Automatic Web Search Query Generation to Create Minority Language Corpora

Rayid Ghani
Carnegie Mellon University
Rayid.Ghani@cs.cmu.edu

Rosie Jones
Carnegie Mellon University
Rosie.Jones@cs.cmu.edu

Dunja Mladenic
J. Stefan Inst., Slovenia and
Carnegie Mellon Univ., USA
Dunja.Mladenic@ijs.si

ABSTRACT

The Web is a valuable source of language specific resources but collecting, organizing and utilizing this information is difficult. We describe CorpusBuilder, an approach for automatically generating Web-search queries to collect documents in a minority language. It differs from pseudo-relevance feedback in that retrieved documents are labeled by an automatic language classifier as relevant or irrelevant and a subset of documents is used to generate new queries. We experiment with various query-generation methods and query-lengths to find inclusion/exclusion terms that are helpful for finding documents in the target language and find that using *odds-ratio* scores calculated over the documents acquired so far was one of the most consistently accurate query-generation methods. We also describe experiments using a handful of words elicited from a user instead of initial documents and show that the methods perform similarly. Applying the same approach to multiple languages show that our system generalizes to a variety of languages.

1. INTRODUCTION

In this paper, we describe techniques which only require the user to give a handful of keywords or documents for automatically collecting language specific resources from the Web and present a system which automatically generates Web search queries to construct corpora for minority languages. Our proposed approach requires no human intervention once the system is provided with a freely available language filter and initial documents in the target language. It is orders of magnitude more efficient than fitting a crawler with a language filter.

We explore different term-selection methods and lengths for generating queries and modify the queries based on feedback from the language filter. We show that our methods can learn to generate queries that can acquire a reasonable number of documents in Slovenian from the Web and that our approach also generalizes to other languages that are also minority languages on the Web.

SIGIR'01, September 9-12, 2001, New Orleans, Louisiana, USA..

2. APPROACH

Our system, CorpusBuilder, iteratively creates new queries, in order to build a collection of documents in a single language. The target language is defined by one or more initial documents provided by the user, and the language filter.

At a high level, CorpusBuilder works by taking as initial input from the user two sets of documents, relevant and non-relevant. Given these documents, it uses a term selection method to select words from the relevant and non-relevant documents to be used as inclusion and exclusion terms for the query, respectively. This query is sent to the search engine and the highest ranking document is retrieved, passed through the language filter and added to the set of relevant or non-relevant documents according to the classification by the filter. The process is then iterated, updating the set of documents that the words are selected from at each step.

The general algorithm is as follows:

1. Initialization
2. Generate query inclusion terms from relevant documents and query exclusion terms from non-relevant documents
3. Retrieve next most relevant document for the query
4. Language Filter, assign document to relevant or non-relevant set
5. Update frequencies and scores based on relevant and non-relevant documents
6. Return to step 2

We generate queries using conjunction and negation of terms. Query term selection methods were as follows: *uniform* (UN) selects k terms from the relevant documents, with equal probability of each term being selected. *term-frequency* (TF) selects the k most frequent terms from the relevant documents. *probabilistic term-frequency* (PTF) selects k words from the relevant documents according to their frequency. More frequent words are more likely to be selected. *rtfidf* (RTFIDF) selects the the top k words ranked according to their rtfidf scores. *odds-ratio* (OR) selects the k terms with highest odds-ratio scores. *probabilistic odds-ratio* (PO) selecting words with probability proportional to their odds-ratio scores.

The simplest measure used as a baseline is random selection of terms, i.e. a uniform probability distribution is imposed over all words in the vocabulary (UN). Scoring terms according to their frequency (TF) has been used as a simple measure known to give good results for feature scoring in document categorization [4]. Using a multinomial distribution over frequency of terms (PTF - *probabilistic-term-frequency*) has been shown to perform better than simple

Query Len	Query-generation Methods ordered by Performance	
	Percentage Target Class after retrieving 3000 documents	Target Documents per Query after issuing 1000 queries
1	68.8%(OR)>46%(PO)>39%(UN)>19.1%(PTF)> 8.9%(TF)	1.6(OR)>0.43(PO)>0.36(UN)>0.25(PTF)>0.09(TF)
3	82.3%(OR)>65.8%(UN)*>64.1%(PTF)>33.2%(TF)>9.4%(PO)*	1.77(OR)>0.38(PTF)>0.18(TF)>0.094(PO)>0.035(UN)
5	92.4%(UN)*>92.3%(PO)*>81.5%(OR) >77.4%(PTF)>77.0%(TF)	1.2(OR)>0.53(TF)>0.14(PTF)>0.01(PO)>0.01(UN)
10	100%(PO)*>88.7%(PTF) >79.2%(OR)>50%(UN)*>7.0%(TF)	0.02(OR)>0.01(PTF)>0(TF)>0(PO)>0.001(UN)

