Mining a News Archive for a Comparable Corpus

Atelach Alemu Argaw

January 12, 2005

1 Introduction

The emergence of the World Wide Web has provided new and important opportunities to easily access and combine data and information from several different sources and thereby enabling the construction of new resources for researchers everywhere. In this paper we describe recent work that we have done in order to construct a comparable corpus consisting of Amharic and English news texts. In doing so, we utilised a number of available software and data sources that were (mainly) found on the Internet. The Amharic and English news texts were collected from Walta Information Center, a private news agency located in Addis Ababa, Ethiopia, that makes daily news in Amharic and English available through their web page http://www.waltainfo.com. Although the center provides Ethiopian news in both Amharic and English, only a small portion of the articles in the archive refer to the same event. Thus a major part of the work consisted of identifying the relevant comparable news items in the archive and aligning them with their respective corresponding version.

The work is motivated by the fact that Amharic is a language for which very few computational linguistic tools or corpora (such as part-of-speech taggers, parsers or tree-banks) exist. A solution to this problem (which is shared by a number of so called ”low density languages”) is through the transfer of linguistic knowledge, tools and techniques from languages for which linguistic resources are more developed and more readily available. There has recently been a number of research projects with this focus (see e.g. [14], [6], [11], [3] and [2]). It is our hope that the availability of even a small parallel Amharic - English corpus can provide a starting point for such transfer as well as provide a useful basis for a number of other higher level natural language processing activities for Amharic.

In order to align the Amharic and English news articles, we used the Levenshtein Distance algorithm to find fuzzy matches for the words in the title of articles that were published on the same date and that occurred in the same place. The performance of the matching algorithm was evaluated and experimental results for news items from the Gregorian year 2001 are presented. Depending on
match threshold values, the precision of the matching algorithm varies between 1.0 with a very low recall and 0.43 with a high recall.

2 Background

2.1 Web Mining

The Internet has so far been predominated by English. Nevertheless, it shows great promise as a source of multilingual content, due to the fact that texts in more and more languages are becoming available on the WWW in recent years. Resnik [9] uses figures from the Babel survey of multilinguality on the Web (http://www.isoc.org/) and presents estimated figures that as of June, 1997, there were on the order of 63000 primarily non-English Web servers, ranging over 14 languages. A follow-up investigation of the non-English servers suggests that nearly a third contain information expressed in more than one language.

A number of researchers have implemented web mining algorithms and tools to construct bilingual corpora from the web c.f. [5], [7], [13], [8], [9], [10]. The most common approaches in these systems have been to use structural resemblance or content analysis, or a combination of the two. One example is the STRAND (Structural Translation Recognition for Acquiring Natural Data) system developed by Philip Resnik and colleagues [8], [9].

STRAND uses structural filtering to compare language pairs, linearizing the HTML structure of both documents and aligning the resulting sequences. STRAND’s approach is to identify naturally occurring pairs of Web pages in parallel translation. STRAND locates pages that might be translations of one another via a number of different strategies, and filters out page pairs where the page structures diverge by too much. To attain this, it exploits an observation about the way that Web page authors disseminate information in multiple languages: that when presenting the same content in two different languages, authors exhibit a very strong tendency to use the same document structure. Hence, STRAND is based on the insight that translated Web pages tend quite strongly to exhibit parallel structure, permitting their exploitation even without looking at content.

The original STRAND architecture used the AltaVista search engine to accomplished the first step by searching for two types of Web pages. A parent page (one that contains hypertext links to different language versions of a document) and a sibling page (a page in one language that itself contains a link to a version of the same page in another language). When considering only parent or sibling pages, pairing up potentially translated pages is simply done by pairing the child or sibling pages. When all the pages on a site are under consideration, it is done by URL-matching which exploits the fact that the directory structure on many Web sites reflects parallel organization when pages are translations of each other. Another possible criterion the authors consider for matching is the use of document lengths and the fact that texts which are translations of each other tend to be similar in length.
STRAND has been used to mine bilingual documents from the web for language pairs such as English-French, English-Spanish and the authors claim that rigorous evaluation using human judges suggests that the technique produces an extremely clean corpus - noise estimated between 0 and 8% even without human intervention [9].

In [10] STRAND is enhanced with content based similarity measures and applied over the Internet Archive (www.archive.org) to obtain an English-Arabic parallel corpus of more than 1M tokens per language, with a precision of 0.95 and a recall of 0.99 over the extracted candidate pairs.

