Using WordNet in Document Clustering of a Consumer Health Web Collection

Patrick Herron
INLS 110 KDD
Spring 2005

Abstract

The present study evaluates a method of deriving topics from a large collection of consumer health web pages using unsupervised learning techniques. Can a set of a useable number (5-9) of exhaustive topics be generated via document clustering, via simple k-means, of this collection, specifically using a simplified WordNet representation of those documents?

Within this general document clustering task, a number of smaller subtasks were required. 40 different feature reduction schemes, 10 generated from each of four different general types (including ones based upon the SPECIALIST lexicon and WordNet) were evaluated in terms of whether they prevented the overfitting of data, via a set of 120 learning experiments, for k=5, 7 and 9.

The use of a binarized word vector helped avoid monster clusters, yet term frequency-based word vectors invariably led to poor cluster performance. Further, using random projection of attributes in order to reduce features helped reduce the likelihood of overfitting even further.

Clusters generated at k=5 and 7 for three of the original 40 representations were selected for further evaluation, namely to see if either of two different types of mean term vectors might elicit latent topic labels for those clusters. The three representations selected were all reduced by binarization and random projection at 25% of their original feature sets.

On the basis of the most frequent terms within clusters, the WordNet-based (wtlrposwnsyn) clusters for both k=5 and k=7 appeared to have no readily-apparent latent topics at all, while the wtlragrstemmed-based and the wtlrposstemmed-based representations seemed vaguely suggestive of labels based on an analysis of term frequency-related data for the clusters. A closer examination of clusters for wtlragrstemmed-based and the wtlrposstemmed-based representations at k=5 using a novel metric named TCFICF (essentially a cluster-centric, rather than document-centric, version of TFIDF) revealed that the largest cluster in every scheme was quite possibly based on some font-specific web-code noise that escaped preprocessing and screening of preprocessing data. The other clusters did not suffer the same problem, but did suggest a heavy overlapping of common themes within documents that did at least weakly suggest some topical separation, not compelling enough to suggest clear topic labels.

Finally, a manual qualitative inspection of a random sample of 5 pages from each cluster showed results that appeared to be more promising than was evident from the term frequency data. The most coherent clusters came from the WordNet-related representation according to the small qualitative analysis.

Non-mutually exclusive or hierarchical clustering algorithms might be better-suited for document clustering in the present domain in order to compensate for the overall topical "overlap," as many of the pages in the collection reflect multiple purposes. However, the overlap may also be due to a truly random effect – the random selection of attributes in random projection, and so the problems of the system are nothing but problems of randomness itself.

Regardless of whether some problems are wholly random, the preprocessing step should be refined to eliminate all html-level and css-level data from the files. Adaptation of WordNet to clustering purposes should be carried out in conjunction with POS-tagging in the preprocessing step along with some discriminatory employment of hypernymy relations if the use of WordNet is to be optimized. The author's analysis of clusters by term frequency data should be refined so that the data coincides with and augments the qualitative assessment of the web pages. Further development of the system should be continued in coordination with a greater effort to gather qualitative data for a broader range of the candidate representations and results.

1. Introduction

The aim of the present project is to measure the performance of clustering when employing WordNet to provide a concept-level semantic feature representation of web pages from a large heterogeneous medical portal. The goal of the clustering task is to identify latent subject headings with the ultimate goal of improving website usability.

The NC Health Info (NCHI) website (http://nchealthinfo.org), a web collection maintained by UNC-Chapel Hill and the National Library of Medicine, is a portal for approximately 3600 web sites independently maintained by North Carolina-based health care programs, providers, and services. NCHI pages are currently organized by a set of topics which is represented on the front page of the site by a drop-down menu. The browsable and selectable drop-down topic menu on the main page contains 493 different subjects that provide a navigation aid to the content of the site.

While there is no question the present 493 subject headings, in representing the NCHI collection, have some value to a user of the website, it is a daunting task for a user to actually browse that long of a list in order to identify what topic it is the user wishes to select for further exploration. A more user-friendly version of the site would have, in place of a 493-subject heading-long dropdown menu, a different menu with five to seven subject headings. Determining what those five

to seven representative subject/menu headings would be is a daunting task—one that is often done in the course of constructing a website, and done manually by information architects. For a very large website such a task may be performed more efficiently using a data-driven model. It is possible that in clustering the documents (represented at a concept level) using k-means for k={5, 7, 9} we may facilitate the "uncovering" of good subject headings. Documents from the collection will be clustered using k-means where k=5, 7 and 9 (because those numbers are the most "useable" in terms of the number of menu options), and then each of those clusters will be evaluated to find its most frequent meaningful concepts and terms. It is hoped that those frequent meaningful terms may then be coherent enough either to proxy as a representative set of terms for their clusters or, preferably, allow for the selection of a single representative term for that cluster. The set of representative terms for all clusters will then, if apparent from their analysis, serve as that user-friendly menu of subject headings for navigating the site in lieu of the ungainly 493-subject-long menu. If no such terms are readily available from the clustering, it is hoped that the frequent term lists may be indicative of the problems with the current approach.

To sum, providing a shorter list of higher-level subject headings may make the user's browsing task a more efficient one. While it is well-established that a short set of subject headings for a site navigation is more useable than a very long one, data-driven means (particularly clustering of documents with concept level

representations) for determining terms for such a menu are not so well-established.

A "naive" WordNet representation ("wtlrposwnsyn[1]") will be evaluated alongside three other simpler representations to see if WordNet used in this naïve fashion provides better clusters. The other three simpler representations include a bag-of-words, stemmed and stopped ("webtermtwostemmed"), a bag-of-words joined on the SPECIALIST lexicon (henceforth "wtlragrstemmed"[2]), and the necessary intermediary between the wtlragrstemmed and wtlrposwnsyn, the wtlragr set reduced to the only four parts of speech in WordNet: N, V, ADJ, and ADV ("wtlragrposstemmed").

By "naïve" it is meant that WordNet synset identifiers are merely used to represent terms, yet no efforts are made to use many of the features of the WordNet network such as meronymy or hypernymy. Terms from a previous representation are joined on terms in a WordNet synset table, and then the appropriate synset identifiers are used to represent those terms. While some synset identifiers represent multiple terms (synonymy), some terms have multiple identifiers (ambiguity). Ultimately the question being answered is,

[1] 'wt' for web term, 'lr' for lragr, 'pos' for part of speech reduction, and 'wnsyn' for WordNet synset.
[2] 'lragr', because the specialist db table is titled 'lragr.'

then, will the benefits of synonymy inherent within WordNet outweigh the costs of ambiguity?

Evaluation of the system will come in several phases. First, minimum and maximum term frequencies for representations will be evaluated using distinct term counts and the levels will be selected as a result. 40 different feature reduction approaches (binarization and random projection) will be applied to the core four feature representations, leading to a total set of 40 feature representations, and then all 40 resulting representations will be run through the automated clustering task. The first cluster-based evaluation will be designed to eliminate representations that tend towards overfitting, and to select a tractable number of feature representations for further evaluation and for returning focus towards examination of any possible advantage to the naïve use of WordNet and the potential for the system to generate a small set of useable & easily labeled clusters. The second cluster evaluation step will involve inspecting lists of the top ten most frequent terms for two feature selections, wtlragrstemmed binarized at random projection=25% and wtlrposwnsyn binarized at random projection=25%. Clusters from both k=5 and k=7 for these two representations will be inspected using these term frequency-based lists. The third cluster evaluation will compare the clusters of the aforementioned two representations with wtlragrposstemmed binarized at random projection=25% for k=5 only using not only term frequency but also a novel measure I have named TCFICF, a

measure that refits the for-document TFIDF measure to the purpose of examining adjusted term frequencies in different clusters. Finally, a random sample of 5 documents from each of the five clusters for each of the three representations (a total of 5*5*3=75) documents will be qualitatively examined to see if the clustering performed in the present study has any immediate promise.

1.1 Target Research Questions

- Does a naïve employment of WordNet improve topic clustering? Specifically, in the present implementation, will the benefits of synonymy inherent within WordNet outweigh the costs of ambiguity? How much of a factor is the intermediate step of reducing the term set by part of speech on performance for clustering?

- Can clustering, in particular clustering in the complete absence of any manually derived topic data, even for the purposes of evaluation (purely unsupervised) be used to devise a useable (n=5 to 9) topic menu for the NC Health Info consumer health web portal, and possibly other web portal collections like it?

- What feature reduction approaches lead to better clusters? Specifically, what feature reductions help us avoid monster clusters, that dreaded product of overfitting?

- Can a novel measure provide more information about clusters than term frequency?

2. Background

2.1 Usability, the number seven, cluster sizes, and the *Duh Factor*

Most well-organized heterogeneous web collections typically are organized into four to ten general topics. Those topics are typically listed horizontally at the top of the page and are provided to the user of the web sites as a navigation aid for navigating the contents of the web collections contained therein. From my own experience not only as a web user but also as an information architect over the last decade or so, less than four terms seems never to provide enough information about the contents of the site, while having more than ten topics makes the organization of the site, at least on a high level, less immediately graspable. It seems that there is something to this notion that more than ten items are less immediate in their general graspability. George Miller, the cognitive scientist responsible for WordNet, himself has documented some basis for this observation, noting that, "there is a clear and definite limit to the accuracy with which we can identify absolutely the magnitude of a unidimensional stimulus variable. I would propose to call this limit the span of absolute judgment, and I maintain that for unidimensional judgments this span is somewhere in the neighborhood of seven." (Miller, 91) It appears people need to actually decompose larger and larger sets of information in order to grasp them rather than grasp them as wholes. It should be little surprise that over the history of the Internet web site design that the number of general, "top-level"

organizing topics for web collections regularly hover somewhere around the number seven.

