Stop Word Lists in Document Retrieval Using Latent Semantic Indexing: an Evaluation

A N K Zaman
School of Computer Science
University of Guelph
Guelph, ON, Canada
azaman@uoguelph.ca

ABSTRACT: *Removing stop words is very useful for many text processing applications e.g. text/document retrieval, cross language translation, text categorization, text summarization etc. In this world, different language has different stop word lists, and those are useful for text processing applications. Literature claims that the use of such lists improves retrieval performance. The goal of this research is to evaluate the effect of using English stop word lists in Latent Semantic Indexing (LSI)-based information retrieval (IR) systems with large text dataset. Here, three different lists are compared: two were compiled by IR groups at the University of Glasgow, and the University of Tennessee, and the third one is our own list developed at the University of Northern British Columbia. We also examined the case where stop words were not removed from the input dataset. Our research finds that using tailored stop word lists improves retrieval performance. On the other hand, using arbitrary (non-tailored) lists or not using any list reduces the retrieval performance of LSI-based IR systems with large text dataset.*

Keywords: Document retrieval, Latent Semantic Indexing, Stop word list, Recall-precision

Received: 1 October 2011, Revised 6 November 2011, Accepted 12 November 2011

1. Introduction

While working with text data, some particular words are very common in every document, and those have very little influence to distinguish documents from each other; these words are considered as stop words. In 1990, Deerwester et al., [1] removed a set of 439 common (stop) words from MED and CISI datasets. From our knowledge, this is the first stop word list used to work with latent semantic indexing (LSI). Example of stop words include articles (e.g., a, an, the), prepositions (e.g., at, by, in, to, from, with), conjunctions (e.g., and, but, as, because) etc.

2. Related Works

Proper stop word identification and removal are a central problem for many text processing applications in different domains. Stop words of a given text dataset might not be stop words of other datasets [2]. Stop words have significant impact on the text retrieval processes in different languages. In [3], Dolamic et al., evaluated two stop word lists with lengths 571 and 9 respectively on the Cross-Language Evaluation Forum (CLEF) dataset. According to the authors, the Divergence from Randomness (DFR) model shows lower retrieval performance when a short or no stop word list is removed from the input dataset. However, in case

of a revised Okapi (information retrieval system) implementation, retrieval performance does not show any significant difference whether a short, long or no stop word list is removed from the dataset. The authors also draw the same conclusion for other natural languages such as French, Hindi, and Persian. The main weaknesses of their work are that they use arbitrary stop word lists and do not use the *tf-idf* (term frequencyinverse document frequency) weighting scheme [4]. Zou et al., [5] show that the removal of stop words from Chinese text is important for Chinese word segmentation and improves the performance of Chinese text retrieval. The removal of stop words also improves systems' performance in Arabic IR and Arabic text summarization [6] [7]. The removal of stop words has positive impact on English text categorization [8]. Stop word removal also improves retrieval performance in case of cross-language IR. A number of crosslanguage based IR systems are reported in literature, e.g., Bengali-Hindi, Turkish-English, Japanese-English [9] [10] [11]. Schuemie et al., [12] removed stop words for cross-language IR for biomedical literature. The removal of stop words plays an important role in different text processing domains in different languages.

Although the Text REtrieval Conference (TREC) encourages text retrieval research, it does not provide any evidence or rule to use stop words in IR research. Different IR research groups use different stop word lists, and the size of these lists vary. As there is no standard stop word list for English text, one open question is the following: What are the effects of tailored (based on a certain dataset) vs. arbitrary (not tailored) stop word lists on LSI-based text retrieval systems with large datasets? In this study, we used TREC-8 LA Times dataset for our experiments. Two existing, arbitrary stop word lists, as well as our own tailored stop word list are considered (see table 2). The retrieval results are compared by removing three different sets of stop words from the input dataset (one at a time) and also without removing any stop words. The rest of the paper is organized as follows. Section 3 gives an overview of the LSI technique. Section 4 presents some characteristics of the input dataset and describes the way our stop word list is compiled. The experimental setup and retrieval results are presented in Section 5. Concluding remarks are in Section 6. Appendix A presents our developed stop word list (also called UNBC stop word list).

3. Latent Semantic Indexing

In 1988, Dumais et al. [13] introduced the idea of LSI to process textual data, and to handle synonymy, and polysemy. After that, LSI is used for many applications including document retrieval.