2.2 Amharic

Amharic is the official government language spoken in Ethiopia. It is a Semitic Language of the Afro-Asiatic Language Group that is related to Hebrew, Arabic, and Syrian. Amharic, the syllabic language, uses a script which originated from the Ge’ez alphabet (the liturgical language of the Ethiopian Orthodox Church). The language has 33 basic characters with each having 7 forms for each consonant-vowel combination, and extra characters that are consonant-vowel-vowel combinations for some of the basic consonants and vowels. It also has a unique set of punctuation marks and digits. Unlike Arabic, Hebrew or Syrian, the language is written from left to right. Amharic alphabets are one of a kind and unique to Ethiopia (see Figure 1).

According to the 1998 census in (Arthur Lynn.s World Languages) Amharic is spoken widely throughout different regions of Ethiopia: by over 17 million people as a first language and by over 5 million second language users. Some estimates indicate that Amharic is the mother-tongue of around 15 to 30 million Ethiopians. Manuscripts in Amharic are known from the 14th century and the language has been used as a general medium for literature, journalism, education, national business and cross-communication. A wide variety of literature including religious writings, fiction, poetry, plays, and magazines are available in the language.

Processing Amharic language using computers has become common for more than a decade. A growing number of people these days use computer systems for processing the language. People, for instance, use computers for various purposes: doing document writing and correction, storage and retrieval of Amharic texts and databases. Hence the production of more and more documents (information) and databases to be made available in the language, in electronic form.

Amharic documents written in Ethiopic are available on the web, and the amount is increasing by the day. Ethiopic or Ethiopian script refers to the Ge’ez alphabet, and is the official writing system of Ethiopia. Different character encoding schemes and different keyboard layout are used to represent Ethiopic electronically. Some are unicode compliant (e.g. Visual Ge’ez, Ethiopia Jiret) while some are not. Although much effort has been made to have a standard for Ethiopic encoding, so far there is no standard and texts written in these different encoding schemes are not compatible with one another.
Figure 1: The Amharic alphabet (Fidel) from http://www.omniglot.com/
Although the amount of Amharic text on the web is growing in size, the availability of an equivalent translation of the texts in another language is still very limited.

2.3 The Ethiopian Calendar

The Ethiopian calendar runs approximately seven years and eight months behind the Gregorian calendar (the current year 2005 is 1997 in the Ethiopian calendar). The calendar is divided into 12 months of 30 days each and one 13th month consisting of five or six days depending on whether the current year is a leap year. The Ethiopian new year starts on September 11 (or September 12 on Gregorian leap years). A calendar conversion table was extracted from a free trial version of the 7000 Years Calendar v1.4.1[1]. This table was used to convert the dates from one calendar to the other and use the information to filter out the articles written on the same date during the alignment process.

2.4 Parallel texts

Parallel corpora play an important role in machine translation and multilingual natural language processing. They represent resources for automatic lexical acquisition, they provide indispensable training data for statistical translation models, and they can provide the connection between vocabularies in cross-language information retrieval [10]. Recently, the trend to utilize parallel corpora to transfer linguistic resources from resource-rich languages such as English onto lesser supported languages has been referred to as "Cross-Language Projection" [14]. Comparable corpora on the other hand could be used as resources for constructing parallel corpora, as well as resources for different corpus based linguistic, natural language processing and information retrieval experiments.

3 Preprocessing

3.1 The Data Set

There are very few parallel/comparable Amharic - English news texts available on the Internet. While there are a few cites (such as Ethiopian News Headlines http://www.ethiozena.net/) that contain Ethiopian News in Amharic only, and others (such as Addis Tribune http://www.addistribune.com) that contain Ethiopian News in English, we have only been able to find one, at the Walta information center (http://www.waltainfo.com) that contains a substantial amount of comparable versions of both Amharic and English News with URLs that give a clue for automatic alignment.

Not all texts at this site have a corresponding comparable version, but a portion of the news texts here describe the same event and are comparable (but not direct) translations of each other. These comparable news items are relatively few and hard to identify automatically but are in general archived under the corresponding dates. The Ethiopic news are archived using the Ethiopian calendar.
while the English news are using the Gregorian calendar, but most comparable news stories can be found within a couple of days from the corresponding date.


