AutoCor: A Query Based Automatic Acquisition of Corpora of Closely-related Languages *

Davis Muhajereen D. Dimalen[a], Rachel Edita O. Roxas[b]

[a] Information Technology Department, School of Computer Studies
Mindanao State University-Iligan Institute of Technology, Tibanga, Iligan City
d_dimalen@yahoo.com
[b] College of Computer Studies, De La Salle University-Manila,
roxasr@dlsu.edu.ph

Abstract. AutoCor is a method for the automatic acquisition and classification of corpora of documents in closely-related languages. It is an extension and enhancement of CorpusBuilder, a system that automatically builds specific minority language corpora from a closed corpus, since some Tagalog documents retrieved by CorpusBuilder are actually documents in other closely-related Philippine languages. AutoCor used the query generation method odds ratio, and introduced the concept of common word pruning to differentiate between documents of closely-related Philippine languages and Tagalog. The performance of the system using with and without pruning are compared, and common word pruning was found to improve the precision of the system.

Keywords: document acquisition, document classification.

1. Introduction

A corpus is a term used to designate a body of authentic language data that can be used as a basis for linguistic research.[1] It is also applied to a body of language texts that exist in electronic format. It is estimated that there are currently over 4 billion pages on the world wide web (WWW) covering most areas of human endeavor. And as more information are becoming electronically available on the web, we need more effective methods and techniques to access these information. To date, there has been limited effort in taking advantage of this available information on the web for building natural language resources especially for sparse languages (or minority languages) like Tagalog and other Philippine languages. Unfortunately, to manually collect and organize a language specific corpus over the Web is difficult. The process is tedious and time consuming. To add, an expert in linguistics is needed to manually determine the language where the document collected is written.

* This project is funded by the Philippine Council for Advanced Science and Technology for Research and Development, Department of Science and Technology, Philippine Government.

[1] Orasan, C. and R. Krishnamurthy 2000. An Open Architecture for the Construction and Administration of Corpora. *Proceedings of the Second International Conference on Language Resources and Evaluation*. pp. 793-800.

A system that automatically acquires language specific documents from the Web is one good solution in corpora building. Creating such a system requires knowledge in information retrieval and natural language processing.

2. Automatic Corpora Builder on a Closed and Open Corpus

Several components are required for an automatic corpora builder: a set of seed documents, a language modeler, a query generator, a web search engine, and a language filter[2].

The CorpusBuilder takes advantage of existing search engine database to collect documents from the web[3]. It iteratively creates new queries to build a corpus in a single minority language. Sets of relevant and non-relevant documents are taken as initial inputs. Relevant documents are those that belong to the target language, while non-relevant documents are other documents that belong to other languages. These documents are used as inclusion and exclusion terms for the query. The query is sent to the search engine and the document that has the highest rank will be retrieved. The document retrieved is processed through the language filter and classified as either relevant or non-relevant document. The newly classified set of documents is the product of the system and is the basis for the next term selection as the system iterates.

CorpusBuilder is a system that automatically builds a minority language corpus. An examination of this corpus showed that the corpus also contained documents in languages that are closely-related to the identified minority language. Specifically, there were documents retrieved that are closely-related Philippine languages to the identified minority language Tagalog. Thus, in this study, we considered the three most closely-related languages in the Philippines, Bicolano, Cebuano and Tagalog, as identified by Fortunato[4], that belong to the Austronesian family of languages. This can be explained by the fact that closely-related languages within the same family of languages exhibit common linguistic phenomena. For instance, there are several Bicolano, Cebuano and Tagalog words which are common to these languages as illustrated in Tables 1 to 3.

Table 1: Words Common to Tagalog and Cebuano.

Tagalog/ Cebuano	English
apo	grandchild
anak	son/daughter
bayaw	in-law
langgam (Tagalog)	ant
langgam (Cebuano)	bird
bangka	sailboat

Table 2: Words Common to Tagalog and Bicolano.

Tagalog/Bicolano	English
hayop	animal
tao	human
langit	heaven
pakpak	wings

[2] Ghani, R., R. Jones and D. Mladenic. 2001. Using the Web to Create Minority Language Corpora. *Proceedings of the 10th International Conference on Information and Knowledge Management.* pp. 279 – 286.