While we have an external and contextual impetus for selecting a number of clusters between in the range of 5 to 9, do we have reason to worry that this rather unnatural selection will produce more or less "correct" or less "natural" clusters? From 20th century mathematicians and philosophers such as Kurt Gödel and Hilary Putnam we know that if there is a correct functional or computational model for something then we cannot justify it by the methods we used to generate it (Putnam, xv). In other words, the selection of a specific number form a computational standpoint seems internally arbitrary, yet it is our context that gives us the justification for selecting the number. Further, there is a strong sense that our domain of study, in this case the corpus of NC Health Info web pages, can be characterized by every possible functional description (Putnam, 121). Rigorously proven by Hilary Putnam, this notion that there is no naturalistic or objectively correct model when judged by formally descriptive/computationally descriptive means (specifically, "every ordinary open system is a realization of every abstract finite automaton" [Putnam, 121]), permits and encourages us to use non-functional means for setting standards. Taken as such, the computational features we select are arbitrary from a computational point of view.

Algorithmic criteria for selecting minimally distant centroids do not reliably select global minimal distances. Further, minimal distance is not necessarily reflective of "best," either (*e.g.,* see Banerjee, 2). Fortunately we do have a context for evaluating whether cluster size (number of cluster members) should be at least a factor in what constitutes "best".

I may elicit whether size matters as a criterion for evaluation with a thought experiment. Let us assume we have a website of 1000 documents, and we want to assign that collection into 5 groups, for the purposes of aiding navigation of the site. Let us also assume we randomly create two different schemes for the five groups. The first scheme puts exactly, at random, 200 documents into each of five groups. The other scheme puts 125 into two groups, and 250 into each of the other three groups. Which one is better? There's really no way to know, not at least without looking at the groups themselves. It may be that the second scheme happens to better match the distribution of document subjects. However, what if we throw away the second document schema, and instead replace it with a new one. The new one puts 996 of the documents in one group, and the remaining four documents each get their own groups. In this case it is obvious that this third schema is undesirable. It is as if this third schema does not help us navigate the collection at all—it is as if there are no clusters. We may not know the difference between two reasonable cluster representations on the basis of size alone, but we do know when we have a cluster representation that is

worthless based on size. Herewith this approach will be called the *Duh Factor* – we may not know which clusters are good by virtue of their size variations, but we can certainly tell when the clusters are bad. In other words, while we may not be able to decide which of a pair of birds is the better one, say, we may easily be able to determine the better bird when presented with a bird and a pig. "Duh, obviously, it isn't the pig."

2.2 Clustering and the Simple K-means algorithm

The Simple K-means algorithm was first developed in 1967 (MacQueen, 1967); an algorithmic process quite similar to Simple K-means was first applied to information retrieval soon after (Salton, 1971). The cluster hypothesis, first formulated in 1971, postulated that, "the associations between documents convey information about the relevance of documents to requests." (Jardine & van Rijsbergen, 1971). A more complex implementation of K-means called spherical K-means was recently studied in the context of text clustering (Dhillon, 2001) . While numerous attempts have been made to establish evaluative criteria for document clustering, no evaluative model has been established as a gold standard. The means for evaluating document clusters is a wide-open question.

Frequently clusters are evaluated by the classes-to-clusters methods, whereby clusters are evaluated in comparison to a previously established set of manually assigned classes or topics. There are a number of striking problems with such an

evaluation. For one, if there is no number agreement between the previously assigned topic set and the current number of clusters, we should not expect our clusterer to match or even remotely imitate the Further, we may be trying to get away altogether from manual assignments, instead looking for something that the machine algorithm may suggest on its own. Ultimately, and perhaps realistically, we may want to understand how to generate topics in the very real context of operating without manually assigned topics. In other words, not only is the use of manually assigned topics not convincingly helpful, it may go against the very intent of the use of unsupervised learning in the first place.

Simple K-means, as the name goes, is perhaps the simplest of algorithms that solves the clustering problem in a finite number of steps. Essentially a number of locations (k) in the problem space, called centroids, are selected such that they are at once random and far apart from one another. Then, for every point of the problem space, the closest centroid is identified. At this step we have our first cluster assignment, but the process has not yet finished. New centers are then calculated according to a distance-minimizing function, in this case a squared error function for *n* points, $J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$, such that $\left\| x_i^{(j)} - c_j \right\|^2$ is the distance from a point $x_i^{(j)}$ to the centroid c_j. The process then iterates again, reassigning points to minimally distant centroids, until no more minima

are found. The function is highly dependent upon the initial random seed (the random location of the initial centroid assignments), and the process is in no way guaranteed to locate the globally minimal *k* centroids. Overfitting, particularly with highly dimensional spaces such as is the case with text data, is a frequent concern, and numerous approaches have been taken to minimize the discovery of local minima that lead to overfitting. One such approach is to run the process repeatedly, taking the "best" result of the multiple trials, whereby "best" may be based on such metrics as the variance of cluster sizes or simply a more global or cumulative distance measure. Another approach is to "prematurely" terminate the repetition of the process arbitrarily so that centroids stop migrating before they have reached their minimal location according to the distance-minimizing function.

The Simple K-Means algorithm for the current project was chosen in part for two reasons: because of its simplicity and because of its availability in the Weka machine learning environment. *Simplicity*, because while we don't expect to derive truly mutually exclusive document clusters due to the relatively cross-topical nature of the consumer health document collection, it would provide for an easy-to-understand topic model. *Availability*, because given time constraints, Weka is a reasonably easy-to-implement, easy-to-use machine learning toolset. And lends itself well to the pedagogical aims of putting machine learning into practice.

3. Study model

The present study is based on a number of quantitative and qualitative measures issued differently at different stages in the process.

First, when the initial set of four feature representations are built, the number of distinct terms and their frequencies will be calculated. On the basis of those calculations, minimum and maximum term frequencies will be set.

Once these representations are "filtered" by removing both rare and trivial words, these filtered representations will be generated into a data format readable by Weka ("arff"). With these four arff files in hand, variants of the representations will be generated in a combinatorial fashion. Those combinations are based upon two general variations, binarization--whether term frequencies are used in the word-wordcount document vectors or whether those frequencies are binarized (1/0, or present/absent)—and *random projection*—how many of the attributes (words/stems/synset ids) are selected at random. Random projection is decomposed into two general classes—by percentage, or by a fixed number. It would be preferable if we could select a percentage rather than a fixed number, so that random projection ("RP") is relative to a representation's number of attributes rather than a fixed number. Further, both RP by percentage and RP at a constant value are tested at two different levels each. RP-percent is tested at 25% and 3%, and RP-constant is tested at 50

attributes and 150 attributes. Therefore, for each of the four general document representations (webtermtwostemmed, wttlragrstemmed, wtlrposstemmed, and wtlrposwnsyn, all reduced to a fixed set of 1499 documents with minimum term frequency of 5 and maximum of 1950), there are 10 variants produced. The forty representations are screened to remove representations that tend towards overfitting by running them in Simple K-means with k=5, 7 and 9, and using the percentage of maximum standard deviation of cluster size to identify and eliminate overfit representations.

It is hoped that one combination of the 10 will work well in all four representations, at least in terms of not overfitting, and if so, that combination will be used for closer inspection of the clusters. The remaining evaluation will proceed as follows: a base representation (either one based on webtermtwostemmed or wttlragr) will be established, to be used as a basis for comparison of the WordNet-based variant. The two will be compared, for at least two of the three k values, on the basis of their clusters, particularly on the basis of the most frequent terms in each cluster and whether they readily point out or elicit a cluster label. If the WordNet representation does not seem to work well, then the intermediate representation between WordNet and the SPECIALIST join set, the set reduced for part-of-speech, will be evaluated as the WordNet one was, by term frequency. In other words, we are trying to determine whether WordNet is a good feature representation for clustering

documents, and if not whether the other ones are at all. A second quantitative evaluation of term frequency will be performed for at least one of the two k for whichever of the two feature representations remain. That second quantitative measure, a measure I call TCFICF, is essentially TFIDF redefined for clusters rather than documents. It is hoped that this measure may better elicit good cluster labels. Finally, documents from the remaining clusters & representations will be qualitatively inspected to see if the clustering makes intuitive sense from a user perspective.

In some general sense, all variations described above will receive some analysis of their own, with the sole exception of the very last evaluation step.[3] A complete summary of the factors in the present study as well as their levels is contained in Table 1 below.

[3] It would not make sense to evaluate the value of the qualitative evaluation of documents in clusters, for evaluating intuition itself is beyond the scope of the current project.

1. **Word-level feature set**
    a. Raw words "WEBTERM"
    b. words joined on the SPECIALIST lexicon "WTLRAGR" (stemmed & stopped)
    c. Only N, V, ADJ, ADV from the above join "WTLRAGRPOS" (stemmed & stopped)
    d. Only synset ids (unique synset/concept identifiers) based on the last table "WTLRPOSWNSYN" (not stemmed)

2. **Feature reduction**
    a. Setting minimum and maximum term frequencies as well as minimum terms per document
    b. Term frequency vs binarized
    c. Random projection vs no random projection
    d. Constant random projection vs. proportional random projection
    e. Random projection, 50 atts vs random projection, 150 atts
    f. 3% random projection vs. 25% random projection

3. **Cluster sizes**
    a. K=5
    b. K=7
    c. K=9

4. **Cluster evaluations**
    a. 10 most frequent terms in cluster
    b. 10 highest ranked terms, by TCFICF

Table 1. A summary of experimental factors and levels

3.1 A brief note on the computing environment

All preprocessing was performed on jade.ils.unc.edu, a UNIX-based workstation, and the oracle db is also located on jade. All machine learning tasks were performed using Weka 3.4.4 on the Xeon 2.8 GHz 96 hour serial compute node on baobab.unc.edu, UNC's high performance Linux-based Beowulf cluster. Much of the data analysis was performed on the author's local laptop or home workstation, both Windows-based.

4. The classic text mining model

The present study uses the now-classic stepwise text mining process, described below:

1. Corpus selection
2. Preprocessing & generating preliminary data sets
3. Selecting & setting multiple feature representations
4. Learning to reduce candidate feature representations
5. Analysis:
    a. Evaluating clusters of remaining feature representations quantitatively
    b. Qualitatively evaluating clusters from an even smaller subset of feature representations.

This process is essentially identical to the knowledge discovery process illustrated in Figure 1 below.