3.2 Downloading

We downloaded all English and Amharic news articles available at the Walta Information Center archives using a web crawler (an evaluation version of Offline Explorer Pro 3.5 from MetaProductsSoftware Corporation (www.metaproducts.com)). Since the pages at the archive contain a large number of links to other web pages that were irrelevant for our purposes, it was important to be able to filter and control exactly what was being retrieved.

The news texts at the Walta archive are structured in folders for Amharic news and English news with subfolders for each year, month and day respectively. The Amharic news is archived under folders according to the Amharic calendar while the English news is stored in folders with names according to the Gregorian calendar so a matching pair of news items would for example be stored under:

http://www.waltainfo.com/EnNews/2002/feb/01Feb02/Feb1e8.htm

and


respectively. A particular day would typically contain between five and ten separate news items in each language. Since the file names did not contain sufficient information to identify the date (information about the year is missing) we downloaded all available English and Amharic articles from the archive while retaining the original folder structure.

3.3 HTML tag removal

When the file structure for the relevant news articles had been downloaded it was then flattened and the html code for each page was removed using a publicly available freeware (Emsa HTML Tag Remover v1.0 Build 20). This software allows for controlled removal of html tags as well as whitespace and other special characters from html files. The extra degree of control was required
3.4 Transliteration

In order to simplify the analysis and matching of news texts and to have a unified representation of the Amharic texts, we decided to transliterate all Amharic texts into SERA [12].

The Ethiopic script in the Amharic texts are represented using a variety of fonts. For the Amharic years 1993 until the first half of 1996, Visual Geez 2000 was the most common, while after that a mixture of fonts have been used, which complicated the transliteration step.

The transliteration was done using a file conversion utility called g2 which is available in the LibEth package (LibEth is a library for Ethiopic text processing written in ANSI C http://libeth.sourceforge.net/). g2 was made available to us by Daniel Yacob of the Ge’ez Frontier Foundation (http://www.ethiopic.org/).

3.5 Restructuring

Most of the Amharic and English news texts have a semi-structured format that includes title, place, date, newsagency, and body. In order to simplify the matching of news texts, we have preprocessed the news articles and stored them in an xml structure that identifies each of these fields separately. Figures 2 and 3 show (parts of) an Amharic and English news text describing the same event.

4 Alignment

The news articles at Walta Information center are more comparable than parallel. There are no direct translations, it appears that the Amharic and English news are written by different reporters. The ones we tried to match in this experiment are those that describe the same incident. The date the articles are written, the place the incident occurred and the title of the articles are the major information source used for the alignment. The basic assumption here is based on the fact that titles tend to be a highly summarised version of the news text and tend to have content words that are nouns or noun phrases which in turn are usually place names, person names, organization names, or dates, numbers etc. From the outset it would appear as if the Amharic and English titles are very different, since they use different scripts. But once the Amharic version is transliterated, it becomes more apparent that many nouns and proper names are in fact quite similar. This information can be used to match parallel Amharic English text without using any lexical resource which is hardly available for this language pair. Even if it was available, it is very unlikely that the
The Debub University in Awassa disclosed that it had admitted some 730 students in seven new field of studies it launched at degree and diploma levels during the current academic year. Academic and Research Vice president of the University, Dr. Tesfaye Teshome told WIC that the new fields of studies include Accounting, Economics, Management, Language, Chemistry, Physics, Biology and Mathematics.

Figure 3: An English news text in xml-format
An algorithm was designed and implemented to automatically align documents that are comparable. The basic idea is to retrieve the date information from the Amharic news articles, get the corresponding Gregorian date from the calendar conversion list, and match the English news articles with that same date. From the set of English articles with the same date, it gets the place information and matches those which have similar names with that of the Amharic article under consideration. From the set of English articles with matching place names, it matches each word in the Amharic article’s title with each word in the English article’s title, and retrieves the one with the best edit distance score above a match threshold specified by the user as the best match. The number of words in the Amharic news title that have a match in the corresponding English title are then counted and divided by the total number of words in the longer title. See Appendix I for the perl implementation. The fuzzy string matching was done using a perl implementation of Levenshtein Distance\cite{4}.

