Bayero Journal of Pure and Applied Sciences, 1(1):62 – 66
Received: October, 2008
Accepted: November, 2008

PROBLEMS OF INDEXING CLASSES OF NEWS BASED ON THE COMPUTED IMPORTANCE OF WORDS

Echezona, S.C.
Department of Computer Science, University of Nigeria, Nsukka.
E-mail: Echez2003@yahoo.com

ABSTRACT
News is classified. Such classes as sports, politics, news on crime, gossips, business, etc., are common amongst newspapers in Nigeria. Interestingly most readers and patrons of newspapers adopt the rule of the thumb in choosing a suitable newspaper to read/buy. However, most newspapers try to cover all classes but end up in strikingly stress areas. This paper firstly explains the basic steps in generation of Document Indexes then secondly, highlights some of the problems associated with using full automation in classifications. Some of the problems identified are loss of relevance, loss of coverage and partial automation.

Keywords: Classification, Indexing, keywords

INTRODUCTION

Automatic keywords generation has been used in many areas of information indexing and classification, such as, articles, abstracts, captions and books. Every class of news goes with certain keywords that are unique for a given class. Such words as "Domination, National Assembly, Speaker, House of Assembly, Senate, etcetera", suggest politics even when some can suggest other areas. The method adopted by this paper is to extract relevant titles that belong to various classes of news heuristically and processed them down to keywords (word stems) that will represent these classes using software. Thereafter a user can use any article/caption from any newspaper to match these word stems using the same method that will be discussed.

MATERIALS AND METHODS

To generate word stems for every class of news, newspaper captions/titles/headlines of Nigerian origin are used. Such articles/captions/headlines are heuristically identified, extracted and separated into documents then entered into software which will process them in the following sequence (Doyll and Blankenship; 2002):

- Tokenization
- Noise/common/stop words removal
- Reduction to word stems by removal of suffices
- Weighting factor attached to the word stems
- Keywords extraction by choosing suitable threshold

Developing a user-friendly software to perform this task within seconds will definitely make things easier and more interesting for the end users. People from all walks of life read newspapers everyday; business men looking for business leads, politicians, researchers, students seeking information, etc. Many of these people have to read a large number of newspapers, page after page, perusing tons of seemingly useless information and actually missing the essential due to fatigue and boredom.

Being able to skim through scores of newspapers, reading only headlines and letting the software decide if the story could answer ones questions would exponentially improve the productivity of most newspaper readers. Relevant articles could be sorted out and later read in detail after all the sorting and categorization has been done.

The original ideas of Luhn on which most of automatic text analysis has been built goes on to describe a concrete way of generating document representatives through weighing or classifying keywords are discussed. Luhn's earlier paper (Edmundson and Wyllus, 2007) states: "It is here proposed that the frequency of word occurrence in an article furnishes a useful measurement of word significance. It is further proposed that the relative position within a sentence of words having given value of significance furnishes a useful measurement for determining the significance of sentences. The significant factor of a sentence will therefore be based on a combination of these two measurements." His assumption is that frequency data can be used to extract words and sentences to represent a document.

Let f be the frequency of occurrence of various word types in a given position of text and r their rank order, that is the order of their frequency of occurrence, then a plot relating f and r yields a curve similar to hyperbolic curve below. This is a curve demonstrating Zipf's Law (as contained in Yu and Salton 2006) which states that the product of the frequency of use of words and the rank order is approximately constant. Zipf (Yu and Salton 2006) verified his law on American newspaper English.

Luhn used it as a null hypothesis to enable him specify two cut-offs, an upper and a lower, thus excluding non-significant words, The words exceeding the upper cut-off were considered to be common and those below the lower cut-off rare and therefore not contributing significantly to the content of the article. He thus devised a counting technique for finding significant words, by which he meant the ability of words to discriminate content, reached a peak at a rank order position half way between the two cut-offs and from the peak fell off in either direction reducing to almost zero at the cut-off points.