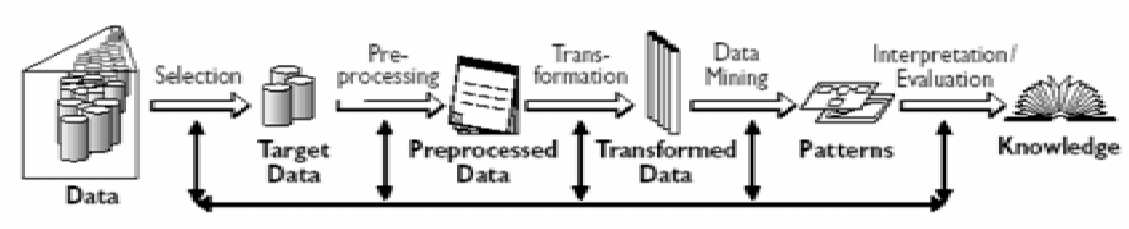

Figure 1. The KDD Process[4]

The particular variation of this general model, a rather common variation wholly

[4] Image taken from Fayyad, U., Piatetsky-Shapiro , G., Smyth,P. The KDD process for extracting useful knowledge from volumes of data, *Communications of the ACM*, 1996, vol. 39, no. 11.

consistent with the above illustration, is in the repetition of steps 3 and 4, and the progressive whittling down of candidate feature sets in that repetition.

4.1 Corpus

The corpus used in the present study is a set of web pages from the North Carolina Health Info website (http://nchealthinfo.org). The collection of pages is curated by employees of UNC Health Sciences Library in conjunction with the School of Information and Library Science. The collection of pages is in essence a manually curated portal, whereby consumer-health-related web pages related to heath care in the state of North Carolina are added to the site. The pages contained therein cover the entire spectrum of health care and consumer health information for the state of North Carolina, from listing of physicians' names and addresses to clinic and hospital websites to information about support groups, alternative medicine, medical insurance, and general health issues. The pages appear to have been authored in a wide variety of ways, from MS FrontPage to Macromedia Dreamweaver to manual authoring, in various and not always valid html-based formats.

The collection used here comes from a spidering of approximately 3600 html files in the site collection executed back in the Summer of 2004 for a research assistant of Dr. Catherine Blake. Of those 3600 files, approximately 1800 appear to have been successful downloads of non-zero-length complete html files. For the

present study, 1499 of those files are used; these 1499 documents meet minimum length requirements for all of the four primary feature representation classes evaluated in the present study.

4.2 Preprocessing

Preprocessing of the files took place in November of 2004 for a previous data mining project. A series of java functions were written to parse out the particularly inconsistent broad variety of HTML, CSS, and JavaScript code contained in the page files. Because there were as many variations of invalid HTML uses as there were documents, a number of SED scripts were written to pre- and post-process the data passing into the Java HTML-parsing classes. An escape code class written by Dr. Catherine Blake was used as part of the preprocessing process, in order to clean out or replace escape codes, and it was extended and enhanced for the peculiarities of the corpus with additional SED scripts and Java code.

The purpose of the preprocessing was not only to extract the free-text words from the documents but also to maintain a record of their position—in other words, the purpose was to record the structure of the document. The format of processing records not only the document containing the word but also whether the text was in a header or a paragraph, and in what section, paragraph, sentence, and at what sentence position each word resided. For the present

study no positional information was utilized; only document ID-word pairs were used.

Preprocessing was completed when the data from the preprocessing step was loaded into an oracle table on jade, a table called “webterm.”

The preprocessing steps are illustrated at the top of Figure 2 below, depicted in relation to the feature representation building process.

4.3 Building the four primary feature representations: webtermtwostemmed, wttlrstemmed, wtlrposstemmed, and wtlrposwnsyn

Once the corpus was preprocessed and loaded into the webterm table, a set of useable representations needed to be built for the purposes of the study. A complete illustration of the feature representation process is illustrated in Figure 2 below.

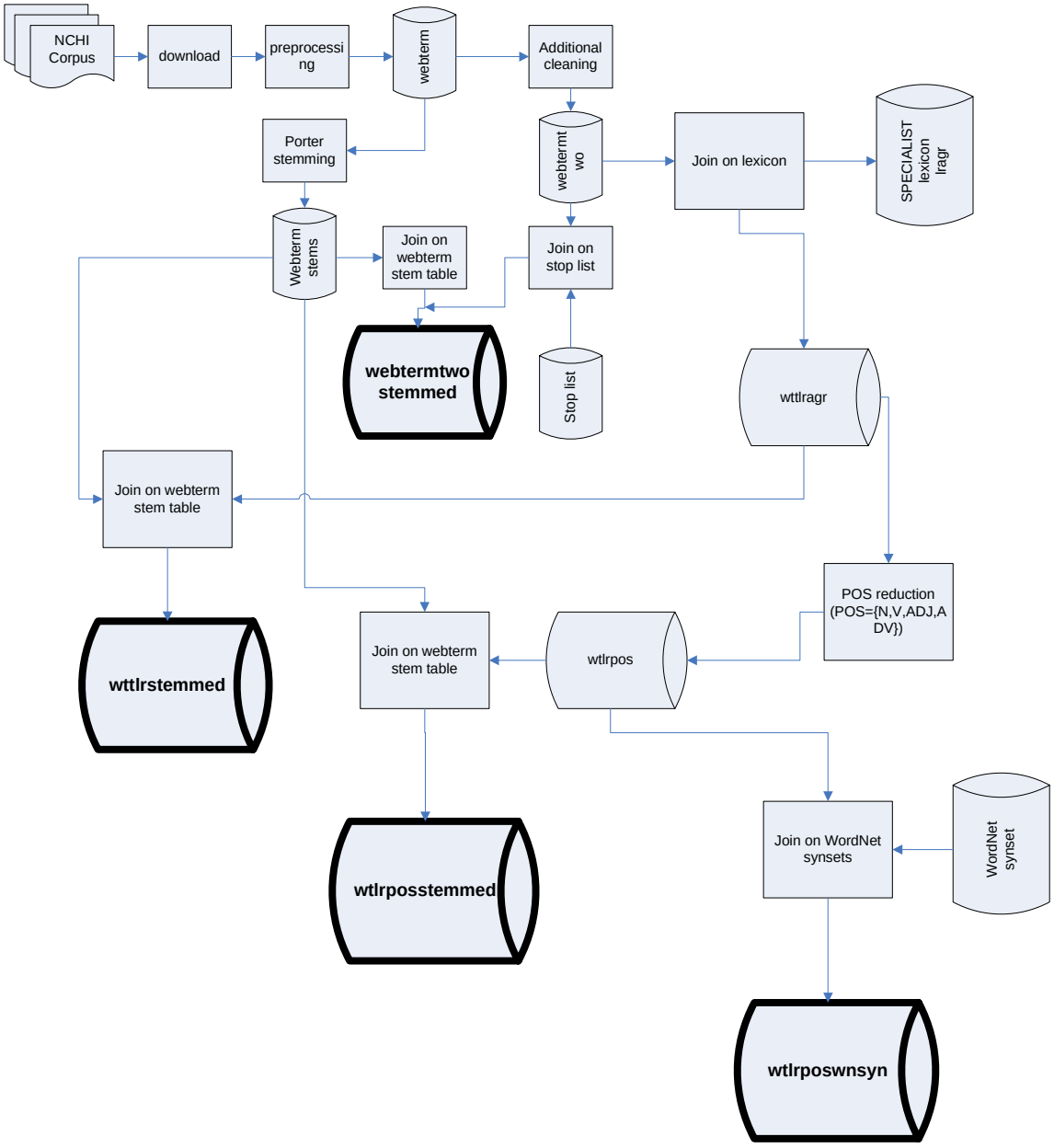

Figure 2. Building the basic feature representations

4.3.1 Building webtermtwostemmed

Webtermtwostemmed was constructed from the initial webterm table, but an intermediate representation, webtermtwo, was constructed that contained all the pre-stemmed results. Much of the process from webterm to webterm two

involved cleaning and joining on a stop word list. Any detected remaining web code was removed, along with any strings containing non-numeric characters other than hyphens and loaded into a table called webtermtwo. This table, webtermtwo, provides the basis for the building of all the other three remaining representations—wttlrstemmed, wtlrposstemmed, and wtlrposwnsyn. Finally, a stem list generated from the web term table using the Porter stemmer was outer-joined on webtermtwo table (*i.e.*, anything stemmable was replaced with its stem, while anything not containing a stem remained, namely so that stemming would not act as a filter but instead purely as a feature reduction step) and the results were loaded into webtermtwostemmed.

4.3.2. Building wttlrstemmed

The wttlrstemmed table is essentially the webtermtwo table joined on the NLM's SPECIALIST medical lexicon table, named 'lragr' in the data base. The join was chosen initially as another data cleaning step, as from early on it appeared nearly impossible to anticipate all html noise from the large collection of corpus documents. The assumption was that anything not in the SPECIALIST lexicon was ultimately not a word. The result of this join was placed into a table called webtermtwolragr, and then this table was outer joined on the stem list and the results placed into wttlrstemmed.

4.3.3. Building wtlrposstemmed

The aforementioned webtermtwolragr table formed the basis for the part-of speech reduction representation. The reason for the part-of-speech reduction was that it is a necessary intermediary step before creating a WordNet-based representation. WordNet contains only nouns, verb, adjectives and adverbs. Since SPECIALIST contains part of speech information in the form of a numeric index value, it was easy to reduce the webtermtwolragr table by filtering out only those terms considered nouns, verbs, adjectives or adverbs by the SPECIALIST lexicon. It is important to note that this is being performed in the absence of any POS tagging and is merely a filtering step. However rather by accident this part-of-speech reduction may prove to be a valuable reduction step in and of itself. The POS-reduced terms were inserted into the wtlrpos table and then joined on the stem table and the resulting data was inserted into a table called wtlrposstemmed.