Levenshtein distance (LD) is named after the Russian scientist Vladimir Levenshtein, who devised the algorithm in 1965. The metric is also sometimes called edit distance. Levenshtein distance is a measure of the similarity between two strings. The distance is the number of deletions, insertions, or substitutions required to transform a source string into a target string. For example If the source (s) is ”addis” and the target (t) is ”addis”, then LD(s,t) = 0, because no transformations are needed. The strings are already identical. If s is ”adis” and t is ”addis”, then LD(s,t) = 1, because one insertion is sufficient to transform s into t. The greater the Levenshtein distance, the more different the strings are. In our implementation, to calculate the edit distance score, we divided the Levenshtein distance for a word, by the length of the same word, in order to normalise and compensate for the fact that longer words tend to have a larger edit distance.

5 Results and Discussion

For the experiments reported here news articles from the Gregorian year 2001 has been selected.

The experiments were conducted with a varied set of parameters. Different parameter settings of title match threshold (for example 0.25 would restrict the match in such a way that at least 25% of the words in the title must have an edit distance match) and word match threshold (for example 0.5 would restrict the word match to have 50% or better edit distance match).

The 2001 experiments were conducted on a data set consisting of 1923 English news articles published during the year 2001 (according to the Gregorian calendar), and 1219 Amharic articles published during the time interval between the 7th month of 1993 and the 4th month 1994 (Ethiopian calendar). The Amharic articles corresponding to 2001 should have started from the 5th month of 1993, but what is available in the archive starts from the 7th month of 1993.
Figure 4: Examples of some matching news items
Table 1: Results for the 2001 data set

<table>
<thead>
<tr>
<th>Title threshold</th>
<th>Word threshold</th>
<th>Matches</th>
<th>Correct</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.15</td>
<td>0.4</td>
<td>153</td>
<td>89</td>
<td>0.58</td>
</tr>
<tr>
<td>0.20</td>
<td>0.4</td>
<td>44</td>
<td>33</td>
<td>0.75</td>
</tr>
<tr>
<td>0.25</td>
<td>0.4</td>
<td>13</td>
<td>12</td>
<td>0.92</td>
</tr>
<tr>
<td>0.30</td>
<td>0.4</td>
<td>5</td>
<td>5</td>
<td>1.00</td>
</tr>
<tr>
<td>0.35</td>
<td>0.4</td>
<td>2</td>
<td>2</td>
<td>1.00</td>
</tr>
<tr>
<td>0.15</td>
<td>0.5</td>
<td>225</td>
<td>127</td>
<td>0.56</td>
</tr>
<tr>
<td>0.20</td>
<td>0.5</td>
<td>77</td>
<td>57</td>
<td>0.74</td>
</tr>
<tr>
<td>0.25</td>
<td>0.5</td>
<td>30</td>
<td>29</td>
<td>0.97</td>
</tr>
<tr>
<td>0.30</td>
<td>0.5</td>
<td>13</td>
<td>13</td>
<td>1.00</td>
</tr>
<tr>
<td>0.35</td>
<td>0.5</td>
<td>3</td>
<td>3</td>
<td>1.00</td>
</tr>
<tr>
<td>0.20</td>
<td>0.6</td>
<td>302</td>
<td>129</td>
<td>0.43</td>
</tr>
<tr>
<td>0.25</td>
<td>0.6</td>
<td>122</td>
<td>74</td>
<td>0.61</td>
</tr>
<tr>
<td>0.30</td>
<td>0.6</td>
<td>62</td>
<td>50</td>
<td>0.81</td>
</tr>
<tr>
<td>0.35</td>
<td>0.6</td>
<td>38</td>
<td>35</td>
<td>0.92</td>
</tr>
</tbody>
</table>

The result of the matching was manually evaluated in order to calculate the precision values. Recall has not been calculated but we have done an estimate for a randomly selected 15 days of data (February 1 - 15, 2002). The amount of matching news articles in this subset is 10 out of 98 or approximately 10%. In conducting the experiments, we aimed at finding corresponding comparable English articles for at least 10% of the total amount of Amharic articles. Examples of some news items with matching titles are shown in Figure 4.

As can be seen from the results reported in Table 1 above, the more constraints there are, the better the precision of the matches is, at a cost of very limited recall. When the word threshold was set to 0.5, and 0.4 the precision was 100% for title threshold values of 0.3 and 0.35. These same title threshold values give a lesser precision and better recall with a less constrained word threshold value of 0.6. With a more constrained word threshold value of 0.4, all experiments show an increase in precision and decrease in recall compared to the experiments with word threshold values of 0.5 and 0.6.

