **for** each word $x_i \in d$ **do**
2: $MLS(x_i) \leftarrow monoKBest(x_i, L, K)$
3: **end for**
4: Total score of l_j in a document d is
$S_j \leftarrow \sum_{x_i} W_k * S_k^i$, where $l_j \in L$ and (l_k^i, S_k^i) occurs in the vector $MLS(x_i)$ for some $x_i \in d$ and $l_k^i = l_j$
5: $returnList \leftarrow < l_j, S_j >$
where *retunList* contains l_j, S_j pairs
6: Sort *returnList* based on the scores
7: $L \leftarrow K$-best languages-encodings (l_j's) from the *returnList*
8: **if** $K == m$ **then**
9: return *returnList*
10: **else**
11: $languageEnumerator(d, K-1, L, W, m)$
12: **end if**

Algorithm 2 monoKBest(x_i, L, K)

1: $MLS(x_i) \leftarrow K$-best possible n-gram models (language-encodings) l_k^i and their corresponding scores for the word x_i, where $l_k^i \in L$.
2: Sort MLS based on n-gram model scores.
3: return $MLS(x_i)$.

In each iteration for language enumeration, we go through the following steps:

1. Using the monolingual identifier, select the K-best n-gram models, *i.e.*, language-encoding classes for each word type or token.
2. Assign some weight to each of these classes depending on the relative rank of the class, *i.e.* assigning the weights to weight vector $\overrightarrow{W}$. What is crucial here is to assign the best weights to classes according to their ranks. To see why this is important, consider a document that has mostly Hindi words with a few English words. English and Hindi are the correct language classes present in this document. When the language classes are computed for Hindi words, the next best estimate of classes for these words is Marathi because it is very similar to Hindi. Thus, a large number of words in this document have Marathi as their second rank language. English appears as a first rank language for very few words. If we choose linearly decreasing weights for ranks, Marathi may overtake English in the cumulative score due to the sheer number of words falling into this class. However, if we

choose exponentially decreasing weights for ranks, the score difference between a class occurring at rank 1 a few times (English) and a class occurring mostly at rank 2 for a large number of words (Marathi) will be wide enough to allow correct discrimination of true language classes based on the cumulative scores.

3. After this has been done for all the words, a cumulative score for each language class is computed as the sum of the weights of all the words for which this class has been assigned.

4. Steps 1-3 are repeated after reducing the value of K, till we get the required number of m classes, say $m = 2$ or $m = 3$.

In the language enumeration stage, we do not necessarily have to select exactly the number of languages which are supposed to be in the document if our final purpose is to identify the language-encodings of segments or words. For example, if the document is known to be bilingual, we can select the top three language-encodings. This will ensure that in cases of errors in enumeration, we do not miss out on one of the correct languages in the document while identifying the segments or words. However, our results show that this logic may not apply in practice. Also, during this stage (*i.e.*, enumeration), we do not consider small words (less than 6 bytes), though we do consider them while identifying the language-encoding of word types and tokens.

There is another technique that our system uses optionally for language enumeration. This is based on preparing the list of unique n-grams (UNGs) for each language-encoding class. The accumulated scores of each class are multiplied by the normalized count of unique n-gram matches between the n-gram model of the class and the n-gram model of the test data.

6.2. Word Type Identification

Once the best possible m language-encodings have been identified for the document, we can simply use the monolingual identifier to tag the language-encoding of each word type. The important point is that we only have to discriminate between m classes and m will be usually only 2 or 3.

6.3. Word Token Identification

In the current work, we assign the language-encoding class of a word token to be the same as that of the word type of which it is an instance. In other words, we are not taking the context of the token into account. We plan to explore how context can be used to improve token identification.

7. Language Distances and Confusability

Intuitively, the difficulty of identifying the correct language out of two possible candidates should be more if the two candidates are closer, *i.e.*, if the languages are related.

This fact can also be stated in terms of the linguistic notion of *language distance* or *divergence*, sometimes also called the *genetic distance* between languages. Earlier attempts at this were based on comparing a list of (say, 200) words (Swadesh 1952). For purposes of information extraction, linguistic distances were adapted from the Dyen *et al.* (1992), who also used Swadesh like lists of words. In general, the language-language distance can also be calculated if the distribution of words is known, by using a distributional similarity measure like relative entropy. Nerbonne and Heringa (1997) measured dialect distance phonetically. Ellison and Kirby (2006) recently described an attempt at building genetic language taxonomies using a measure based on language internal similarities within the forms.