3.1 Overview

LSI is a method that exploits the idea of vector space model and singular value decomposition (SVD). SVD is an effective dimensional reduction scheme. It has been proved to be a very good choice for uncovering latent semantic structure [1]. SVD can be applied with an arbitrary rectangle matrix with the entries on the rows and columns. The matrix is then decomposed into three matrices containing singular vectors and/or singular values. These three matrices with special forms show a breakdown of the original matrix into linearly independent components or factors. Many of these components are very small, leading to an approximate model that contains many fewer dimensions. Thus, for IR purposes, SVD provides a reduced model for representing the *term-to-term*, *document-to-document* and *term-to-document* relationships. By dimension reduction, it is possible for documents with somewhat different profiles of term usage to be mapped into the same vector of factor values. This property helps to eliminate the noise in the original data, thus improving the reliability of the algorithm. Suppose we obtained a *td* term-by-document matrix *M* from the collection indexing process of the traditional vector space method. We can apply SVD on *M*, which is then decomposed into three special matrices *U*, *S* and *V*. The decomposition can be written as:

$$M = USV^T \qquad (1)$$

U is the $t * t$ orthogonal matrix ($UU^T = I_t$) having the left singular vectors of M as its columns, and V is the $d * d$ orthogonal matrix ($VV^T = I_d$) having the right singular vectors as its columns, and S is the $t * d$ diagonal matrix having the singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(t,d)}$ of M in order along its diagonal. It should be noted that for any arbitrary matrix, such a factorization exists [14].

Generally, in (1), the matrices *U*, *S* and *V* must all be of full rank. However, SVD offers a simple strategy for optimal approximation to fit using smaller matrices [1]. If the singular values in *S* are ordered by size, the first *k* largest values may be kept and the remaining smaller ones set to zero. The product of the resulting matrices is a matrix M_k which is only approximately equal to *M*, and is of rank *k*. Since zeros were introduced into *S*, the representation can be simplified by deleting the zero rows and columns of *S* to obtain a new diagonal matrix S_k, and then deleting the corresponding columns of *U* and *V* to obtain U_k and V_k respectively. The rank-*k* model with the best possible least-squares-fit to *M* can be written as follows:

$$M_k = U_k S_k V_k^T \quad (2)$$

Where, M_k is a matrix of size $t * d$, U_k is of size $t * k$, S_k is of size $k * k$, and V_k is of size $k* d$.

SVD provides an optimal solution to dimensionality reduction in that it derives an orthonormal space, where the dimensions are ordered. Therefore, projecting the set of documents onto the k lowest dimensions is guaranteed to have, among all possible projections to a k dimensional space, the lowest possible least-square distance to the original documents.

3.2 Weighting Schemes

One significant issue in LSI-based IR systems is the term weighting, i.e., assigning weight to a term so that the assigned weight properly reflects the contribution of the term in distinguishing the considered document from other documents. Let, L_{ij} be the local weight of the term i in the document j and tf_{ij} be the frequency with which the term i appears in the document j. The local weight in terms of raw term frequency is defined as follows:

$$\text{Raw term frequency: } L_{ij} = tf_{ij} \quad (3a)$$

Let G_i be the global weight of the term i, let tf_i be the frequency of the term i in the entire collection, let df_i be the frequency of documents in which i occurs, and let d be the number of documents in the whole collection.
The following equations define the *idf* and *tf-idf* weighting schemes:

$$idf : = G_i = log\ (\frac{d}{df_i}) \quad (3b)$$

$$tf - idf : G_i = tf_{ij} \times idf_i \quad (3c)$$

For our experiments we used *tf-idf* weighting scheme to create term-document matrix.

4. Input Dataset and Stop Word Lists

The input data/text collection includes the articles published by the Los Angeles Times in the two year period from Jan 1, 1989 - December 31, 1990. Each file contains the articles from one day (e.g., a file with the name "LA123190" contains articles published on 31 Dec 1990). Every such file contains a number of documents (e.g., the LA123190 contains 134 different documents). Table 1 presents the important characteristics of the TREC-8 LA Times dataset.

Number of documents	131,321
Size of the input dataset	476MB
Average vocabulary size (approximately)	500
Average document size (approximately)	40 KB
Largest file size	828 KB (LA052089_0101)
Smallest size	352 Bytes (LA070189_000)
Number of words in the smallest file	91
Number of words in the largest file	167,045
Number of relevant files (out of 131,312 files) with respect to TREC-8 query set	1,151

Table 1. Characteristics of the TREC-8 LA Times Dataset (1989, 1990)

Manning et al., [15] describe a way in their book to prepare a list of stop words: "*the general strategy for determining a stop list is to sort the terms by collection frequency (the total number of times each term appears in the document collection), and then to take the most frequent terms, often hand-filtered for their semantic content relative to the domain of the documents being*

indexed, as a stop list, the members of which are then discarded during indexing." Our own stop word list has been compiled by following the above idea. It includes the University of Glasgow (319 words) and the University of Tennessee (439 words) stop word lists, 730 TREC file names (input dataset), 22 tag names (e.g., doc, docno, etc) and other words (e.g., alphanumeric words, roman numbers). Its total length is 1891 unique words. The algorithmic steps to create this stop word list are given below:

- Consider the terms whose frequency is at least 2 (a term must be present in the document at least twice).
- Create an initial stop word list by combining the stop word lists of the IR University of Tennessee and
- University of Glasgow groups (without duplication of terms in the list).
- Remove all the punctuation from the input TREC-8 LA Times dataset.
- Create a list of terms from the input dataset, in descending order of term frequencies, i.e., the term with the highest term frequency will be at the top of the list.
- Manually extract the special items to be added to the initial list (those terms are not already in the initial list) to create an extended stop word list.
- Add all file names to the initial list as every file contains file names, e.g., LA123190.
- Add all tag names, e.g., doc, docno, to the initial stop word list.
- Add roman numbers to the initial list, e.g., xvii.
- Add scale units, e.g., ft, mm, etc.
- Add adjectives and adverbs, e.g., ago.
- Add prefixes from words, e.g., non (as in non-governmental).
- Add special words, e.g., haven (as in haven't), doesn (as in doesn't).
- Add dates, e.g., l9th.
- Add foreign words as dataset in newspaper articles. Add suspicious words, e.g., aaftink, aachen, ora.
- Add other words, e.g., ext (telephone extension), 19th, z90, v6 (engine).

To compile this stop word list, first we searched the high frequency, low frequency, and then other special terms out of 132,785 terms in the frequency table. We repeat this in a number of cycles by removing different stop words from the TREC-8 LA Times dataset. Searching stop words is very time consuming as the dataset as well as the number of terms are large. The most difficult thing is to choose a word as a stop word. Since the TREC-8 LA Times dataset contains newspaper articles (on politics, sports, geography, history, science-technology, etc.), there are variations in the contents. Some characteristics of this dataset are presented in the table 1.

5. Experiments

This study finds out the effects of "*stop words / common words*" on a LSI-based text document retrieval process for the TREC-8 LA Times dataset. Evidence is developed to indicate the most effective stop word lists for LSI-based ad hoc IR processes for the mentioned dataset in table 1. We performed our experiment by removing the stop word lists mentioned in table 2, as well as without removing them. We applied Porter's stemming [16] to find the root words from the input text. 50 TREC-8 queries were used (associated with the mentioned dataset) to evaluate the retrieval performance. Our findings are presented in terms of recallprecision graph [14].

Table 3 shows the 10-point interpolated precision of the four different retrieval systems: the UNBC system which uses our stop word list developed at the University of Northern British Columbia; the UTen system which uses the University of Tennessee stop word list; the UGla system which uses the University of Glasgow stop word list; and the STEM system which does not use any stop word list. The recall-precision graph based on the results in Table 3 is shown in Figure 1.

5.1 Recall-Precision Graph

In Table 3, the recall value of 0.1 represents the top 10% of the retrieved documents (in the collection) which are relevant to a query set. As an example, using the UNBC system, the precision associated with the top 10% of the documents is 0.1757 (i.e.,

Dataset	Stop Word Lists	Stemming	Weighting Scheme	Number of Queries
TREC-8 LA Times	University of Glasgow	Porter's Stemming	tf - idf	50
	University of Tennessee			
	University of Northern British Columbia			
	No list			

Table 2. Parameters for the study of stop words

10-Point Recall	UNBC	UTen	UGla	STEM
0.1	0.1757	0.1513	0.1221	0.1385
0.2	0.1108	0.0994	0.0864	0.1058
0.3	0.0888	0.0735	0.0799	0.0841
0.4	0.0724	0.0693	0.0722	0.0743
0.5	0.0678	0.0672	0.0694	0.0710
0.6	0.0659	0.0647	0.0671	0.0662
0.7	0.0646	0.0609	0.0660	0.0632
0.8	0.0621	0.0588	0.0628	0.0592
0.9	0.0554	0.0489	0.0578	0.0553
1.0	0.0436	0.0316	0.0374	0.0247