4.3.4. Building wtlrposwnsyn

The ultimate representation of the study, the WordNet based representation, was built upon what I will term a "naïve " approach. Typically WordNet terms must be first POS-tagged in order to identify their synset identifiers, namely since all synset identifiers (concept identifiers) are defined by a word/POS pair. We do not know the specific POS of each word in our initial representation, but, we

have a pretty good idea each term is of a POS in WordNet. The wtlrpos table was joined on the terms in a WordNet synset table. The WordNet synset table contains all 2 million-plus unique synset identifiers plus the terms they represent, along with POS and word-sense information. In many cases multiple words from wtlrpos mapped onto single synset ids (synonymy), while in other cases single words mapped to multiple synset ids (ambiguity). While having POS tagging information up front would have reduced the scale of ambiguity, it would not have eliminated it altogether. We do however have an interest in whether WordNet can be used effectively in such a "naïve" fashion; in fact, it constitutes one of the central questions of the present study.

4.4 Reducing & refining the four primary representations

In order to create representations I might be able to use for the machine learning/clustering experiments, I first needed to identify and screen out terms that occur either too infrequently or too frequently. Terms that happen too frequently might likely tend to be trivial terms, trivial to the collection, such as "health" or "north" or "carolina." Terms not happening frequently enough, such as terms that occur only once or twice, will likely tend only to add noise to our representation, as they are so statistically insignificant taken one at a time but as a collection the most infrequent terms might take up a good amount of our data points, unless they are of course filtered out. We should expect from Zipf's

Law that the number of these infrequent terms should be quite high, while there should be fewer and fewer terms of higher and higher frequency.

Competing against this general interest to remove insignificant or trivial terms is the need to preserve valuable attributes. In order to evaluate the tradeoffs of eliminating features, every representation table was evaluated for distinct term frequencies in depth (or, in the case of the stemmed tables, stem frequencies, or, in the case of the WordNet-based representation, both synset id frequencies and term frequencies). The first attempt, captured in detail in Appendix 1, was unfortunately confounded by an erroneous join on the SPECIALIST lexicon. The second attempt, captured in detail in Appendix 2, was more successful. Tables 2 through 10 below reflect an abbreviated version of the data generated in the review. The full data set may be viewed in Appendix 3.

A guiding principle in selecting minimum and maximum levels for me was that I wanted to be conservative about selecting a maximum term frequency threshold yet more aggressive about setting a minimum one. The reason was that high frequency terms that might be lost may actually be a dominant feature of a subgroup of the collection & represent something essential to a set of documents, yet low frequency terms seem not to provide much insight about their documents while keeping the dimensionality of the data sets high.

WEBTERM

before stemming & stoplist

table name: webterm

number of unique terms	number of documents	number of documents w/ >10 terms	no of terms w/term frequency=1	no of terms w/term frequency=5	no of terms w/term frequency <10	no of terms w/term frequency <100	no of terms w/term frequency > 250	no of terms w/term frequency > 1000	most frequent term
45078	1854	1740	21558	1319	37563	44115	359	76	_SYM_comma (36689)
100%	100%	94%	48%	3%	83%	98%	1%	0%	

after lower cased, stoplisted, & stripped of strings containing non-alpha characters

table name: webtermtwo

number of unique terms	number of documents	number of documents w/ >10 terms	no of terms w/term frequency=1	no of terms w/term frequency=5	no of terms w/term frequency <10	no of terms w/term frequency <100	no of terms w/term frequency > 250	no of terms w/term frequency > 1000	most frequent term
24888	1822	1614	11032	831	20112	24203	235	36	health (6569)
55%	98%	87%	44%	3%	81%	97%	1%	0%	

after stemming

table name: webtermtwostemmed

after stemming

table name: webtermtwostemmed

number of unique stems/terms	number of documents	number of documents w/ >10 stems	no of stems w/stem frequency=1	no of stems w/stem frequency=5	no of stems w/stem frequency <10	no of stems w/stem frequency <100	no of stems w/stem frequency > 250	no of stems w/stem frequency > 1000	most frequent stem
19213	1822	1614	8753	578	15417	18490	278	44	health (6577)
43%	98%	87%	46%	3%	80%	96%	1%	0%	

number of unique terms not stemmed: 584

Tables 2-4. Term frequencies, webterm-based representations

WTLRAGR

before stemming & stoplist

table name: webtermtwolragr

number of unique terms	number of documents	number of documents w/ >10 terms	no of terms w/term frequency=1	no of terms w/term frequency=5	no of terms w/term frequency <10	no of terms w/term frequency <100	no of terms w/term frequency > 250	no of terms w/term frequency > 1000	most frequent term
14392	1820	1604	4394	611	10322	13745	221	35	health (6569)
100% (58% of webtermtwo)	100%	87%	31%	4%	72%	96%	2%	0%	

after stemming

table name: webtermtwolragrstemmed

number of unique stems	number of documents	number of documents w/ >10 stems	no of stems w/stem frequency=1	no of stems w/stem frequency=5	no of stems w/stem frequency <10	no of stems w/stem frequency <100	no of stems w/stem frequency > 250	no of stems w/stem frequency > 1000	most frequent stem
9241	1820	1604	2547	364	6164	8558	263	43	health (6576)
100%	100%	88%	28%	4%	67%	93%	3%	0%	

The join on lragr sheds 10496 terms. Almost all of the lost terms are either 1) proper names(~ 60%);
2) obscure organizational acronyms (~10%); 3)spanish words (~10%), 4)misspelled words (~15%), or
5) odd words with non-alpha characters inside (typically two terms separated by ellipses or typos ~3-5%);
try SQL>select * from losttermtmp sample(0.3); to evaluate

Tables 5-6. Term frequencies for wtlragr-based representations

WTLRAGRPOS

before stemming & stoplist

table name: wtlrpos

number of unique terms	number of documents	number of documents w/ >10 terms	no of terms w/term frequency=1	no of terms w/term frequency=5	no of terms w/term frequency <10	no of terms w/term frequency <100	no of terms w/term frequency > 250	no of terms w/term frequency > 1000	most frequent term
12707	1792	1593	3912	536	9098	12116	206	33	health (6569)
100% (88% of webtermt wolragr)	100%	89%	31%	4%	72%	95%	2%	0%	

after stemming

table name: wtlrposstemmed

number of unique stems	number of documents	number of documents w/ >10 stems	no of stems w/stem frequency=1	no of stems w/stem frequency=5	no of stems w/stem frequency <10	no of stems w/stem frequency <100	no of stems w/stem frequency > 250	no of stems w/stem frequency > 1000	most frequent stem
7723	1792	1593	2129	295	5086	7096	249	41	health (6576)
100% (61% of wtlrpos)	100%	89%	28%	4%	66%	92%	3%	1%	

Tables 7-8. Term frequencies for POS-reduction-based representations

WTLRPOSWNSYN

table name: wtlrposwnsyn

number of unique terms	number of documents	number of documents w/ >10 terms	no of terms w/term frequency=1	no of terms w/term frequency=5	no of terms w/term frequency <10	no of terms w/term frequency <100	no of terms w/term frequency > 250	no of terms w/term frequency > 1000	most frequent term
8796	1792	1499	2516	363	6026	8323	174	31	health (6569)
	100%	84%	29%	4%	69%	95%	2%	0%	

number of unique IDs	number of documents	number of documents w/ >10 IDs	no of IDs w/ID frequency=1	no of IDs w/ID frequency=5	no of IDs w/ID frequency <10	no of IDs w/ID frequency <100	no of IDs w/ID frequency > 250	no of IDs w/ID frequency > 1000	most frequent ID
25634	1792	1737	4658	921	13440	22698	1206	232	113628836 (as terms 'health' and 'wellness' occurs 6799 times)
	100%	97%	18%	4%	52%	89%	5%	1%	

Tables 9-10. Term & synset ID frequencies for WordNet-based representations

Ultimately, the minimum term frequency of 5 for each representation was selected, while a maximum frequency of 1950 was chosen. By selecting 1950 as

my maximum term frequency, I was able to chop out quite trivial terms like "health" yet retain, in every case, the term "cancer." Selecting the maximum frequency that high kept this effort conservative. On the other hand, by selecting 5 as the minimum term frequency, the dimensionality of each data set was dramatically reduced. I might have been more aggressive and set the minimum value higher, but ultimately I was afraid of losing needed information.

I also set the minimum document length at 10. This was a rather arbitrary decision. It seemed that documents with less than 10 features wouldn't constitute enough content, and it's not unusual to find web pages with very little content.

When reducing the four primary representations by limiting the terms sets by minimum term frequency of 5 and max of 1950 (see Figure 3 below), and when selecting the minimum document feature length at 10, some documents would essentially be lost, and the number lost would vary from representation to representation. I wanted to ensure I was using the exact same documents for all four representations, so I selected the lowest common denominator set of documents: 1499 documents remained in the wtlrposstemmed_min5_max1950 representation, and so the other three representations, all of which contained all 1499 documents, were restricted to just those 1499 documents.