[3] Jones, R. and R. Ghani. 2000. Automatically Building a Corpus for a Minority Language on the Web. *In the Proceedings of the Annual Meeting of the Association of Computational Linguistics 2000.* pp. 29-36.

[4] Fortunato, F. T. 1993. *Mga Pangunahing Etnoling-guistikong Grupo sa Pilipinas.* Malate, Manila, Philippines: De La Salle University Press.

pinsan	cousin

Table 3: Words Common to Bicolano, Cebuano, and Tagalog.

Bicolano/Cebuano/Tagalog	English
agaw	snatch
bawi	snatch
kadena	chain
belen	manger

Thus, AutoCor considered closely-related languages rather than a single minority language, and used document classification using common word pruning which has shown to improve the precision of the system.

The corpus that was used in this research contains documents from the web. The corpus contains 4,000 documents, wherein the target or relevant documents were tagged correspondingly, having 250 documents each in Bicolano, Cebuano and Tagalog, and the rest of the documents functioned as the non-relevant documents were in English, Hungarian and Polish. The selection of the set of non-relevant documents was based on similar character sets and the availability of documents.

Figure 1 illustrates the overall architecture of AutoCor on a closed corpus. There are 5 main routines namely, the Language Modeler, Common Word Pruning, the Query Generator, Sampling, and finally the Document Classifier. Each routine is done in sequence. Initially the first routine (Language Modeler) requires initial seed documents for each of the selected closely-related languages (L) and for the other languages (OL). Each language in (L) and (OL) is denoted by the sets {L_1…L_n} and {OL_1…OL_n}, respectively. The “Initial Documents” is defined by the sets (iD_L) and (iD_{OL}) wherein (iD_L) is the set of initial documents in closely-related languages (L) and (iD_{OL}) is the set of initial documents in other languages (OL). The language models are composed of (LM_L) and (LM_{OL}) wherein (LML) is the set of language models for the closely-related languages (L) and (LM_{OL}) is the set of language models for the other languages (OL). The Pruned Language Models are the sets (PLM_L) for the closely-related languages (L) and (PLM_{OL}) for the other languages (OL). The output corpus is composed of a set of documents classified as closely related languages (D_L) and another set of documents classified as other languages (D_{OL}) wherein (D_L) is also equal to the set {D_{L1},D_{L2},…,D_{Ln}}. Documents are retrieved via Sampling from a Closed Corpus. The system works as follows:

a. Select one seed document each from the set of initial documents in iD_L and the set of initial document in iD_{OL}.
b. Using the seed or initial documents in the target language and other languages, build language models LM_L and LM_{OL} for each of the languages in L and OL.
c. Prune words that are common in the set of language models in LM_L and LM_{OL} and let the PLM_L be the set of pruned language models for L, and PLM_{OL} for OL.
d. Using Odds-ratio, inclusion and/or exclusion terms for the query are determined from PLM_L and PLM_{OL}, respectively.
e. Using the query generated, documents are sampled from the closed corpus that matches the query.
f. The documents retrieved are classified by using a language classifier. Decide whether to add the list of documents in the output corpus, and update the language models in LM_L and LM_{OL}.
g. Repeat step 1 until the stopping criterion is reached.