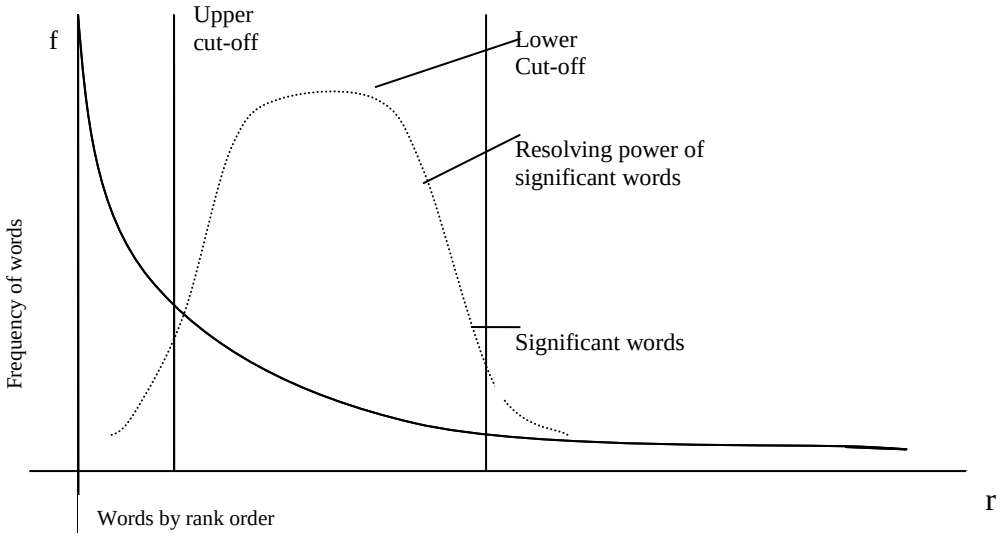

Fig 1. Zipf's curve, Yu and Salton; 2006.

Typical News Classes of Nigerian Newspapers are as follows:

- Sports
- Politics
- Religion
- Entertainment
- Information Technology
- Health
- Crime
- Environment
- Business
- Education
- Government
- Etc.

Some important algorithms developed for this study are as follows: (Echezona's 2000)

1. Algorithm for Tokenization (close to Pascal language).
   Two recognizable functions are "Expunging from the text special characters" and "from left to right cut the remaining text to strings with blank as delimiter". List of special characters are punctuation symbols, such as: ',' , ';' , ':' , '?' , '.' , '"" , '/' , '(' , ')' , '_', etc.

```
procedure expunge;
var  writt:string; ch:char;
while not eoln(text) do
begin
        read(text,ch);
        if not (ch in skipsett)
        then  begin
        write(chara, ch)
        end;
end;

procedure word;
var t,v:string;


begin
```

```
while not eoln(comp) do
        begin
        read(comp,t);
        if (t<>' ') then
        v:=Copy(t,1,1);
        strg:=concat(strg,v)
        end
end;
```

2. Removal of some common words. Some common words encountered are tabulated below: (Chang, 2007)

BOW	IF	REAL	INTO
STAND	THAT	CONSPICUOUS	A
RESPITE	THEN	THE	TAKE
RESPECTIVE	SPITE	THEIR	OFF
INSPITE	THUS	NUMBER	OF
BETWEEN	ROLL	MAKE	FOR
AT	IN	IT	REGRET
TO	RELAX	TICK	WHO
PULL	PASS	AS	DOWN
MADE	RAISE	RETURN	HIS
REQUIRE	WITH	WITHIN	HARD
WITHOUT	RUSH	OUT	HELP
LIFT	ESTABLISH	ALL	PART
DUE	DAY	DEPLORE	NIGHT
BRIEF	ANY	BEHIND	BUT
BECOME	AMONG	ALONE	AMONGST
CAN	ACROSS	ANYTHING	ANYWHERE
AFTERWARDS	BELOW	BEEN	ALREADY
TOUGH	GOES	SOFT	BEFORE
THOUGH	HOW	VIEW	BESIDE
THOROUGH	CALL	FAIL	AGAIN
ALTHOUGH	UP	AROUND	IS
GO	HER	BOTH	ETC.

An algorithm for noise words extractions is as follows:

```
Procedure noisewords(text, arr:noisewd; n:integer);
var wd: string;
begin
        while not eof() do
                begin
                readln(text wd);
                for j:=1 to n do
                        begin
                        if strcomp(wd, arr[j]) then
                        writeln(text,"");
                        else
                        write{text,wd};
                        end
                end;
end;
```

3. Suffix removal algorithm uses common suffices available to compare with the end of each recognized term and chops off the part that match. Simple algorithm is given below. Other checks like morphological transformations existing in English language which may alter the stem of suffixed words; for example the word "absorb" is transformed into "absorption" when the suffix "tion" is added. Similarly "hop" is transformed to "hopping", "relief" becomes "relieving" and so on. Transformational rules can be set up (outside the algorithm below) in order to recode various automatic generated stems following suffix removal. A typical rule might state "remove one of the possible occurrences of b, d, g, l, m, n, p, r, s, t, from the end of the generated term". These rule's algorithm is not included.

```
Procedure remsuf;
var t,l:integer;
```

```
        s,c,strr,suf:string;
    begin
        reset(cf,'retainer'); \* New file of word stem. Reset positions the pointer to the first
    record*\
        rewrite(cff,'wordfile');  \* Tokenized words requiring suffix removal*\
        reset(suff,'suffix.text');   \* File of suffixes  *\
        while not eof(cf) do
            begin
            readln(cf,strr);
            while not eof(suff) do
                begin
                read(suff,suf);  l:=0; t:=length(suf);
                for j:=1 to length(suf) do
                    begin
                    s:=copy(suf,t-j+1,1);
                    c:=copy(strr,length(strr)-j+1,1);
                    if(s=c) then  begin l;=l+1 end
                    end;
                if(l = t) then begin chara:=copy(strr,1,length(chara)-t);
                writeln(cff,chara) end
                end;
                rewrite(cf,'retainer');
                while not(eof(cf) and eof(cff)) do
                    begin
                    readln(cff,chara);
                    writeln(cf,chara);
                    end;
            end;
            close(suff);
            close(cff);
            close(cf);
    end;
```