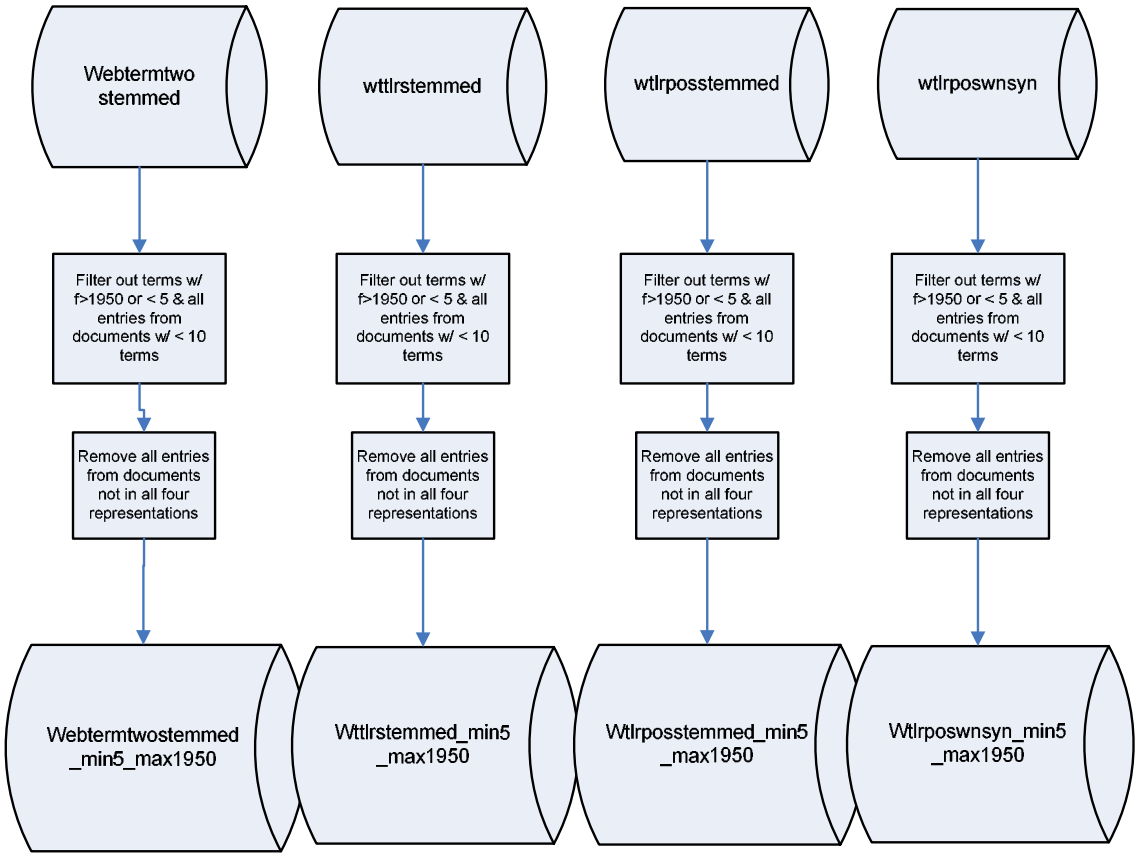

Figure 3. Reducing the four basic representation by term frequencies

4.4.1 Distinct terms counts, before and after reductions

webterm: 45078
webterm_min5_max1950: 11650

webtermtwostemmed: 19213
webtermtwostemmed_min5_max1950: 5485

wttlrstemmed: 9241
wttlrstemmed_min5_max1950: 4172

wtlrposstemmed: 7723
wtlrposstemmed_min5_max1950: 3603

wtlrposwnsyn, synsetids: 25,694
wtlrposwnsyn, terms: 8796
wtlrposwnsyn_min5_max1950, synsetids: 15596
wtlrposwnsyn_min5_max1950, terms: 5828

It should be readily apparent that all four of the focus feature representations show a good deal of dimensionality reduction, with the largest less than 35% of the original dimensionality of the initial representation, and all but one hovering at around 10% of that original dimensionality. We should expect this dimensionality reduction to at least make our machine learning experiments more efficient.

4.5 Generating data sets from the tables

With tables with the four focus feature representations in hand, the next task was to extract the data from the tables and construct data in a format readable by the Weka Data Mining system. The format of choice for Weka is the attribute-relation file format (ARFF), but the most optimal format for text mining, given the sparsity of the document term vectors, is a specific type of arff format known as a sparse arff. The sparse matrix version of arff is a perfect solution to compactly representing text mining data sets because the terms with 0 frequency in a document do not need to be explicitly represented.

A java class authored by Dr. Catherine Blake named DBAccess was extended for the purposes of extracting the data, and a new arff-generating function was constructed in order to generate the sparse arff data. Unfortunately no sparse arff generating utility is currently available publicly or is included in the Weka

toolkit. Creating the sparse matrix-generating function was complicated by an equally sparse amount of documentation about the format, particularly with regard to its use for unsupervised learning.

Each of the four central representations, *webtermtwostemmed_min5_max1950*, *wttlrstemmed_min5_max1950*, *wtlrposstemmed_min5_max1950*, and *wtlrposwnsyn_min5_max1950*, were used to generate four corresponding arff files. Each of these arff files were then subsequently used to generate variants of these representations. Variants included all combinations of the following features: term frequencies binarized (or not—default), random projection @25%, 3%, 50, and 150. All variants (depicted in Figure 4 below), 10 for each of the four central representations, 40 in all, were rendered using the appropriate Weka filter classes at the command line.

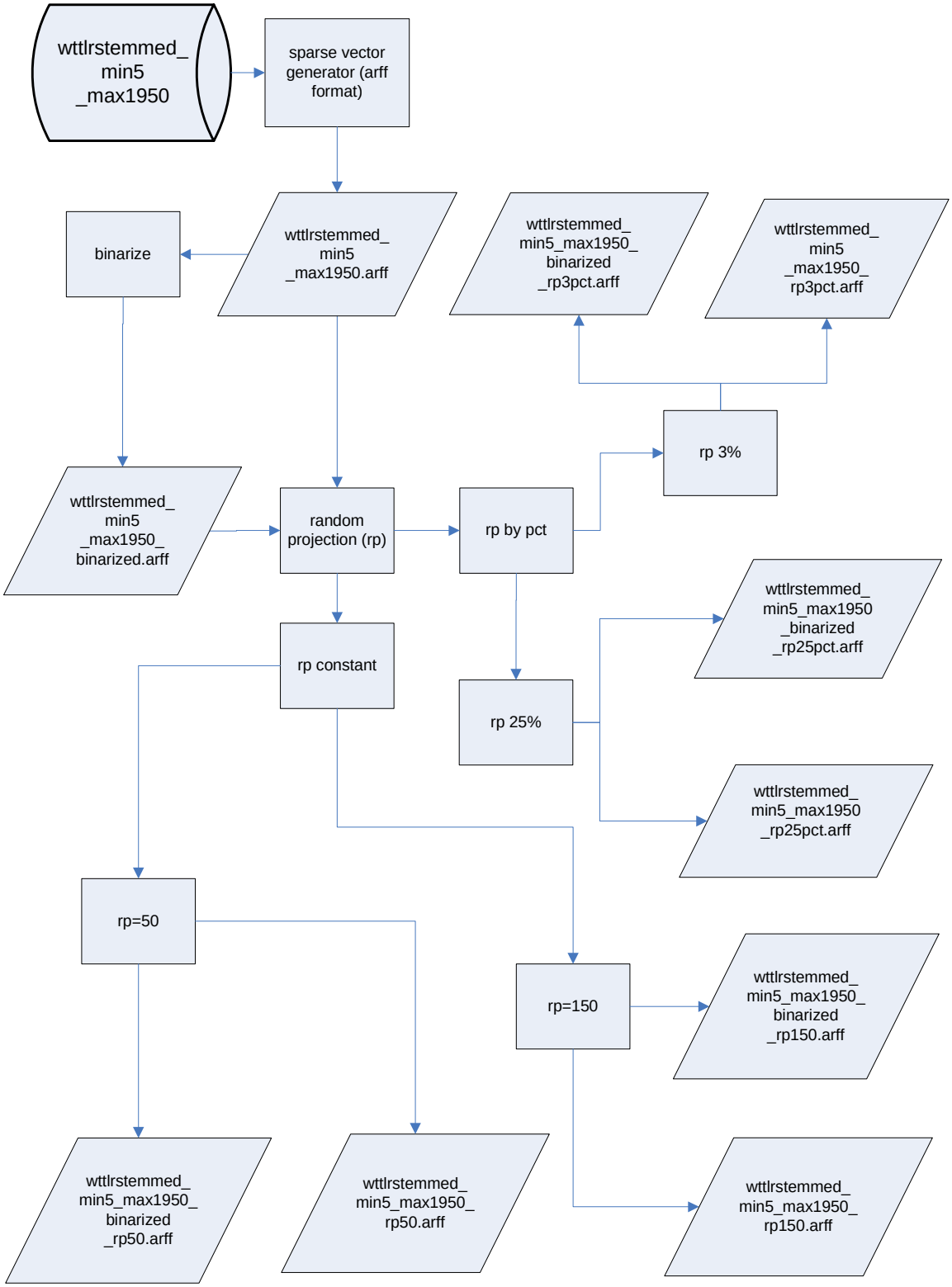

Figure 4. Generating 10 feature representations in arff format from base representation table *wttlrstemmed_min5_max1950*

4.6 Learning experiments

With the 40 candidate representations in arff format in hand, I was ready to begin the actual clustering experiments[5]. Clustering experiments were run using Weka's Simple K-means implementation for k=5,7, and 9 for all 40 arff representations, 120 experiments in all. Experiments were performed on the baobab Linux Beowulf high-performance computing cluster. Output generated contained cluster membership identifiers for each instance/document and summary statistics about the size of each cluster.

4.7 Screening clusters for overfitting

The first level of evaluation—reviewing cluster sizes for the 120 cluster experiments--is for the sole purpose of screening the 40 representations such that the resulting representations that at least to clusters that might have some promise for document clustering. That is to say, the reason for the proliferation of representations is that overfitting has been a problem, and the author has no prior experience with discovering what it is that might sway us from overfitting. A more experienced text miner might not need to go through this process. This step's aim is to identify representations that do not tend to lead to the formation of monster clusters.

[5] In truth, I ran countless (approximately 100) clustering experiments with the initial arff files before I attempted the 120 structured experiments. In those initial test experiments I experienced a terrible problem with overfitting so with a little curiosity I tried to use what Weka offered that might reduce the overfitting problem. These experiments first helped me debug problems with the sparse arff generator, and then they

This step allows up to observe whether such factors as k, binarization vs. tf, the four core representations, and random projection lead to or prevent the formation of monster clusters. For convenience's sake, I will also use this step to pick a small subset of results (cluster sets that do not overfit) for further cluster evaluation. In other words, there may be more representations that are worthy of further evaluation than are actually subjected to greater scrutiny in later evaluations. Given the scope of the current project and time demands, performing these additional evaluations to all non-overfitting representations is simply unrealistic.

Checking for monster clusters will be executed by measuring the standard deviation of cluster sizes for all representations, all k, and will be broken down for evaluating the difference between binarization and frequency as well as the use of various values for random projection. The metric used to detect unbalanced clusters, called FACTOR, is simply the standard deviation of a cluster model as a percentage of the maximum possible standard deviation. A very high score (75% or above) indicates overfitting--a general failure of the algorithm to avoid focusing in on local minima for the given representation.[6]

---

led me to binarization and random projection. Instance normalization, EM clustering, and Principle Component analysis were heavily explored, but not in any structured way.