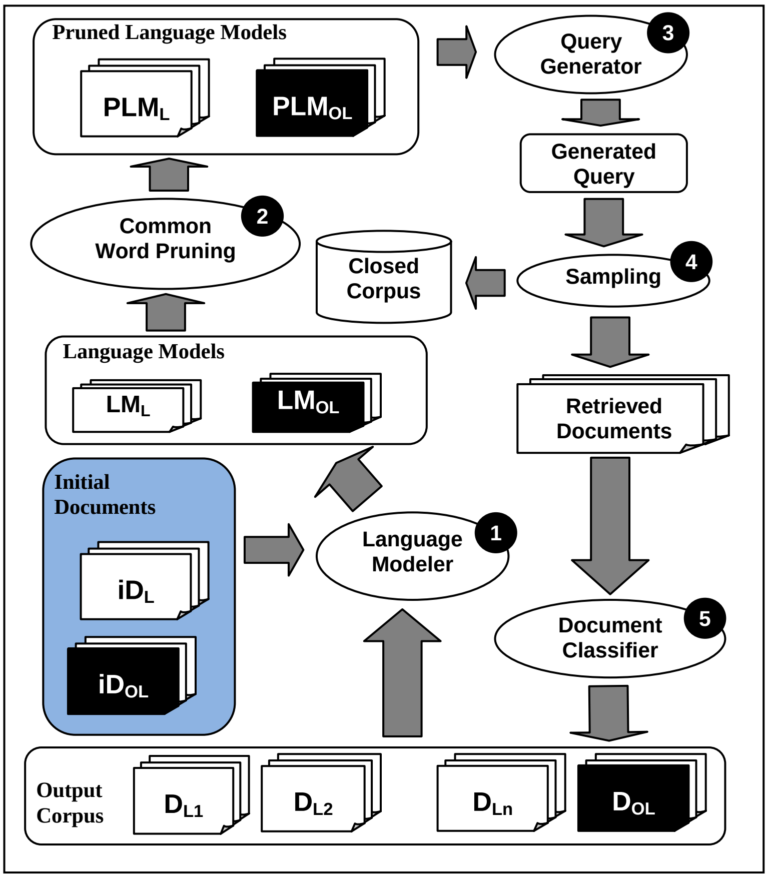

Figure 1: AutoCor on a Closed Corpus.

AutoCor repeats the process of language modeling, common word pruning, automatic query generating, document sampling and document classifying. Stopping criterion is user defined and depends on the number of queries that has to be generated.

AutoCor was extended to access documents from the Web. Information retrieval (IR) on the web poses more challenges as compared to classical IR due to the bulk of information that is available on the web, the heterogeneity of documents, variety of languages, duplication of information, documents having high linkages, ill-formed queries, wide variance of users and specific behavior of the users. The algorithm is similar with that of AutoCor on a closed corpus except that the resource where documents are retrieved is an open corpus or specifically the World Wide Web.

3. Language Modeling

We employed a statistical language modeler using the n-gram distribution-based language modeling. For general text, more training data will always improve a language model (LM). However, as training data size increases, LM size increases which can lead to models that are too large for practical use[5]. Training data is usually biased on its mixture of elements. An automatic language modeling system, that gets its training data from articles in the web recursively, would often process words that are not supposed to be present in the training set, thus, the effect of noise in the LM based on documents from the web must be minimized.

Count cut-off is commonly used to prune language models. The method removes from the LM those n-grams that occur less frequent in the training data, assuming they will be equally

[5] Gao, J. and K. Lee. 2000. Distribution-based Pruning of Backoff Language Models. *The 38th Annual Meeting of the Association for Computational Linguistics*. Hong Kong. pp. 579 – 588.

infrequent in all test data. Also, the count cut-off intensifies the bias of the training data. For instance, if we use the bible in training, a word like “sin” may have high frequency in certain chapters but not others. Thus, “sin” can be cut-off in some chapters[5]. These are domain specific issues.

The training set representing a specific language will be processed by a profile generator to generate a profile which will be used for text categorization (see section 5.1 and 5.2). The generation of profile is part of the n-gram distribution based language modeling process.

4. Common Word Pruning

Pruning language models keeps word n-grams that are more likely to occur in a given document. Early language modeling algorithms remove words that are likely to be infrequent in a test data[5]. AutoCor adopted the idea of pruning but instead of removing infrequent words, words that are common in at least any two documents are removed to maintain a language model containing words that are unique across the language models used by AutoCor. If a common word is found in the target languages, there is no way of identifying to what specific target language the word belongs. Thus, removing common words to all the set of input documents will see to it that the words that are left are words that are unique to each of the set of documents, which are used to model our languages, and will be used in the automatic query generation module.

Documents considered as input are HTML documents. Words such as “about”, “us”, “contact”, and “home” are one of the most common words that appear in most language specific HTML documents. These are called general or standard navigation hyperlinks[6]. Thus, words used as labels to general navigation hyperlinks are also pruned if they appear in any two or more sets of documents.