Table 3. 10 – point Interpolated Precision of the Four Systems

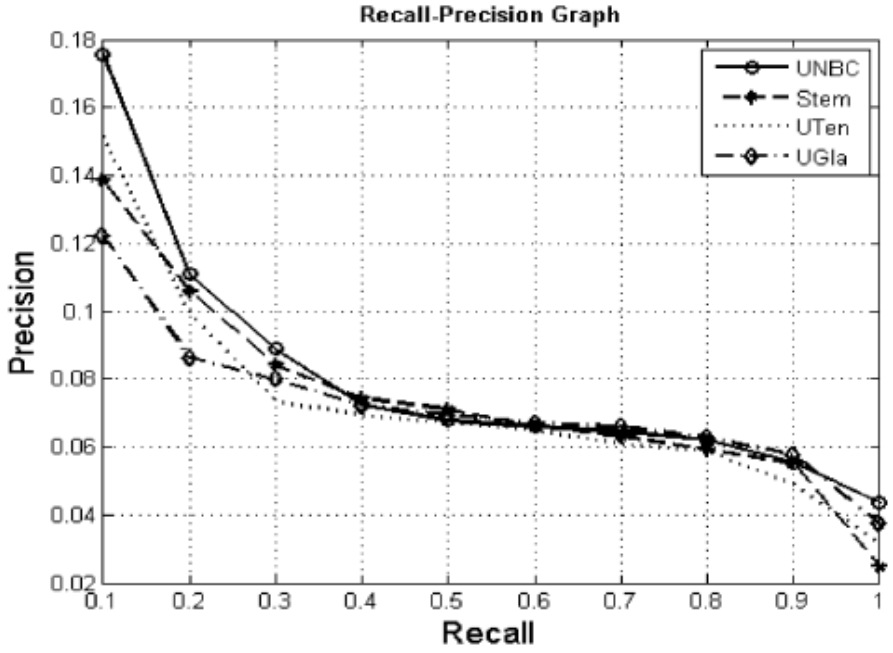

Figure 1. Recall-Precision Graph

17.57%). This value is calculated by interpolating the precision values of all 50 queries used for this research at the standard recall value 0.1.

The retrieval systems are compared in terms of precision in different standard recall points, e.g., 0.1, 0.2. For example, at recall point 0.3 (top ranked 30% documents), the precision values for UNBC is 8.88%, and it is 7.35% for UT. So, UNBC shows 1.53% (8.88%-7.35%) better retrieval performance than UT for the top 30% retrieved documents. If we look at the recall point 0.3 in figure 1, we can see the differences.

In the end, the system UNBC with extended stop word list provides the best result when compared to the three other systems. For the top 10% retrieval, it shows 5.37% better retrieval performance than UG, 3.68% better retrieval performance than STEM, and 2.44% better retrieval performance than UT. However, after top 40% retrieval all the systems show almost the same retrieval performance. Note that in STEM, we just applied Porter's stemming without removing stop words, and the retrieval performance is 1.64% better than UG's. From the above results, it is clear that the use of an arbitrary set of stop words reduces retrieval performance in case of LSI-based ah hoc IR with large dataset. Dai and Sun [17] showed in their paper that topic specific stop word removal from news stories helps to identify accurate event. This idea also proves our idea that tailored stop word list removal from specific text dataset improves document retrieval. Removal of proper stop word lists not only benefitted LSI based text processing but also text processing with other methods e.g. kK-Nearest neighbour(k-NN) [18], tagged Huffman compressed text searching [19] etc.

6. Conclusion

To identify and use of proper stop word lists from a specific dataset is very important in doing research with text. In our research, we investigated the performance of LSI by using three different stop word lists, and also, without using any stop word list, i.e., without removing stop words from the input dataset. Our main finding is that for a LSI-based ad hoc IR system, the use of an arbitrary stop word list reduces retrieval performance; for better retrieval performance, a tailored stop word list must be assembled for every unique large dataset.