[6] A similar measure was used by Efron, et al (2004), but in order to choose k. It is the present author's feeling that overfitting—a phenomenon related to the size of the data set—should be controlled by other factors, ones related to the input size rather than the quantity of clusters. FACTOR should be relatively independent of k, given there are no "natural" correct cluster representations of a data set. The lack of

It must be restated that while we do not want to necessarily obtain perfectly balanced clusters & have no idea what the right balance is, we do know don't want overfit, unbalanced models (the duh hypothesis). An overfit model means we are not really generating any clusters, certainly not form the point of view of navigating a collection of web documents by a 5 to 9 term menu.

As argued earlier, we should expect that cluster divisions should be somewhat arbitrary—*e.g.*, if I give you 100 documents and separate them into two piles, that separation reflects nothing more "inherent" about the documents than if I had you separate the 100 documents into 5 piles—you'd probably set different criteria for fiveness than you would for two, but those criteria cannot be differentiated by deciding which criteria set is the more "natural."

For 1499 documents and k=5, the maximum possible std deviation is based on cluster sizes={1,1,1,1,1495}. The SimpleKMeans does not assign cluster values of zero; one is the minimum size. The calculation for the denominator for FACTOR is shown in Table 11.

---

variance between k in FACTOR shown in the present study at the very least does not reject the author's application of Putnam's theorem.

	k=5	k=7	k=9
	K=5	K=7	K=9
			1491
			1
		1493	1
		1	1
	1495	1	1
	1	1	1
	1	1	1
	1	1	1
	1	1	1
max SD	**668.14**	**563.92**	**496.67**

Table 11 . Maximum standard deviation calculations, 1499 documents.

As discussed in section 4.3.2., the join on the SPECIALIST was initially conducted as a data cleaning set. As a result of these experiments, it seems clear that this was a useful step, as the representation preceding this join, the webtermtwostemmed representation, consistently led to parsing errors in Weka, for about half of its 30 experiments. Characters or character sequences unreadable by Weka remained in the data. This leaves us with only 90 experiments remaining to evaluate—k=5,7, 9 for 10 different versions of three of the four core representations.

4.7.1. Results

For all 90 experiments, the most consequential factor "causing" overfitting is term frequency, as shown in Table 12 below. All statistics are available in Appendix 4.

		bin	tf
binarized	*SD*	314.270626	525.641
vs. tf	*%MAX*	51.54%	91%
	+/-	+/- 3%	+/- 6%

Table 12. Binarization vs. term frequency, 90 clustering experiments

For the purposes of the current study only binarized representations will be inspected, thus reducing our clustering experiment set to 45, and our candidate representation set to 15.

Fortunately for the purposes of the current study, it does not appear from the results that varying either k or the representation base (wttlr, wttlrpos, wtlrposwnsyn) shows much variance in cluster size, variance that for particular values leads to overfitting.

It does appear that Random Projection, in any form, seems to benefit us at least in terms of preventing any further tendency towards overfitting when random projection is not used. No random projection tends towards borderline overfitting, while random projection does not. See Tables 13 and 14 below.

No rp vs. rp	No rp	rp
SD	416.56849	288.69616
%MAX	72.13%	50.04%
+/-	8.08%	4.36%

Table 13. Random Projection or not, all 45 binarized experiments

k5-9	No rp	rp3pct	rp25pct	rp50	rp150
SD	416.5685	305.0295	264.6503	290.02193	295.08291
%MAX	72.13%	52.97%	45.65%	50.30%	51.24%
+/-	8.08%	3.38%	4.40%	2.56%	3.22%

Table 14. Random projection variants, all 45 binarized experiments

While Table 14 does not show that random projection necessarily produces "better" clusters than no random projection, it does, in all four variations here, do a good job of keeping away overfitting.

Based on the above results, further it was decided to conduct further inspection of only the clusters produced using binarization and a random projection of 25%. In terms of cluster size, none of the random projections are better than the other in any way. Here a heuristic applies. Given a choice between a fixed number and a percentage, especially for the purposes of evaluating representations of differing dimensionality, it would be best to be able to use a percentage-based random projection. Given the choice between preserving 3% or 25% of the original attributes, I would choose the latter. Therefore, random projection at 25% was chosen. Figure 5 shows the pruning of candidate feature representations. We are left at this stage with 3 candidate representations and 9 experiments.

	→ narrowed to		→ narrowed to	
webtermtwostemmed_min5_max1950				
webtermtwostemmed_min5_max1950_rp3pct				
webtermtwostemmed_min5_max1950_rp25pct				
webtermtwostemmed_min5_max1950_rp50				
webtermtwostemmed_min5_max1950_rp150				
webtermtwostemmed_min5_max1950_binarized				
webtermtwostemmed_min5_max1950_binarized_rp3pct				
webtermtwostemmed_min5_max1950_binarized_rp25pct				
webtermtwostemmed_min5_max1950_binarized_rp50				
webtermtwostemmed_min5_max1950_binarized_rp150				
wttlragrstemmed_min5_max1950				
wttlragrstemmed_min5_max1950_rp3pct	→	wttlragrstemmed_min5_max1950_binarized		
wttlragrstemmed_min5_max1950_rp25pct		wttlragrstemmed_min5_max1950_binarized_rp3pct		
wttlragrstemmed_min5_max1950_rp50		wttlragrstemmed_min5_max1950_binarized_rp25pct		
wttlragrstemmed_min5_max1950_rp150		wttlragrstemmed_min5_max1950_binarized_rp50		
wttlragrstemmed_min5_max1950_binarized		wttlragrstemmed_min5_max1950_binarized_rp150		
wttlragrstemmed_min5_max1950_binarized_rp3pct		wtlragrposstemmed_min5_max1950_binarized		
wttlragrstemmed_min5_max1950_binarized_rp25pct		wtlragrposstemmed_min5_max1950_binarized_rp3pct	→	**wttlragrstemmed_min5_max1950_binarized_rp25pct**
wttlragrstemmed_min5_max1950_binarized_rp50		wtlragrposstemmed_min5_max1950_binarized_rp25pct		**wtlragrposstemmed_min5_max1950_binarized_rp25pct**
wttlragrstemmed_min5_max1950_binarized_rp150		wtlragrposstemmed_min5_max1950_binarized_rp50		**wtlragrposwnsyn_min5_max1950_binarized_rp25pct**
wtlragrposstemmed_min5_max1950		wtlragrposstemmed_min5_max1950_binarized_rp150		
wtlragrposstemmed_min5_max1950_rp3pct		wtlragrposwnsyn_min5_max1950_binarized		
wtlragrposstemmed_min5_max1950_rp25pct		wtlragrposwnsyn_min5_max1950_binarized_rp3pct		
wtlragrposstemmed_min5_max1950_rp50		wtlragrposwnsyn_min5_max1950_binarized_rp25pct		
wtlragrposstemmed_min5_max1950_rp150		wtlragrposwnsyn_min5_max1950_binarized_rp50		
wtlragrposstemmed_min5_max1950_binarized		wtlragrposwnsyn_min5_max1950_binarized_rp150		
wtlragrposstemmed_min5_max1950_binarized_rp3pct				
wtlragrposstemmed_min5_max1950_binarized_rp25pct				
wtlragrposstemmed_min5_max1950_binarized_rp50				
wtlragrposstemmed_min5_max1950_binarized_rp150				
wtlragrposwnsyn_min5_max1950				
wtlragrposwnsyn_min5_max1950_rp3pct				
wtlragrposwnsyn_min5_max1950_rp25pct				
wtlragrposwnsyn_min5_max1950_rp50				
wtlragrposwnsyn_min5_max1950_rp150				
wtlragrposwnsyn_min5_max1950_binarized				
wtlragrposwnsyn_min5_max1950_binarized_rp3pct				
wtlragrposwnsyn_min5_max1950_binarized_rp25pct				
wtlragrposwnsyn_min5_max1950_binarized_rp50				
wtlragrposwnsyn_min5_max1950_binarized_rp150				

Fig. 5. Pruning the candidate feature representations list for further evaluation

4.8. Evaluating clusters from the WordNet-based representation

We have reduced the representations to the following factors:

- Wttlr vs. wtlrpos vs. wtlrposwnsyn
- all binarized
- all RP @ 25 %

At this point in the study we have no further need to reduce the candidate set of representations. We have now have control—an even playing field, if you will—for comparing the base representation wttlrstemmed (min=5, max=1950, binarized, RP=25%, 1499 documents) to the naïve WordNet representation, wtlrposwnsyn (min=5, max=1950, binarized, RP=25%, 1499 documents), and we have preserved the corresponding intermediate model, wtlrposstemmed (min=5, max=1950, binarized, RP=25%, 1499 documents).

While data for three experiments (k=5,7,9) for each of the three representations are available at this point in the study, I only need to look first at two different experiments for two of the representations. Since the point is to evaluate clustering using naïve WordNet, I will look at k=5,7 for the WordNet-based representation and comparing it to our "basis" representation, the wttlrstemmed representation.

In order to compare these clusters I chose to generate the list of the top 10 most frequent terms / stems for each cluster (see Efron, 2004 for a similar approach). I then identified which of those top 10 terms from each cluster did not appear in the other clusters, and tried to at least intuit a sense of topic given those features. Since the terms / stems are the features, this should be a good choice.

Tables 15 and 16 below provide a basis for comparing the most frequent terms for wttlr vs. wtlrposwnsyn at k=5; Tables 17 and 18 provide the same comparison but for k=7.