Some suffices in English are tabulated below: (Chang, 200]

S	ISM	AL	LY	LLY
IVENESS	IVE	NESS	D	MENT
OUS	CEOUS	ACEOUS	IES	ALIC
ER	UOUS	ABILITIES	ACIDEOUS	AIC
ABILITY	ACIDEOUSLY	AICAL	ACIES	ABLE
AICALLY	ACEOUSNESS	ABLED	AICISM	AICS
ACTIES	ABLENESS	ACITIES	AICISMS	AL
ALISATION	ABLINFUL	ABLER	ABLING	AE
ALISATIONALLY	ALISATIONAL	ABLY	AGER	ACY
ACEUOSLY	ALISEDLY	ACITY	ACISE	AGE
ACEUOSNESSES	AGINGFUL	ALISER	ALISED	AGES
AGED	AGING			

4. Finally, weighting by calculating frequencies within and outside documents and applying selected weighting formula, the final indexes results which can now represent content of the body of classes. Typical example includes: To find each term's weight using Inverse Document Frequency Weight – INVDFWT.

   $WEIGHT_k = FREQ_{ik} * (\ln(n) - \ln(DOCFREQ_k) + 1)$

   Where $FREQ_{ik}$ is the frequency of the term k in document i, n is the number of documents, and, $DOCFREQ_k$ is the number of documents the unique term appeared.

Associated Problems

This work was done at two periods covering a decade; the following problems were seen to be associated with the results, (that is, the index terms generated). These problems are:

- Loss of relevance
- Loss of coverage
- Partial automation

Loss of Relevance

This refers to the inability of index terms to still be relevant over time. The researcher has successfully carried out this indexing twice within a period of a decade, and has observed that most of the index terms generated tends to lose relevance with time.

This might be because no area is static. New syllables continue to be generated while others are dropped. This is most prominent when names of persons are used as part of the content identifiers. It is obvious that new entrants are made into the field of discuss, and obsolete ones are seldom referred to. For instance, if the name like "Patricia Etteh" that has been foremost in most newsprints in the recent past is used to represent part of the corpus of Nigerian politics will lose relevance in say next decade.

Loss of Coverage

Most classes of news command vast syllables. And so, to cover or index using abstracts/captions/titles/headlines extracted from say newspapers may not finitely explore all possibilities. This may bias the result of the search and hence lead to loss of coverage. An attempt to cover all may mean expanding the coverage of the input data. Meaning; sampling more newspapers for over almost endless periods.

Partial Automation

Some steps of computation of index terms are painfully manual, not completely automatic. The classification of the headlines which forms the input source must at best be done by an expert in the field – say expert from Mass Communication Department – This is manual. There is no way computer can be made to recognize a term as belonging to Politics or Crime class of news, unless instructed to do so.

Another step that is manual is the making the choice of a collection of suffix striped terms that will be assigned a representative word so that thereafter the representative words will be used as part of index. These steps make the automation partial.

Conclusion

The processes of automating the indexing of news emphasis of Nigerian newspapers have been on for a long time with some hiccups. A way out of the first two problems stated above, is not to use the body of the articles; rather, the period under study has to be expanded as much as possible to accommodate more inputs. The loss of relevance may be remedied by regular updates of the index terms. This will have the overhead of maintenance cost. This is because, as a result of loss of coverage remedy; large data would be expected each time it is run.

REFERENCES

Chang,C.(2007): Finding Prototypes for Nearest Neighbour Classifiers *IEEE Trans on Computers.*

Doyl, L. B. and Blackenship, D.A.(2002): "Technical Advances in Automatic Classification" in Progress in Information Science and Technology: *Proceedings of the American Documentation Institute Annual Meeting*, (10): 3 – 7, 1966, Santa Monica, Addriaenne Press.

Echezona, S.C. (2000): "Comparison of News Emphasis (of Nigerian Newspapers) Based on the Frequency of Occurrence of Importance Words." Project Submitted for the Degree of Bachelor of Science in Computer Science, University of Nigeria (unpublished) .

Edmundson, H.P., Wyllus, R.E. (2007): Automatic Abstracting and Indexing – Survey and Recommendations *Communications of the ACM*, Vol. 4, No. 5 (5): 226 – 234.

Igwe, E.L. Ayoghu, C.S., IloanyaA, I.S. and Mmerole, L.C. (2007): "News Emphasis of Nigerian Newspapers." Project Submitted for the Degree of Bachelor of Science in Computer Science, University of Nigeria, Nsukka, (Unpublished) .

Yu, C. T. Salton, G. (2006): Precision Weighting– An Effective Automatic Indexing Method. *Journal of ACM*. Vol. 23, No. 1 (1):76 – 88.