While conducting the alignment experiments, the words that are returned as closest matches could be used in further alignment experiments. Some examples of the correctly paired ones are given below in Table 2.

5.1 Conclusion

In this paper we have presented a report on experiments conducted to align Amharic-English comparable news texts that are available on the WWW. Promising results have been obtained that encourage further investigation and improve-
<table>
<thead>
<tr>
<th>Score</th>
<th>Amharic</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1667</td>
<td>dolar</td>
<td>Dollar</td>
</tr>
<tr>
<td>0.2</td>
<td>abeba</td>
<td>Ababa</td>
</tr>
<tr>
<td>0.25</td>
<td>ministEr</td>
<td>Ministry</td>
</tr>
<tr>
<td>0.125</td>
<td>ministr</td>
<td>Ministry</td>
</tr>
<tr>
<td>0.375</td>
<td>mini’strna</td>
<td>Ministry</td>
</tr>
<tr>
<td>0.25</td>
<td>yeinternEt</td>
<td>Internet</td>
</tr>
<tr>
<td>0.2222</td>
<td>instityut</td>
<td>Institute</td>
</tr>
<tr>
<td>0.2857</td>
<td>beoromiya</td>
<td>Oromiya</td>
</tr>
<tr>
<td>0.3333</td>
<td>polis</td>
<td>Police</td>
</tr>
<tr>
<td>0.2857</td>
<td>kefEdEral</td>
<td>Federal</td>
</tr>
<tr>
<td>0.2857</td>
<td>kabinE</td>
<td>Cabinet</td>
</tr>
<tr>
<td>0.375</td>
<td>kuntal</td>
<td>Quintals</td>
</tr>
<tr>
<td>0.2</td>
<td>adis</td>
<td>Addis</td>
</tr>
<tr>
<td>0.1429</td>
<td>kapital</td>
<td>Capital</td>
</tr>
<tr>
<td>0.3333</td>
<td>polisi</td>
<td>Policy</td>
</tr>
<tr>
<td>0.2857</td>
<td>mesfn</td>
<td>Mesfin</td>
</tr>
<tr>
<td>0.2</td>
<td>yeinvestment</td>
<td>Investment</td>
</tr>
<tr>
<td>0.3333</td>
<td>polisi</td>
<td>Policy</td>
</tr>
<tr>
<td>0.3333</td>
<td>yemetema</td>
<td>Metema</td>
</tr>
<tr>
<td>0.2222</td>
<td>prEzidant</td>
<td>President</td>
</tr>
<tr>
<td>0.1429</td>
<td>projekt</td>
<td>Project</td>
</tr>
<tr>
<td>0.3333</td>
<td>dEmokrasi</td>
<td>Democracy</td>
</tr>
<tr>
<td>0.1667</td>
<td>tarif</td>
<td>tariff</td>
</tr>
<tr>
<td>0.3</td>
<td>yuniversti</td>
<td>University</td>
</tr>
<tr>
<td>0.1429</td>
<td>isayas</td>
<td>Issayas</td>
</tr>
<tr>
<td>0.2</td>
<td>ambasader</td>
<td>Ambassador</td>
</tr>
<tr>
<td>0.2</td>
<td>kuma</td>
<td>Kuma</td>
</tr>
</tbody>
</table>

Table 2: Some Amharic and English words and their edit distance score
ment of the methods employed.

Since the Amharic and English news under consideration were comparable rather than parallel (direct translations of one another), the method used did not take length into consideration. We believe that under these circumstances, the length could be more confusing than helpful in automatically aligning the articles. Although this may hold true for this dataset, for alignment of potentially parallel data, the length information could contribute immensely.

The body of the text (the news article) could also be considered in the alignment process supplementarily instead of using the title only. There are many occurrences of names and digits that would help in aligning the news articles in the text body as well. When considering the text body, resources such as lexica (e.g. for word by word translation), stop word list (to remove non content bearing words that appear in many of the articles), morphological analysis or steming (to consider the root word only) etc may be used.

Machine learning approaches could also be implemented to learn from known matching words in the two languages from previous experiments so that an advanced weighting scheme can be used during the fuzzy matching. Some sort of a knowledge base that would give supplemental information such as place name heirarchy, both digit and letter representations of numbers, abbreviations, etc can also be included.