4.8.1. Results of wttlrstemmed vs. wtlrposwnsyn, k=5

Table 15 shows the most frequent terms in each cluster for wttlrstemmed. Cluster 1 seems vaguely suggestive of pages related to contact information: *phone, mail, courier, box.* Cluster 2 seems suggestive of cancer treatment, therapy, and the like. Cluster 3 seems to suggest patient education and consumer information-seeking assistance; Cluster 4 seems related to issues related to medical supplies, while Cluster 5 seems possibly suggestive of physical therapy, particular athletics-oriented therapy. What is concerning at this point is the degree to which terms overlap. While I expect terms like "hospital" and "support" to show up frequently in multiple clusters, I didn't expect terms like

"cancer" to show up quite so high in three clusters. Further, the labels are not "popping out."

cluster 1 freq	cluster 1 size (pct)	10 most frequent terms	freq	cluster 2 freq	cluster 2 size (pct)	10 most frequent terms	freq	cluster 3 freq	cluster 3 size (pct)	10 most frequent terms	freq
787	53%	**phone**	**1361**	226	15%	**breast**	**982**	378	25%	_tim	703
		director	**1246**			cancer	939			**help**	**426**
		mail	**1103**			support	865			hospit	407
		courier	**954**			**resourc**	**738**			**librari**	**382**
		contact	782			therapi	579			patient	354
		box	**676**			**treatment**	**570**			**educ**	**331**
		street	625			patient	503			commun	325
		hospit	608			contact	486			hour	324
		commun	502			hospit	486			support	302
		site	**473**			**surgeri**	**379**			**call**	**282**

cluster 4 freq	cluster 4 size (pct)	10 most frequent terms	freq	cluster 5 freq	cluster 5 size (pct)	10 most frequent terms	freq
33	2%	_tim	433	74	5%	**medicin**	**499**
		product	**402**			**sport**	**442**
		children	**387**			therapi	435
		assist	**385**			**suit**	**309**
		insur	**337**			**abc**	**287**
		hour	284			**physic**	**285**
		cancer	274			cancer	245
		plan	**269**			street	241
		prosthes	**262**			**clinic**	**240**
		commun	258			patient	239

Bold indicates term that does not appear in top 10 list for the other clusters

Table 15. wttlrstemmed binarized @ 25% RP, k=5

Table 16 shows the clusters for the WordNet representation at k=5. The number of terms in the top 10 lists only in 1 cluster has dropped dramatically. What is interesting is that we are in effect seeing the same types of clusters, but the overlapping dominant features is significantly higher in the WordNet representation. In other words, from this approach it appears that the WordNet representation does not work as well as its current competitor. Any possibility for cluster labels seems highly strained at best, if not downright impossible.

cluster 1 freq	cluster 1 size (pct)	10 most frequent terms	freq	cluster 2 freq	cluster 2 size (pct)	10 most frequent terms	freq	cluster 3 freq	cluster 3 size (pct)	10 most frequent terms	freq
845	56%	**font**	**625**	380	25%	contact	481	148	10%	program	287
		family	559			program	402			**am**	**258**
		library	**530**			hospital	330			hours	251
		hospital	440			community	290			family	232
		color	**415**			family	280			hospital	215
		program	407			site	280			community	201
		site	400			cancer	242			cancer	192
		community	377			**education**	**233**			breast	188
		support	343			treatment	230			**available**	**182**
		department	333			**board**	**221**			**insurance**	**177**

cluster 4 freq	cluster 4 size (pct)	10 most frequent terms	freq	cluster 5 freq	cluster 5 size (pct)	10 most frequent terms	freq
27	2%	cancer	664	98	7%	yes	1461
		contact	577			**phone**	**1271**
		medicine	**433**			director	1205
		service	428			department	1094
		program	390			**mail**	**1066**
		family	368			**courier**	**947**
		support	351			breast	843
		suite	**348**			**box**	**580**
		therapy	**347**			cancer	545
		md	**315**			**street**	**517**

Bold indicates term that does not appear in top 10 list for the other clusters

Table 16. wtlrposwnsyn, bnarized @ 25% RP, k=5

For k=7 it seems that the problems we experienced with k=5 have been amplified, particularly in the case of the WordNet-based representation. Based on the most frequent term set for the WordNet-based clusters as shown in Table 18 below, the WordNet-based model seems completely unreadable, at least in terms of selecting a single label.

cluster 1 freq	cluster 1 size (pct)	10 most frequent terms	freq	cluster 2 freq	cluster 2 size (pct)	10 most frequent terms	freq	cluster 3 freq	cluster 3 size (pct)	10 most frequent terms	freq
770	51%	**phone**	**1297**	162	11%	cancer	297	350	23%	_tim	648
		director	**1195**			patient	242			help	396
		mail	**1109**			site	208			**librari**	386
		courier	**953**			commun	185			hospit	383
		contact	753			resourc	172			patient	336
		box	635			educ	170			**hour**	316
		street	578			**public**	170			commun	313
		hospit	499			help	167			educ	301
		commun	496			hospit	162			support	289
		site	440			includ	160			**call**	261
cluster 4 freq	**cluster 4 size (pct)**	**10 most frequent terms**	**freq**	**cluster 5 freq**	**cluster 5 size (pct)**	**10 most frequent terms**	**freq**	**cluster 6 freq**	**cluster 6 size (pct)**	**10 most frequent terms**	**freq**
22	1%	**children**	336	88	6%	_tim	248	93	6%	breast	858
		assist	328			patient	218			cancer	753
		violenc	220			includ	194			support	698
		elig	219			commun	185			resourc	556
		adult	209			hospit	149			therapi	495
		child	206			**activ**	148			**treatment**	466
		commun	187			**nurs**	147			hospit	449
		_tim	181			educ	139			contact	381
		support	170			support	137			**surgeri**	308
		contact	168			**resid**	130			**cell**	287
cluster 7 freq	**cluster 7 size (pct)**	**10 most frequent terms**	**freq**								
13	1%	**medicin**	433								
		sport	431								
		therapi	412								
		product	403								
		cancer	385								
		suit	307								
		insur	298								
		abc	289								
		breast	281								
		_tim	277								

Table 17. wttlrstemmed binarized @ 25% RP, k=7

While it appears possible that the clusters for the wttlrstemmed-based representation will, when a sample of documents from its clusters are inspected manually, be helpful towards identifying a set of topics, it appears that will be

cluster 1 freq	cluster 1 size (pct)	10 most frequent terms	freq	cluster 2 freq	cluster 2 size (pct)	10 most frequent terms	freq	cluster 3 freq	cluster 3 size (pct)	10 most frequent terms	freq
652	44%	library	388	313	21%	contact	415	123	8%	program	246
		family	378			program	323			family	233
		program	318			hospital	294			hospital	174
		site	287			site	246			community	164
		font	279			community	240			education	145
		hospital	256			family	235			help	133
		community	237			cancer	219			cancer	131
		department	229			treatment	215			**call**	125
		medicine	217			**staff**	190			font	118
		contact	215			**board**	188			contact	116
cluster 4 freq	**cluster 4 size (pct)**	**10 most frequent terms**	**freq**	**cluster 5 freq**	**cluster 5 size (pct)**	**10 most frequent terms**	**freq**	**cluster 6 freq**	**cluster 6 size (pct)**	**10 most frequent terms**	**freq**
15	1%	service	349	96	6%	**yes**	1460	282	19%	font	365
		cancer	281			**phone**	1262			support	260
		contact	275			**director**	1202			family	252
		family	255			department	1107			**color**	242
		program	235			**mail**	1062			hospital	229
		assistance	212			**courier**	946			community	221
		violence	209			breast	778			program	200
		community	131			box	588			library	195
		social	129			street	518			cancer	189
		treatment	125			support	506			service	180
cluster 7 freq	**cluster 7 size (pct)**	**10 most frequent terms**	**freq**								
17	1%	cancer	504								
		breast	364								
		medicine	334								
		contact	309								
		suite	309								
		md	292								
		street	291								
		therapy	275								
		am	274								
		insurance	274								

Table 18. wtlrposwnsyn, binarized @ 25% RP, k=7

highly unlikely for the WordNet-based representation.

4.9 Evaluating clusters from the POS-reduction representation

Having become fully skeptical of the naïve WordNet representation, I wish to see if perhaps some advantage over the wttlrstemmed-based representation might be found in the intermediate representation wtlrposstemmed, the one reduced by part of speech. For present purposes, I will stick to k=5, namely since k=7 for wttlrstemmed appears less promising than k=5 clusters.

For the present analysis I wish to better elicit terms and stems that distinguish one cluster from another. In other words, I want some measure, much like TFIDF (Salton, 1971), that might represent the very terms/stems that separate one cluster from another. For that purpose, I have essentially rewritten TFIDF to work for clusters just as TFIDF works for documents, a measure I call TCFICF, which stands for *term cluster frequency, inverse cluster frequency*. As you might see from the definition below, it is exactly TFIDF but clusters are substituted where documents normally would go. *ln* was selected over log because of the small number of clusters; the rate of change of *ln* from 0 to 9 (our range for N and cf) is higher than the rate of change of log, and so *ln* seems intuitively better-suited for lower N.

4.9.1 A definition of TCFICF

For a term / stem / synset_id *i* in cluster *j*,

$$W_{i,j} = tcf_{i,j} \times \ln(N / cf_i)$$

$tcf_{i,j}$ = number of occurrences of *i* in *j*
cf_i = number of clusters containing *i*
N = total number of clusters

ln chosen because of the small number of clusters

4.9.2. Results

Tables 19 and 20 contain, side-by-side, both the top 10 most frequent terms lists and the top 10 highest scoring terms by TCFICF lists for each cluster, for wttlrstemmed (T19) and wtlrposstemmed (T20), respectively.