Appendix A

a	a, a1, a10, a11, a12, a13, a14, a15, a16, a18, a19, a2, a20, a21, a22, a23, a24, a25, a26, a27, a28, a29, a3, a30, a300, a301, a31, a310, a32, a320, a320s, a321, a33, a330, a330s, a340, a340s, a35, a36, a4, a41, a43, a5, a6, a7, a8, a9, aa, aaa, aaaa, aaaah, aable, aachen, aad, aaf, aaftink, aagla, aah, aahed, aahing, aahs, aai, aar, aas, ab, aba, abb, abc, abl, about, above, abt, ac4, aca, acc, accordingly, ace, ach, acn, acoc, across, ad, ada, adl, adm, adn, ae, afc, afi, afl, after, afterwards, ag, again, against, agn, ago, ah, ahl, ahm, ak, al, ala, alf, all, allows, alm, almost, alone, along, alp, already, also, although, always, am, ama, amc, ami, amo, among, amongst, amp, an, ana, anc, and, ang, anh, ani, ann, another, any, anyawi, anybody, anyhow, anyone, anything, anyway, anywhere, ap, apart, appear, appropriate, ar, are, around, as, asa, ash, aside, aso, associated, ast, asu, at, att, av, ava, available, aw, away, awfully
b	b, b1, b10, b100, b12, b2, b210, b3, b4, b5, b52, b6, b7, b747, b8, b9, ba, ba2, ba3, ba4, ba6, baa, baa1, baa2, baa3, back, bbdo, be, became, because, become, becomes, becoming, been, before, beforehand, behind, being, below, ben, ber, beside, besides, best, better, between, beyond, blm, bmg, bnd, bo, boa, bol, boo, both, bp, brief, bs, bsd, bta, btu, bu, but, bv, by, bye, byline, byu
c	c, c1, c10, c11, c12, c13, c14, c15, c16, c17, c18, c19, c2, c20, c21, c22, c23, c24, c2h2, c3, c4, c5, c6, c7, c759915, c8, c9, cac, came, can, cannot, cant, cause, causes, cb, cbn, cbo, cc, ccaa, ccdc, cch, cct, ce, cee, cellrule, certain, cfa, ch, cha, changes, chi, chj, chp, chr, chu, cio, ck, clo, cmc, cmdr, cmv, co, column, come, consequently, contain, containing, contains, coq, correction, corresponding, cot, could, cpa, cpl, cra, crc, currently, cvj, cwl, cy, cya
d	d, d1, d10, d11, d12, d13, d14, d15, d16, d2, d3, d4, d5, d6, d7, d8, d9, date, dateline, day, db, de, dea, ded, def, described, dh, di, dib, did, didn, different, dl, dmc, do, doc, doc, doc, docid, docno, docno, docno, does, doesn, doing, don, done, doo, dosen, down, downwards, dr, ds, du, during, dy

e	e, e1, e5, each, eb, ebb, ec, ed, edit, ee, eel, eg, eh, ei, eight, eip, either, ek, el, elk, else, elsewhere, em, en, enough, eo, ep, epa, epp, er, es, et, etc, ev, even, ever, every, everybody, everyone, everything, everywhere, ew, ex, example, except, ext
f	f, f1, faa, far, fcc, fe, few, fg, fi, fifth, fig, first, five, fl, flo, followed, following, foo, for, former, formerly, forth, four, from, ft, fu, further, furthermore
g	g, g3, g4, g5, g6, ga, gc, ge, get, gets, given, gives, gn, go, gone, goo, good, got, gq, graphic, graphics, great, grp, gte, gto, gtp, gu
h	h, h1, h11, h13, h18, h2, h20, h2a, h2o, h3, h4, h5, h6, h7, ha, haa, had, hadn, hardly, has, have, haven, having, he, headline, hem, hence, her, here, hereafter, hereby, herein, hereupon, hers, herself, hi, him, himself, his, hither, hj, hmmm, hmmmmm, ho, hoo, how, howbeit, however, hoy, ht, hu, hy
i	i, ibf, iby, ic, ida, ie, if, ignored, ii, iii, ik, il, im, ima, immediate, in, ina, inasmuch, inc, indeed, indicate, ing, inner, insofar, instead, into, inward, ip, iq, ir, iri, irk, is, isl, isn, it, its, itself, iv, ivo, ix
j	j, j1, j10, j2, j3, j4, j5, j6, j8, ja, jc, ji, jo, jr, just
k	k, k1, k1n, k2, k2r, k9, ka, ka7, kan, kao, kee, keep, kept, kg2, kg7, kg8, kh1, kh2, kh7, kh8, know, knx, ko, ky
l	l, la, la2, la23, la89, la90, last, latter, latterly, le, least, length, less, lest, let, li, life, like, little, ll, lo, long, loo, lot, lou, lp, ls, lt, ltd, lx, ly
m	m, m1, m16, m2, m3, m4, m5, m6, m62, m71, m78, ma, maa, made, mae, make, mal, man, many, may, mc, mca, mcb, md, mdc, me, mea, meanwhile, mee, men, mfg, mg, mi, might, mk, mm, mme, mo, moc, moo, more, moreover, most, mostly, mott, mr, ms, msg, mt, mu, much, must, mvp, mx, my, myself
n	n, n1, n2, n5, n6, na, na3, na6, name, namely, ncols, ncr, ne, near, necessary, neither, nev, never, nevertheless, new, next, nfc, nfl, ng, ni, nine, nl, nlrb, nmb, no, nobody, nom, non, none, noone, nor, normally, not, nothing, nov, novel, now, nowhere, nr, nu
o	o, oat, oc, occ, och, od, oda, odd, of, off, oft, often, oh, oj, ol, old, on, once, one, ones, only, onto, ooh, ooo, oooo, ooooo, ooz, op, opt, or, ora, ord, ot, other, others, otherwise, ou, ought, our, ours, ourselves, out, outside, over, overall, ow, own
p	p, p1, p2, pa, page, part, particular, particularly, pb, pba, pc, pcb, pcc, pcl, pcp, pe, people, per, perhaps, pfc, pg, ph, pic, pj, placed, plc, please, plo, plus, possible, pp, pr, prc, pre, pro, probably, provides, pt, pta, ptl
q	q, q106, qb, qd2, qd3, qd4, qd7, qd8, qe2, qe4, qe5, qe6, qe7, qe8, qed, qf2, qf5, qf7, qg4, qg5, qg6, qh4, qi, qtr, que, quite
r	r, r1, r2, r2d2, r3, ra, ra3, ra6, rather, rb, rbi, rc, rda, re, really, relatively, respectively, right, rk, ro, roo, rowrule, rv
s	s, s1, s1630, s1w, s2, s358, s4, s605, said, same, sb, sba, sbk, sc, sca, scc, scca, sce, scr, sdg, sdy, se, second, secondly, section, see, seem, seemed, seeming, seems, self, selves, sensible, sent, serious, seven, several, shall, she, shoo, should, si, since, six, smc, smu, so, some, somebody, somehow, someone, something, sometime, sometimes, somewhat, somewhere, soo, sou, specified, specify, specifying, spn, sq, sr, ss, ssi, ssy, st, state, still, stu, su, sub, subject, such, sup, sw, sy, syd
t	t, t4, ta, table, tablecel, tablecell, tablerow, take, taken, tcu, td, te, text, than, that, the, their, theirs, them, themselves, then, thence, there, thereafter, thereby, therefore, therein, thereupon, these, they, thi, third, this, thorough, thoroughly, those, though, three, through, throughout,