Taking the above mentioned suggestions in mind, we plan to further enhance the system and experiment with more data, with the intention of getting better precision and recall and building a parallel/comparable corpus for the language pair Amharic and English.

References


Appendix I: Perl Code

```perl
$i = 0;
$cnt = 0;
$cntEn = 0;
print("Enter the title threshold value \- you can start with 0\.25\n");
$TitleThresh = <STDIN>;
chop ($TitleThresh);
print("Enter the word match threshold value \- you can start with 0\.5\n");
$WordThresh = <STDIN>;
chop ($WordTresh);

open(CALANDER,"Amh-Eng-Date-Conversion.txt") or die;
while(<CALANDER>) {
    chomp(my $line=$_);
    if ($line=~/\d+\-\d+\-\d+\s\d+\-\d+\-\d+/) {
        my $x = $1;
        my $y = $2;
        $i++;
        $Calender{$i}{AmhD} = $x;
        $Calender{$i}{EngD} = $y;
    }
}
open(MATCHT,">results01.txt") or die ("Error opening output file\n");
open(AMHFILE,"AmhNews01.txt") or die;
while(<AMHFILE>) {
    chomp(my $Amh=$_);
    if ($Amh =="/<document>/") {
        $cnt++;
    }
    if($Amh="/<file name="/(["\]+)"/") {
        $Amhhash{$cnt}{filename} = $1;
    }
    if($Amh="/<amhdate date="/(["\]+)"/") {
        $Amhhash{$cnt}{AmhDate} = $1;
    }
    if($Amh="/<title>(.*)") {
        ...
    }
```
$Amhhash{$cnt}{title} = $1;
}
if($Amh="<dateline place="([\"\"]+)\"/>
{
$Amhhash{$cnt}{place} = $1;
}
}

open(ENGFILE,"EngNews01.txt") or die;
while(<ENGFILE>) {
  chomp(my $Eng=$_);
  if ($Eng="/<document>/")
  {
    $cntEn++;
  }
  if($Eng="/<file name="([\"\"]+)\"/>
  {
    $Enghash{$cntEn}{filename} = $1;
  }
  if($Eng="/<gregdate date="([\"\"]+)\"/>
  {
    $Enghash{$cntEn}{EngDate} = $1;
  }
  if($Eng="/<amhdate date="([\"\"]+)\"/>
  {
    $Enghash{$cntEn}{RefAmhDate} = $1;
  }
  if($Eng="/<title>(.*)/"
  {
    $Enghash{$cntEn}{title} = $1;
  }
  if($Eng="/<dateline place="([\"\"]+)\"/>
  {
    $Enghash{$cntEn}{place} = $1;
  }
}

for($l = 1; $l <= $cnt; $l++)
{
  for($z = 1; $z <= $cntEn; $z++)
  {
    if ($Amhhash{$l}{AmhDate} eq $Enghash{$z}{RefAmhDate})
    {
      my $s1 = $Amhhash{$l}{place};
    }
  }
}
my $s2 = $Enghash{$z}{place};
$splacematch =lerenshtein($s1, $s2);
if ($splacematch < 3)
{
$AmhTitle = $Amhhash{$l}{title};
$EngTitle = $Enghash{$z}{title};
$score = titlematch($AmhTitle, $EngTitle);
if ($score > $TitleThresh)
{
    print "$Enghash{$z}{filename}	$Enghash{$z}{title}	
$Amhhash{$l}{filename}	$Amhhash{$l}{title}\n ";
    printf MATCHT "$Enghash{$z}{filename}	$Enghash{$z}{title}	
$Amhhash{$l}{filename}	$Amhhash{$l}{title}\n ";
}
}
}
} ## for z
}
} ## for l

sub titlematch
{
    my ($t2, $t1) = @_
    @tempEngT = split(/\s+/, $t1);
    @tempAmhT = split(/\s+/, $t2);
    $matchcnt = 0;
    $x = @tempAmhT;
    $y = @tempEngT;
    for($d = 0; $d<$x; $d++) {
        $minmatch = 1;
        $mys1 = $tempAmhT[$d];
        $AmhLen = length ($mys1);
        for($e = 0; $e < $y; $e++) {
            $mys2 = $tempEngT[$e];
            $EngLen = length ($mys2);
            $titlematch = levenshtein($mys1, $mys2);
            #
            print "$titlematch\t$mys1\t$mys2\n";
            if ((($AmhLen == 0) || ($EngLen == 0))
            {
                $AmhLen = 1;
                $EngLen = 1;
            }
        }
    }
}