5. Query Generation

Odds-ratio (OR) selects the *k* terms with highest odds-ratio scores. The odds-ratio score for a word w is defined as:

$$\log_2\left(\frac{\mathrm{P(w\,|\,relevant\,doc)*(1\text{-}P(w\,|\,non\ relevant\,doc))}}{\mathrm{P(w\,|\,non\ relevant\,doc)*(1\text{-}P(w\,|\,relevant\,doc))}}\right)$$

where: P (w | relevant doc) – Probability of a word from a relevant document
P (w | non-relevant doc) – Probability of a word from a non-relevant document

Odds-ratio (OR) achieves very good results compared to other methods such as uniform, term frequency, and RTFIDF.

5.1. Text Categorization on Language Classification

Text categorization is a basic task in document processing. It allows automated handling of enormous streams of documents in electronic form. N-gram based approach is a technique that can be used in text categorization. It is tolerant of textual errors and works very well for language classification and is able to achieve up to 99.8% correct classification[7].

[6] Yu, S., D. Cai, J. Wen and W. Ma. 2003. Improving Pseudo-Relevance Feedback in Web Information Retrieval Using Web Page Segmentation. *Proceedings of the twelfth international conference on World Wide Web*. pp. 11 – 18.

[7] Ghani, R., R. Jones, D. Mladenic. 2001. Using the Web to Create Minority Language Corpora. *10th International Conference on Information and Knowledge Management*. pp. 279 - 286.

An N-gram is an n-character slice of a longer string. A string is sliced into sets of overlapping n-grams. Before the string is sliced, blanks are appended at the beginning and end of the string. The following provides examples of bi-grams, tri-grams and quad-grams.

N-gram-based text categorization is based on calculating and comparing profiles of n-gram frequencies (see Figure 2). It first computes for profiles on training set data that represents the various categories or various languages. A new document with an unknown category is processed by the profile generator. The process of computing the profile for the document to be classified is the same as how profiles are created for each of the training sets. Finally, the distance measure, known as the out-of-place measure, between the documents profile and each of the category profiles are computed and the category whose profile has the smallest distance to the document's profile is the selected category of the new document with unknown category[8].

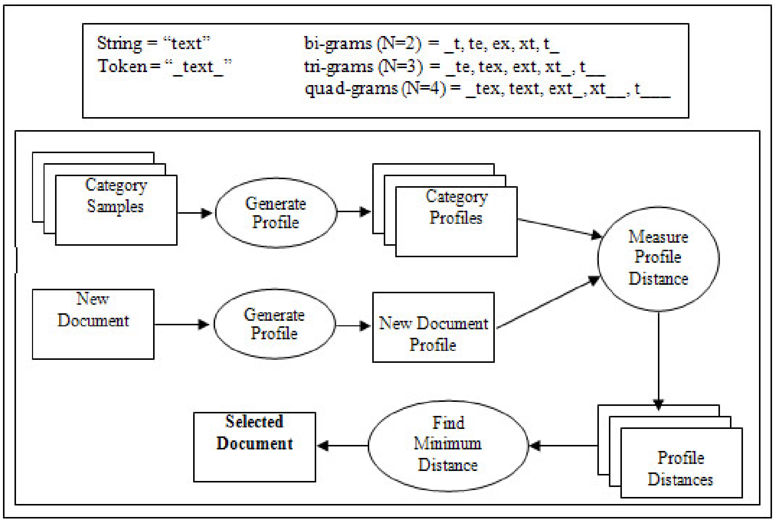

Figure 2: Common Words of Set A, B, C and D.

5.2. Out-of-place Measure Between Two Profiles

The out-of-place measure determines how far out of place an N-gram in one profile is from its place in the other profile. Figure 4 illustrates how the calculation is done using a few sample N-grams. For each N-gram in the document profile, counterparts are matched in the category profile and out-of-place distance is computed. The N-gram "ING" is at rank 2 in the document, but at rank 5 in the category. Thus it is 3 ranks out of place. If an N-gram (such as "ED" in Figure 3) is not in the category profile, it takes some maximum out-of-place value. The sum of all of the out-of-place values for all N-grams is the distance measure for the document from the category[9].