	thru, thus, thy, time, tmc, tnt, to, together, too, tot, toward, towards, ts, tu, twa, twice, two, type
u	u, u2, u60, ua, uc, ul, ulf, un, unc, und, under, unless, uno, until, unto, up, upa, upon, us, use, used, useful, uses, usf, using, usually, ut, uw
v	v, v20, v2500, v6, v8, v8v, v9l, v9w, va, value, various, ve, very, vh, vi, via, vii, viii, vill, vin, viz, vs, vt, vu, vw, vy
w	w, w6, w8, wa, was, wasn, way, wbc, wcb, wdm, we, wee, well, went, were, what, whatever, whatnot, when, whence, whenever, where, whereafter, whereas, whereby, wherein, whereupon, wherever, whether, which, while, whither, who, whoever, whole, whom, whose, why, wig, will, with, within, without, wk, wok, word, words, work, world, would, wouldn, wow, wr, wt, wu, wuz, wwii
x	x, x2, x6, xi, xiii, xiv, xix, xr, xt, xtra, xv, xvi, xxi, xxiii, xxiv, xxv, xxvii, xxx, xyz
y	y, y95, yap, ye, year, years, yet, yo, yoo, you, your, yours, yourself, yourselves, yve
z	z, z28, z90, zac, zero, zx, zz, zzzz,
File Names	la010189, la010190, la010289, la010290, la010389, la010390, la010489, la010490, la010589, la010590, la010689, la010690, la010789, la010790, la010889, la010890, la010989, la010990, la011089, la011090, la011189, la011190, la011289, la011290, la011389, la011390, la011489, la011490, la011589, la011590, la011689, la011690, la011789, la011790, la011889, la011890, la011989, la011990, la012089, la012090, la012189, la012190, la012289, la012290, la012389, la012390, la012489, la012490, la012589, la012590, la012689, la012690, la012789, la012790, la012889, la012890, la012989, la012990, la013089, la013090, la013189, la013190, la020189, la020190, la020289, la020290, la020389, la020390, la020489, la020490, la020589, la020590, la020689, la020690, la020789, la020790, la020889, la020890, la020989, la020990, la021089, la021090, la021189, la021190, la021289, la021290, la021389, la021390, la021489, la021490, la021589, la021590, la021689, la021690, la021789, la021790, la021889, la021890, la021989, la021990, la022089, la022090, la022189, la022190, la022289, la022290, la022389, la022390, la022489, la022490, la022589, la022590, la022689, la022690, la022789, la022790, la022889, la022890, la030189, la030190, la030289, la030290, la030389, la030390, la030489, la030490, la030589, la030590, la030689, la030690, la030789, la030790, la030889, la030890, la030989, la030990, la031089, la031090, la031189, la031190, la031289, la031290, la031389, la031390, la031489, la031490, la031589, la031590, la031689, la031690, la031789, la031790, la031889, la031890, la031989, la031990, la032089, la032090, la032189, la032190, la032289, la032290, la032389, la032390, la032489, la032490, la032589, la032590, la032689, la032690, la032789, la032790, la032889, la032890, la032989, la032990, la033089, la033090, la033189, la033190, la040189, la040190, la040289, la040290, la040389, la040390, la040489, la040490, la040589, la040590, la040689, la040690, la040789, la040790, la040889, la040890, la040989, la040990, la041089, la041090, la041189, la041190, la041289, la041290, la041389, la041390, la041489, la041490, la041589, la041590, la041689, la041690, la041789, la041790, la041889, la041890, la041989, la041990, la042089, la042090, la042189, la042190, la042289, la042290, la042389, la042390, la042489, la042490, la042589, la042590, la042689, la042690, la042789, la042790, la042889, la042890, la042989, la042990, la043089, la043090, la050189, la050190, la050289, la050290, la050389, la050390, la050489, la050490, la050589, la050590, la050689, la050690, la050789, la050790, la050889, la050890, la050989, la050990, la051089, la051090, la051189, la051190, la051289, la051290, la051389, la051390, la051489, la051490, la051589, la051590, la051689, la051690, la051789, la051790, la051889, la051890, la051989, la051990, la052089, la052090, la052189, la052190, la052289, la052290, la052389, la052390, la052489, la052490, la052589, la052590, la052689, la052690, la052789, la052790, la052889, la052890, la052989, la052990, la053089, la053090, la053189, la053190, la060189, la060190, la060289, la060290, la060389, la060390, la060489, la060490, la060589, la060590, la060689, la060690, la060789, la060790, la060889, la060890, la060989, la060990, la061089, la061090, la061189, la061190,