Table 1: *odds-ratio* always finds more target documents per query, and retrieves a large proportion of documents in the target class, regardless of the query length. Results marked with * did not find 3000 documents, so percentage is shown for the maximum number of documents retrieved.

frequency on a similar problem [1]. Haines and Croft [3] show that rtfidf is a good scoring mechanism for information retrieval. Mladenić and Grobelnik [4] have shown that scoring using *odds-ratio* (OR) achieves very good results on document categorization when dealing with a minority concept, which is exactly our problem scenario since Slovenian is a minority language on the Web. Motivated by the superior performance of *probabilistic term-frequency* over *term-frequency*, we derived a variant of *odds-ratio*, which is selecting terms according to a multinomial distribution over odds-ratio scores of terms (PO - *probabilistic-odds-ratio*)

We used van Noord's TextCat implementation [5] of Cavnar and Trenkle's character n-gram based algorithm [6] which was shown to be over 90% accurate on a variety of languages and document lengths.

3. EXPERIMENTAL RESULTS

The goal of our system is to collect as many documents in the target language as possible while minimizing the cost. The evaluation measures we used were (a) percentage of documents retrieved in the target class and (b) the number of documents in the target class per unique web query.

We systematically compared the term selection methods for each length independently. The summary results are shown in Table 1. When fixing the number of queries issued, *odds-ratio* (OR) finds the greatest number of target documents. In terms of the number of documents examined, *odds-ratio* is again the best (length 1 — 3) except when all methods have about the same performance (length ≥ 4).

We experimented with initializing our system with different initial documents and different sets of 10 keywords supplied by native Slovenians and didn't observe significant differences in performance. We also conducted experiments with Croatian, Czech and Tagalog and as in the case of Slovenian, *odds-ratio* found more target documents than both *term-frequency* and *probabilistic term-frequency* for the same number of total documents examined (see Table 2).

Method	Language	Target Docs at 1000 Total Docs
TF-3	Slovenian	178
PTF-3	Slovenian	646
OR-3	Slovenian	**835**
TF-3	Croatian	39
PTF-3	Croatian	410
OR-3	Croatian	**677**
TF-3	Czech	385
PTF-3	Czech	451
OR-3	Czech	**743**
TF-3	Tagalog	440
PTF-3	Tagalog	359
OR-3	Tagalog	**664**

Table 2: *odds-ratio* found more target documents than both *term-frequency* and *probabilistic term-frequency* for the same number of total documents examined, for Slovenian, Croatian, Czech and Tagalog.

4. DISCUSSION

We presented an approach for automatically collecting web-pages in a minority language and showed that it performs well on several natural languages (Slovenian, Croatian, Czech and Tagalog) and only requires a handful of documents to start with. More details about our experiments and findings can be found in [2].

We found that our basic *odds-ratio* query construction method outperforms our other methods and also performs better than simply using the existing Related Pages function in AltaVista. *odds-ratio* picks inclusion query terms that are highly unique to the target language while excluding terms that are unique to non-relevant languages. Since this is the only method which uses both relevant and non-relevant documents simultaneously to select query terms, we believe that this property is the key to its success. *tf* picks terms that are frequent in the target language but not necessarily unique and hence results in queries that are not as precise as those generated by *odds-ratio*.

An interesting question is how our results transfer to other target concepts such as collecting documents on a topic or finding documents that match a user profile. We have conducted some preliminary experiments for collecting Course web pages starting from a handful of examples and we plan to pursue collecting topic-specific corpora in future work.

5. REFERENCES

[1] Ghani R. and Jones, R., Learning a Monolingual Language Model from a Multilingual Text Database, CIKM 2000.

[2] Ghani,R., Jones, R. and Mladenić, D., Building Minority Language Corpora by Learning to Generate Web Search Queries, Carnegie Mellon University, Technical Report CMU-CALD-01-100, 2001. `http://www.cs.cmu.edu/~TextLearning/corpusbuilder/`

[3] Haines, D. and Croft, B., Relevance Feedback and Inference Networks, Proc. of SIGIR 1993.

[4] Mladenic, D. and Grobelnik, M., Feature selection for unbalanced class distribution and Naive Bayes, ICML 1999.

[5] van Noord, G., TextCat, http://odur.let.rug.nl//~vannoord/TextCat/

[6] Cavnar, W. and Trenkle, J., N-Gram-Based Text Categorization, Proc. of SDAIR 1994.