[8] Cavnar, W. B. and J. M. Trenkle. 1994. N-gram-based Text Categorization. *Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval.* Las Vegas: NV. pp. 161-175.

[9] Cavnar, W. B. and J. M. Trenkle. 1994. N-gram-based Text Categorization. *Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval.* Las Vegas: NV. pp. 161-175.

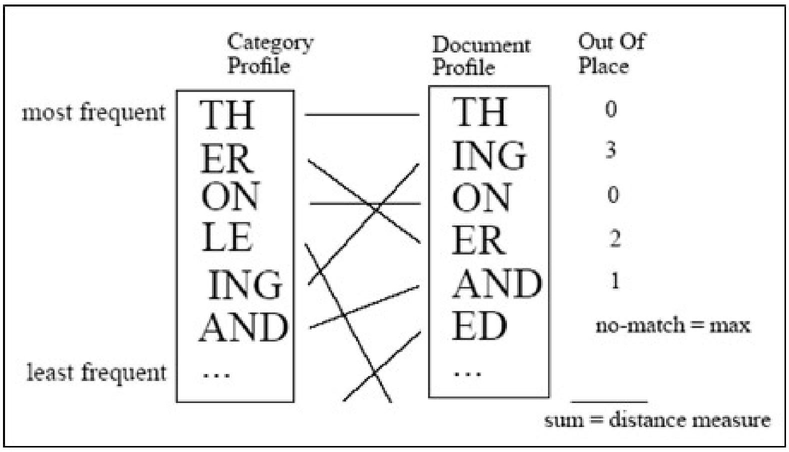

Figure 3: Out-of-place Computation[9].

6. Results and Discussions

The goal in evaluating an Information Retrieval (IR) system is to measure its effectiveness, that is, the ability of the system to retrieve relevant documents. Specifically, precision and recall are used to measure the effectiveness of an IR system[10].

Given a set of documents D and a query Q, A is the set of documents retrieved by the system and R is the set of all relevant documents in D. A ∩ R is the set of documents relevant to query Q (see Figure 4).

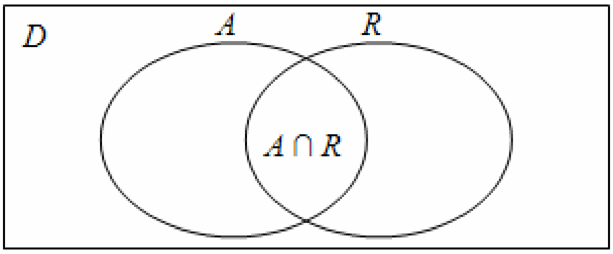

Figure 4: A Diagrammatic View of a Document Collection[10].

The precision of the system is the proportion of retrieved material that is actually relevant. It is the proportion of the items retrieved that are relevant[10]. Precision can be computed by using the following formula:

$$\begin{aligned} \textit{PRECISION} &= \textit{relevant retrieved / total retrieved} \\ &= |A \cap R| / |A| \end{aligned}$$

Recall is the proportion of relevant material actually retrieved in answer to a search request. It is the proportion of relevant items that are retrieved[10]. Recall can be computed by using the following formula:

$$\begin{aligned} \textit{Recall} &= \textit{relevant retrieved / total no. of relevant documents} \\ &= |A \cap R| / |R| \end{aligned}$$

[10] Jizba, R. 2004. Measuring Search Effectiveness. [online]. Available: http://www.hsl.creighton.edu/hsl/Searching/Recall-Precision.html. July 15, 2004.