	la061289, la061290, la061389, la061390, la061489, la061490, la061589, la061590, la061689, la061690, la061789, la061790, la061889, la061890, la061989, la061990, la062089, la062090, la062189, la062190, la062289, la062290, la062389, la062390, la062489, la062490, la062589, la062590, la062689, la062690, la062789, la062790, la062889, la062890, la062989, la062990, la063089, la063090, la070189, la070190, la070289, la070290, la070389, la070390, la070489, la070490, la070589, la070590, la070689, la070690, la070789, la070790, la070889, la070890, la070989, la070990, la071089, la071090, la071189, la071190, la071289, la071290, la071389, la071390, la071489, la071490, la071589, la071590, la071689, la071690, la071789, la071790, la071889, la071890, la071989, la071990, la072089, la072090, la072189, la072190, la072289, la072290, la072389, la072390, la072489, la072490, la072589, la072590, la072689, la072690, la072789, la072790, la072889, la072890, la072989, la072990, la073089, la073090, la073189, la073190, la080189, la080190, la080289, la080290, la080389, la080390, la080489, la080490, la080589, la080590, la080689, la080690, la080789, la080790, la080889, la080890, la080989, la080990, la081089, la081090, la081189, la081190, la081289, la081290, la081389, la081390, la081489, la081490, la081589, la081590, la081689, la081690, la081789, la081790, la081889, la081890, la081989, la081990, la082089, la082090, la082189, la082190, la082289, la082290, la082389, la082390, la082489, la082490, la082589, la082590, la082689, la082690, la082789, la082790, la082889, la082890, la082989, la082990, la083089, la083090, la083189, la083190, la090189, la090190, la090289, la090290, la090389, la090390, la090489, la090490, la090589, la090590, la090689, la090690, la090789, la090790, la090889, la090890, la090989, la090990, la091089, la091090, la091189, la091190, la091289, la091290, la091389, la091390, la091489, la091490, la091589, la091590, la091689, la091690, la091789, la091790, la091889, la091890, la091989, la091990, la092089, la092090, la092189, la092190, la092289, la092290, la092389, la092390, la092489, la092490, la092589, la092590, la092689, la092690, la092789, la092790, la092889, la092890, la092989, la092990, la093089, la093090, la100189, la100190, la100289, la100290, la100389, la100390, la100489, la100490, la100589, la100590, la100689, la100690, la100789, la100790, la100889, la100890, la100989, la100990, la101089, la101090, la101189, la101190, la101289, la101290, la101389, la101390, la101489, la101490, la101589, la101590, la101689, la101690, la101789, la101790, la101889, la101890, la101989, la101990, la102089, la102090, la102189, la102190, la102289, la102290, la102389, la102390, la102489, la102490, la102589, la102590, la102689, la102690, la102789, la102790, la102889, la102890, la102989, la102990, la103089, la103090, la103189, la103190, la110189, la110190, la110289, la110290, la110389, la110390, la110489, la110490, la110589, la110590, la110689, la110690, la110789, la110790, la110889, la110890, la110989, la110990, la111089, la111090, la111189, la111190, la111289, la111290, la111389, la111390, la111489, la111490, la111589, la111590, la111689, la111690, la111789, la111790, la111889, la111890, la111989, la111990, la112089, la112090, la112189, la112190, la112289, la112290, la112389, la112390, la112489, la112490, la112589, la112590, la112689, la112690, la112789, la112790, la112889, la112890, la112989, la112990, la113089, la113090, la120189, la120190, la120289, la120290, la120389, la120390, la120489, la120490, la120589, la120590, la120689, la120690, la120789, la120790, la120889, la120890, la120989, la120990, la121089, la121090, la121189, la121190, la121289, la121290, la121389, la121390, la121489, la121490, la121589, la121590, la121689, la121690, la121789, la121790, la121889, la121890, la121989, la121990, la122089, la122090, la122189, la122190, la122289, la122290, la122389, la122390, la122489, la122490, la122589, la122590, la122689, la122690, la122789, la122790, la122889, la122890, la122989, la122990, la123089, la123090, la123189, la123190
Dates	l9th