From our point of view, there is one important element missing from these methods of finding language distance: the encoding of the document. If the text in different languages is encoded differently and there is no converter available to convert between all possible pairs of language-encodings, then we can't directly work on word lists or distributions of words. What we need is a measure of distance between two language-encodings which gives us a quantitative measure of the confusability of two language-encodings. We have used a simple method for this. We are preparing byte based n-gram models from the training data for all language-encodings being considered. If we compare these models among themselves using a distributional similarity measure such as symmetric or mutual cross entropy, we will get an estimate of language-encoding distance. We can then study the relationship between identification precision and language-encoding distance (Figure 3).

	Enumeration Precision		Segment Identification Precision			
	(2 out of 2)	(2 out of 3)	Word Type Precision		Word Token Precision	
			Unknown	Known	Unknown	Known
Related	88.92%	97.87%	78.15%	85.85%	74.45%	81.96%
Less Related	87.55%	94.51%	81.66%	91.93%	77.75%	88.10%
Unrelated	85.51%	95.89%	81.97%	93.68%	77.86%	89.16%
Mixed	86.93%	96.19%	80.73%	90.91%	76.82%	86.80%

Table 3. Precision for Language Enumeration and Segment Identification

	Languages Unknown		Languages Known	
	50-50	80-20	50-50	80-20
Enumeration	93.59	83.53	-	-
Token	81.06	74.75	86.82	86.75
Type	85.43	78.24	90.96	90.87

Table 4. Precision for Two Different Ratios of Languages in a Document

8. Evaluation

We tested our method on bilingual documents with a fairly high level of diversity. We evaluated language enumeration as well as segment (word) identification. Since lan-

guage distance or confusability can significantly affect the results (section-7), we divided the language-encodings in four categories based on the distances among languages: (a) unrelated, (b) less related, (c) related and (d) mixed. Our evaluation is on these categories (Table 2). We experimented with and without using UNGs, but there was not much difference, which shows that either UNGs have no effect on the precision, or a more effective way of using them has to be found.

Since it is very difficult to get test data (for evaluation) under the high diversity assumption where not only the language-encodings of the document, but the language-encodings of each word or segment are known, we generated such data from monolingual documents by mixing words from documents in different language-encodings randomly, but in definite proportions and preserving the sequential order. The two proportions we used for evaluation were 50%-50% and 80%-20%. The final precision was averaged over documents containing language-encodings in these proportions. The maximum document size was kept at 1000 words. Since the proportion affects precision (though not uniformly), we also report the results for two different proportions. In cases where one language is in much less proportion than the other (80%-20%), the performance was lower in most cases.

For language enumeration, we performed evaluation for the cases when both the language-encodings are correctly identified and also when two out of three are correctly identified. The results are presented in Table 3. In all, we tested on 8498 documents, out of which two out of three were correctly identified in 8175 (96.20%) documents. Both were identified correctly in 7388 documents (86.94%). The separate results for the two proportions (50%-50% and 80%-20%) are shown in Table 3.

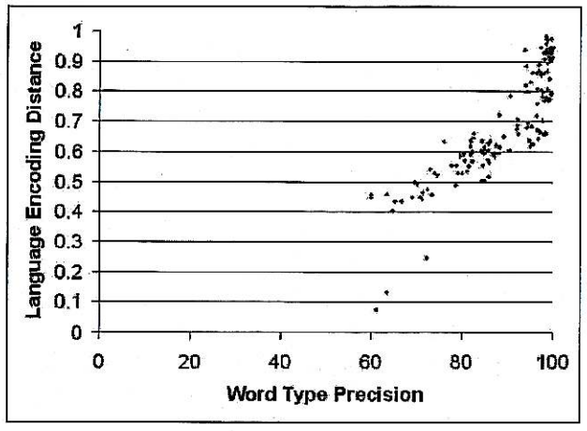

Figure 3. Precision vs. Language-Encoding Distance

For word identification, we have calculated the precision for type as well as token. We considered two cases here, one when the languages in the document are known in advance and the other when they are not. The second case gives the combined precision for language enumeration and word identification. The total number of types tested on (in all documents) was 2447424, out of which 2225079 (90.92%) were correctly tagged. The total number of tokens tested on was 3976754, out of which 3451698 (86.80%) were correctly tagged. The results are shown in Table 3. The results show that word identification for unknown languages is better when we take the top 2 instead of the top three language-encodings during the enumeration stage. This is because the top 3 usually include a language-encoding similar to the one dominant in the document (in terms of proportion) and this makes word identification difficult.