To evaluate performance and efficiency level over a set of N test queries, precision level is averaged at each recall level r. It is the summation of the precision computed per query (level r) wherein the total number of test queries is N:

$$P(r) = \frac{\sum_{i=1}^{N} P(r)_i}{N} \quad \text{Equation 1}$$

If 100% recall is achieved at $i = k$ where in $k < N$ then to compute the average precision, we have:

$$P(r) = \frac{\sum_{i=1}^{k} P(r)_i}{k} \quad \text{Equation 2}$$

The documents were pre-tagged with the language on which the documents were written. The corpus that was used in this research contains documents from the web. The corpus contains 4,000 documents which consist of 250 documents tagged as Bicolano, 250 documents tagged as Cebuano, another 250 documents which are tagged as Tagalog and the rest of the documents were tagged with 3 different languages namely English, Hungarian and Polish. The documents in Bicolano, Cebuano and Tagalog are the target or relevant documents, while the non-relevant documents are documents in English, Hungarian and Polish. The selection of the set of non-relevant documents was based on similar character sets and the availability of documents.

Each of the target languages was tested for query lengths 1 to 5, with 100 generated queries per query length, both with and without pruning. Precision and recall was computed per query, and average precision was computed per query length.

AutoCor on a closed corpus achieved higher average precision with common word pruning for all query lengths 1 to 5, across all the target languages. The highest improvements per language range from 18% to 53% and 19% to 26% for domain and non-domain specific data sets, respectively (DS and NDS); and highest precision values per language range from 21% to 61% and 37% to 51% for DS and NDS data sets, respectively. The results showed that common word pruning improved the precision of the system (Bicolano: with 52.96% highest improvement at query length 4, Cebuano: with 18.00% highest improvement at query length 1, Tagalog: with 19.78% highest improvement at query length 2).

On the other hand, AutoCor on an open corpus yielded the following results: the highest precision values per language range from 14% to 72% and 9% to 61% for DS and NDS, respectively.

These results indicated that the DS data sets yielded better results since the search is more topic-specific and directs the search more effectively. Secondly, the consistent trends of the results show that increasing the query length does not necessarily increase the precision of the system. Thirdly, the test results on the web reveal that using the web as a resource may provide extreme lowest and highest precision values, due to the vast amount of information on the web and their variability.

The test shows that with common word pruning, AutoCor achieves a higher precision than without pruning regardless of query length for all the target languages (Bicolano, Cebuano, Tagalog) that were used during the test. Common word pruning would maintain the language model of each of the target languages to be unique. The results show that with common word pruning, fewer documents in closely-related languages where retrieved since most common words had already been removed in the language models of the target languages. Therefore, the terms that were selected by the query generator for the relevant set are most likely unique to each of the target languages.

Focus on the accuracy of the classifier using common word pruning was made in this study. Although time efficiency in the document classification was not measured in the evaluation of the algorithm, it could be inferred from the minimization of the search space that time efficiency could have been improved by the introduction of common word pruning.

References

Fortunato, F. T. 1993. *Mga Pangunahing Etnoling-guistikong Grupo sa Pilipinas*. Malate, Manila, Philippines: De La Salle University Press.

Gao, J. and K. Lee. 2000. Distribution-based pruning of backoff language models. *The 38th Annual Meeting of the Association for Computational Linguistics*. Hong Kong. 579 – 588.

Ghani, R., R. Jones and D. Mladenic. 2001. Using the Web to Create Minority Language Corpora. *10th International Conference on Information and Knowledge Management*. 279 – 286.

Jones, R. and R. Ghani. 2000. Automatically Building a Corpus for a Minority Language on the Web. *In the Proceedings of the Annual Meeting of the Association of Computational Linguistics 2000*, pp. 29-36.

Orasan, C. and R. Krishnamurthy. 2000. An Open Architecture for the Construction and Administration of Corpora. *In Proceedings of the Second International Conference on Language Resources and Evaluation*. pp. 793 – 800.

Cavnar, W. B. and J. M. Trenkle. 1994. N-gram-based Text Categorization. *Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval*. Las Vegas: NV. pp. 161-175.

Jizba,R.2000.Measuring Search Effectiveness. [online].Available: http://www.hsl.creighton.edu/hsl/Searching/Recall-Precision.html. July 15, 2004.

Yu, S., D. Cai, J. Wen, W. Ma. 2003. Improving Pseudo-Relevance Feedback in Web Information Retrieval Using Web Page Segmentation. *Proceedings of the twelfth international conference on World Wide Web.* pp. 11 – 18.