17
if ($AmhLen > $EngLen)
{
    $titleNewM = $titlematch / $AmhLen;
} else
{
    $titleNewM = $titlematch / $EngLen;
}
#endif MATCHT "$titleNewM\t$mys1\t$mys2\n";
if ($titleNewM < $minmatch)
{
    $minmatch = $titleNewM;
    $bestword = $mys2;
}
} ## for inner
if ($minmatch < $WordThresh)
{
    $matchcnt++;
    $cnthash{$matchcnt}{Amh} = $mys1;
    $cnthash{$matchcnt}{Eng} = $bestword;
    $cnthash{$matchcnt}{wdscore} = $minmatch;
}
} ## for outer
# print "$matchcnt\t";
if ($x >= $y)
{
    $titlescore = $matchcnt / $x;
    print "$titlescore\t$matchcnt\t$x\n";
} else
{
    $titlescore = $matchcnt / $y;
}
return $titlescore;
} ## end sub titlematch

# Return the Levenshtein distance (also called Edit distance)
# between two strings
#
# The Levenshtein distance (LD) is a measure of similarity between two
# strings, denoted here by s1 and s2. The distance is the number of
# deletions, insertions or substitutions required to transform s1 into
# s2. The greater the distance, the more different the strings are.
#
# The algorithm employs a proximity matrix, which denotes the
# distances between substrings of the two given strings. Read the
# embedded comments for more info. If you want a deep understanding
# of the algorithm, print the matrix for some test strings
# and study it
#
# The beauty of this system is that nothing is magical - the distance
# is intuitively understandable by humans
#
# The distance is named after the Russian scientist Vladimir
# Levenshtein, who devised the algorithm in 1965
#
sub levenshtein
{
    # $s1 and $s2 are the two strings
    # $len1 and $len2 are their respective lengths
    #
    my ($s1, $s2) = @_;
    $s1 = lc($s1);
    $s2 = lc($s2);

    my ($len1, $len2) = (length $s1, length $s2);

    # If one of the strings is empty, the distance is the length
    # of the other string
    return $len2 if ($len1 == 0);
    return $len1 if ($len2 == 0);

    my %mat;

    # Init the distance matrix
    #
    # The first row to 0..$len1
    # The first column to 0..$len2
    # The rest to 0
    #
    # The first row and column are initialized so to denote distance
    # from the empty string
    #
    for (my $i = 0; $i <= $len1; ++$i)
    {
        
        
    }
for (my $j = 0; $j <= $len2; ++$j)
{
    $mat{$i}{$j} = 0;
    $mat{0}{$j} = $j;
}

$mat{$i}{0} = $i;
}

# Some char-by-char processing is ahead, so prepare
# array of chars from the strings
#
my @ar1 = split(//, $s1);
my @ar2 = split(//, $s2);

for (my $i = 1; $i <= $len1; ++$i)
{
    for (my $j = 1; $j <= $len2; ++$j)
    {
        # Set the cost to 1 iff the ith char of $s1
        # equals the jth of $s2
        #
        # Denotes a substitution cost. When the char are equal
        # there is no need to substitute, so the cost is 0
        #
        my $cost = ($ar1[$i-1] eq $ar2[$j-1]) ? 0 : 1;

        # Cell $mat{$i}{$j} equals the minimum of:
        #
        # - The cell immediately above plus 1
        # - The cell immediately to the left plus 1
        # - The cell diagonally above and to the left + the cost
        #
        # We can either insert a new char, delete a char of
        # substitute an existing char (with an associated cost)
        #
        $mat{$i}{$j} = min([$mat{$i-1}{$j} + 1,
                           $mat{$i}{$j-1} + 1,
                           $mat{$i-1}{$j-1} + $cost]);
    }
}

# Finally, the distance equals the rightmost bottom cell
# of the matrix
#
# Note that $mat{$x}{$y} denotes the distance between the
# substrings 1..$x and 1..$y
# return $mat{$len1}{$len2};

# minimal element of a list
# sub min
{
    my @list = @{$_[0]};
    my $min = $list[0];
    
    foreach my $i (@list)
    {
        $min = $i if ($i < $min);
    }
    
    return $min;
}

close(CALANDER);
close(AMHFILE);
close(ENGFILE);
close(MATCHT);