We tried to study the relationship between precision and language-encoding distance as defined in section 7. The precision has been plotted against normalized distance in Figure 3. It clearly shows that the problem is harder for closer language-encodings.

9. Future Directions

There can be irresolvable ambiguity in identification because the same word may belong to many languages and those languages may be using the same encoding. This makes segment identification very difficult. We plan to explore ways to overcome this difficulty. For segment token identification, we can also take the context into account. One way to do this will be to find out places where sudden drops in n-gram based probabilities of word sequences occurs. We also plan to modify our system so that it can make use of any available prior information about the scripts and the encodings (*e.g.*, when the documents are in Unicode), the languages, the charsets and the fonts (Shusha font maps to Hindi with Shusha Phonetic encoding) from the tags in web pages, etc.

10. Conclusion

We explored the problem of language and encoding identification in multilingual documents, including crucial assumptions such as global and local diversity assumption etc. We divided the problem into three parts: (a) monolingual identification, (b) language enumeration and (c) segment identification. A method for (b) and (c) was presented. We performed a fairly extensive evaluation on bilingual documents. Enumeration precision was calculated for the case where two out of three language-encodings were correctly identified (96.20%) and also for the case when both were correctly idetified (86.94%). Word token and type precision was calculated when the language-encodings in the document were known (86.80% and 90.91%) and also when they were not known (76.82% and 80.73). The results are promising, but have scope for improvement. We also discussed language distance in the context of electronic documents and showed that it has to be defined differently for electronic text. We found the precision to be higher for distant language-encodings.

References

ARTEMENKO O., MANDL T., SHRAMKO M. and WOMSER-HACKER C. (2006), "Evaluation of a Language Identification System for Mono- and Multilingual Text Documents", in *Proceedings of the 13th Annual Workshop on Selected Areas in Cryptography*, Dijon.

BEESLEY K. (1988), *Language identifier: A computer program for automatic natural-language identification on on-line text.*

CAVNAR W. B. and TRENKLE J. M. (1994), "N-Gram-Based Text Categorization", in *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*: 161-175.

DYEN I., KRUSKAL J. and BLACK P. (1992), "An Indo-European classification: A lexicostatistical experiment", in *Transactions of the American Philosophical Society, 82:1-132.*

ELLISON T. M. and KIRBY S. (2006), "Measuring Language Divergence by Intra-Lexical Comparison", in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Sydney.

GIGUET E. (1995), "Multilingual Sentence Categorisation According to Language", in *Proceedings of the European Chapter of the Association for Computational Linguistics, SIGDAT Workshop, From Text to Tags: Issues in Multilingual Language Analysis, Dublin.*

GORDON R. G. (2005), *Ethnologue: Languages of the World, Fifteenth edition (ed.)*, Online version: http://www. ethnologue.com/web.asp.

INGLE N. C. (1976), "A Language Identification Table", in *The Incorporated Linguist, 15(4).*

KIKUI G. (1996), "Identifying the Coding System and Language of On-line Documents on the Internet.", in *COLING*: 652-657.

NERBONNE J. and HEERINGA W. (1997), "Measuring dialect distance phonetically", in *Proceedings of SIGPHON-97: 3rd Meeting of the ACL Special Interest Group in Computational Phonology.*

PRAGER J. M. (1999), "Linguini: Language Identification for Multilingual Documents", in *Proceedings of the 32nd Hawaii International Conference on System Sciences.*

SIBUN P. and REYNAR J. C. (1996), "Language Identification: Examining the Issues", in *In Proceedings of SDAIR-96, the 5th Symposium on Document Analysis and Information Retrieval.*: 125-135.

SINGH A. K. (2006), "Study of Some Distance Measures for Language and Encoding Identification", in *Proceedings of ACL 2006 Workshop on Linguistic Distance*, Sydney.

STEPHEN J. (1993), "Solving the Problem of Language Recognition", in *Technical Report*, School of Computer Studies, University of Leeds.

SWADESH M. (1952), "Lexico-dating of prehistoric ethnic contacts", in *Proceedings of the American philosophical society, 96(4).*

TAN C., LEONG P. and HE S. (1999), "Language Identification in Multilingual Documents", in *Int'l Symp. Intelligent Multimedia and Distance Education.*

TEAHAN W. J. and HARPER D. J. (2001), "Using Compression Based Language Models for Text Categorization", in *J. Callan, B. Croft and J. Lafferty (eds.), Workshop on Language Modeling and Information Retrieval.*

CLEANEVAL