References

[1] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science,* 41 (6) 391- 407.

[2] Dragut, E. C., Fang, F., Sistla, A. P., Yu, C. T., Meng, W. (2009). Stop word and related problems in web interface integration. The Proceedings of the Very Large Database Endowment, PVLDB 2 (1), p. 349–360.

[3] Dolamic, L., Savoy, J. (2010). When stopword lists make the difference. Journal of the American Society for Information Science and Technology, 61: p. 200–203.

[4] Zaman, A. N. K., Brown, C. G. (2010). Latent semantic indexing and large dataset: Study of termweighting schemes. Fifth International Conference on Digital Information Management (ICDIM), p.1- 4.

[5] Zou, F., Wang, F. L., Deng, X., Han, S. (2006). Evaluation of Stop Word List in Chinese Language. International Conference on Language Resources and Evaluation (LREC), Genoa, Italy.

[6] El-Khiar., I. (2006). Effects of Stops Words Elimination for Arabic Information Retrieval: A Comparative Study. *In*: International *Journal of Computing and Information Sciences*, 4 (3).

[7] Azmi, A., Al-thanyyan, S. (2009). Ikhtasir - A user selected compression ratio Arabic text summarization system. International Conference on Natural Language Processing and Knowledge Engineering, p.1-7.

[8] Feng Xia, Tian Jicun, Liu Zhihui. (2009). A Text Categorization Method Based on Local Document Frequency. Sixth International Conference on Fuzzy Systems and Knowledge Discovery, p.468-471.

[9] Mandal, D., Gupta, M., Dandapat, S., Banerjee, P., Sarkar, S. (2008). Bengali and Hindi to English CLIR evaluation. Advances in Multilingual and Multimodal Information Retrieval (8th Workshop of the Cross-Language Evaluation Forum). Budapest, Hungary Springer Verlag, p. 95–102.

[10] Celebi, E., Sen, B., Gunel, B. (2009). Turkish - English cross language information retrieval using LSI. 24th International Symposium on Computer and Information Sciences, p.634-638.

[11] Li and Shawe-Taylor, Li, Y., Shawe-Taylor, Y. (2006). Using KCCA for Japanese–English crosslanguage information retrieval and document classification. *Journal of Intelligent Information System* 27 (2) 117–133.

[12] Dumais, S. T., Furnas, G. W., Landauer, T. K., Deerwester, S. (1988). Using latent semantic analysis to improve information retrieval. Conference on Human Factors in Computing, New York: ACM, p.281-285.

[13] Schuemie M., Trieschnigg D., Kraaij, W. (2007). Cross Language Information Retrieval for Biomedical Literature. Sixteenth Text REtrieval Conference, Gaithersburg, MD, USA.

[14] Schutze, H., Silverstein, C. (1997). Projections for efficient document clustering. *In*: ACM/SIGIR(SIGIR Forum 31), p. 74–81.

[15] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze. (2008). An Introduction to Information Retrieval. Cambridge University Press Cambridge.

[16] Porter, M. (2002). An algorithm for suffix stripping, avaliable :http://tartarus.org/~martin/PorterStemmer/ Access time.

[17] Dai, Xiangying, Sun, Yunlian, (2010). Event identification within news topics. *International Conference on Intelligent Computing and Integrated Systems*, p.498-502.

[18] de Oliveira Gomes, N., Passos, E. P. L. (2011). Text categorization study case: Patents' application documents. Sixth IEEE Conference on Industrial Electronics and Applications, p.446-450.

[19] Edleno Silva de Moura, Gonzalo Navarro, Nivio Ziviani, Ricardo Baeza-Yates. (2000). Fast and flexible word searching on compressed text, *ACM Trans. Inf. Syst*. 18, p.113-139.