It seems that the TCFICF measure is more useful than expected yet what it is revealing seems to be a bit distressing. The big clusters show their most discriminating features, by way of TCFICF, to be terms like “font” and “courier”—in other words, HTML noise. With this noise we cannot be confident about the relevance of our features to the clustering task at hand and the context of information architecture in which it has been framed. Fortunately, it does appear that noise has rather limited itself to the one big cluster; it may be possible that in both cases the other four clusters are useful. Of course, it is nice

cluster 1 freq	cluster 1 size (pct)	10 most frequent terms	freq	10 highest ranked terms, tcficf	score	cluster 2 freq	cluster 2 size (pct)	10 most frequent terms	freq	10 highest ranked terms, tcficf	score
787	53%	**phone**	**1361**	courier	487.3	226	15%	**breast**	**982**	aap	123.7
		director	**1246**	font	94.2			cancer	939	adc	109.0
		mail	**1103**	serif	56.8			support	865	node	104.5
		courier	**954**	dmc	48.3			**resourc**	**738**	adh	88.5
		contact	782	sickl	35.7			therapi	579	sentinel	57.2
		box	**676**	dyer	35.4			**treatment**	**570**	mutat	43.5
		street	625	psychoanalyt	32.2			patient	503	font	42.8
		hospit	608	ccc	31.2			contact	486	vamc	38.3
		commun	502	finder	29.0			hospit	486	unknown	33.9
		site	**473**	slp	27.5			**surgeri**	**379**	dietitian	27.0
cluster 3 freq	**cluster 3 size (pct)**	**10 most frequent terms**	**freq**	**10 highest ranked terms, tcficf**	**score**	**cluster 4 freq**	**cluster 4 size (pct)**	**10 most frequent terms**	**freq**	**10 highest ranked terms, tcficf**	**score**
378	25%	_tim	703	font	43.7	33	2%	_tim	433	brand	82.2
		help	**426**	hpv	29.0			**product**	**402**	prosthes	58.5
		hospit	407	leagu	24.3			**children**	**387**	airwai	48.6
		librari	**382**	adc	20.2			**assist**	**385**	rectum	46.7
		patient	354	midwif	16.3			**insur**	**337**	fitter	35.9
		educ	**331**	slater	16.1			hour	284	bra	33.7
		commun	325	breastfe	13.8			cancer	274	glove	32.2
		hour	324	abp	12.9			**plan**	**269**	enrolle	30.6
		support	302	eff	12.8			**prosthes**	**262**	neglect	26.6
		call	**282**	bold	12.5			commun	258	wig	26.1
cluster 5 freq	**cluster 5 size (pct)**	**10 most frequent terms**	**freq**	**10 highest ranked terms, tcficf**	**score**						
74	5%	**medicin**	**499**	abc	146.6						
		sport	**442**	abcd	26.6						
		therapi	435	mpt	26.6						
		suit	**309**	labyrinth	22.5						
		abc	**287**	zen	22.5						
		physic	**285**	hei	17.7						
		cancer	245	lmp	17.7						
		street	241	dpm	14.7						
		clinic	**240**	greyhound	14.5						
		patient	239	bloch	12.9						

Table 19. wttlrstemmed binarized @ 25% RP, k=5 using TF and TCFICF

to know that TCFICF provides good post-learning feedback about our preprocessing performance, particularly important to processing html.

cluster 1 freq	cluster 1 size (pct)	10 most frequent terms	freq	10 highest ranked terms, tcficf	score	cluster 2 freq	cluster 2 size (pct)	10 most frequent terms	freq	10 highest ranked terms, tcficf	score
891	59%	hospit	724	font	127.19	179	12%	therapi	473	adh	88.52
		contact	586	text	60.25			**sport**	459	midg	24.14
		font	570	serif	56.81			**medicin**	444	emb	20.98
		site	556	decor	43.74			hospit	370	auditori	19.31
		patient	541	psychoanalyt	32.19			clinic	348	orthopaed	19.19
		commun	528	bold	31.02			**suit**	325	autism	17.37
		clinic	509	dialysi	29.63			physic	322	parenthood	15.58
		physician	474	harp	24.14			commun	321	midwif	15.32
		educ	459	helvetica	22.54			cancer	317	dpm	14.66
		help	397	donor	21.97			help	311	greyhound	14.48
cluster 3 freq	**cluster 3 size (pct)**	**10 most frequent terms**	**freq**	**10 highest ranked terms, tcficf**	**score**	**cluster 4 freq**	**cluster 4 size (pct)**	**10 most frequent terms**	**freq**	**10 highest ranked terms, tcficf**	**score**
344	23%	**phone**	1129	courier	211.54	67	4%	breast	712	sentinel	102.62
		director	1079	leagu	52.10			support	632	node	62.32
		mail	1054	dyer	35.41			**resourc**	591	rectum	46.67
		courier	948	font	30.57			**treatment**	477	colorect	33.90
		_tim	592	bass	20.92			therapi	473	dietitian	25.44
		box	465	text	18.30			**assist**	450	font	22.09
		street	461	breastfe	17.88			contact	436	endocrin	20.92
		librari	450	carmin	17.70			hospit	408	unknown	17.88
		hour	308	rust	17.70			cancer	398	exploit	14.81
		support	302	slater	16.09			**special**	386	nitrogen	14.48
cluster 5 freq	**cluster 5 size (pct)**	**10 most frequent terms**	**freq**	**10 highest ranked terms, tcficf**	**score**						
18	1%	cancer	718	brand	80.20						
		_tim	488	prosthes	61.36						
		breast	432	airwai	48.56						
		insur	414	mutat	43.45						
		product	414	sleev	42.91						
		contact	329	fitter	36.82						
		support	318	enrolle	30.58						
		prosthes	275	labyrinth	22.53						
		hour	251	zen	22.53						
		meet	236	hat	20.31						

Table 20. wtlrposstemmed binarized @ 25% RP, k=5 using TF and TCFICF

On the other hand, TCFICF does not seem to help us in picking labels with the current data. This may however be a good thing, as it appears so far that the clusters are not very good, at least not from the standpoint of useful labels for a menu navigation.

4.10 A qualitative inspection of a sample of documents from the clusters

To get a better idea of the quality of clusters we have obtained, I have opted to generate random samples of documents from each of the clusters. The aim of this evaluation is to get a more hands-on qualitative sense of what the clusters look like.

It is not unusual to see this sort of evaluation in document clustering exercises, yet it is customary to select documents closest to the centroid. Of course, if we were to use these clusters as high-level organization schemes for a large collection of web documents, users won't have any idea as to how close to the centroid the documents they desire are situated. It seems somehow more "fair" and "objective" to take a random sample of the clusters.

For this exercise, 5 documents were randomly selected for every cluster of the k=5 experiments for our three core representations. Random selection was performed using Oracle's sample function.

The cluster samples may be viewed here:

- wttlr k=5: http://www.unc.edu/~pod/kdd/clusters/01wttlr/
- wtlrpos, k=5: http://www.unc.edu/~pod/kdd/clusters/02wtlrpos/
- wtposwnsyn, k=5: http://www.unc.edu/~pod/kdd/clusters/03wtlrsyn/

What we are looking for in the qualitative assessment at this point is simply whether the samples support our earlier claims, namely, whether the clusters from wttlrstemmed and its analogue wtlrposstemmed for the loose description proffered earlier, and whether the WordNet clusters genuinely were not so very good. Recall our doubts about the large clusters given the appearance of "font" and the like, but also our curiosity about how the other clusters might look.

For wttlrstemmed, Cluster 1 documents as expected seem completely incoherent. Cluster 2 documents seem in line with the earlier assessment, in that three of the five randomly-selected documents were about "therapy," and a fourth was from the same web site as one of those three therapy pages. Cluster 3 seemed to be pages focused on public health and alternative health, including a health library page, again conforming to the earlier description of that cluster from term

frequency data. Four of the five documents in cluster 4 focused on "services" particularly hospital services, with the fifth page focusing on medical supplies & prostheses. The fifth cluster seemed less coherent, but the sample did at least contain two pages containing therapy information and two pages regarding exercise.

The assessment of the clustering by qualitative review of the documents for the wtlrposstmmed representation seems as encouraging as the results for wttlrstemmed. Ignoring cluster 1 (again, because of the HTML or CSS noise), we see cluster 2 conform well, with three public health pages. Two of the sample pages in cluster 3 were related to childbirth with a third on pediatrics. Two of the pages in the sample for cluster 4 regarded human services, and finally three of the pages in cluster 5 regarded cancer treatment and screening.

Surprisingly, despite appearances from looking at the term frequency-related cluster data alone, the qualitative assessment shows that the WordNet representation may have performed better than the other two representations. Two of the five clusters were very focused according to the sample, and another was as good as any of the other clusters form the other representations.

WordNet-based representation, k=5:

- Cluster 1: two pages in Spanish

–Cluster 2: four pages heavily info clearinghouse-oriented: two libraries & two fact sheets

–Cluster 3: four public health pages

–Cluster 4: three social services-related ages

–Cluster 5: incoherent

The WordNet representation seemed more coherent upon a qualitative assessment, leaving open the question of how to better quantitatively assess clusters.

5. Conclusions

5.1. Feature Reduction

The results of the first step of the study demonstrated that binarization of the data sets invariably prevented an otherwise inevitable overfitting. It remains to be seen whether this is an idiosyncrasy of the present data set or whether it is related to more general factors, such as size or heterogeneity (higher dimensionality) of the corpus.

While random projection did not have as dramatic an impact on avoiding overfitting, it did help quite a bit in doing so, as evidenced from the data. The utility of random projection and binarization in combination underscore the utility of grand-scale dimensionality reduction in text mining.

5.2. Balanced clusters and the competitive representations approach

The present “competing models” approach seems to have promise for selecting optimal feature representations. It might be performed programmatically & expanded to include other candidate feature representation. This of course only becomes practical if the text mining system becomes more integrated.

Using the FACTOR balance measure proved a useful measure for automatically calculating the relationship between a particular representation factor and its relationship towards overfitting. Its virtues rest with its simplicity and its reflection of the needs related to information architecture. This balance measure may make less sense in other domains and types of problems. In any case it should only be used to reject overfitting models rather than to establish “best” models.

5.3. TCFICF for preprocessing feedback

TCFICF does produce some valuable insight into clusters that TF cannot provide. Namely, the TCFICF measures elucidated the prevalence of font-specific information, ostensibly noise, noise that eluded other preprocessing validations. It however is not clear whether the measure provides any useful information at this time about identifying good labels for document clusters, namely since it is so sensitive to highly specific terms, rather than more general terms, the sort of

terms we might want to use for a small & broad navigation menu.

5.4. WordNet representation & qualitative vs. quantitative assessment

The naïve approach to using WordNet introduces noise due to ambiguity that we might easily be rid of by using more WordNet features. This is evidenced not only by the cluster-based term frequency data but also by the increase in dimensionality it demands. At the same time, when qualitatively assessed, the clusters that appeared most coherent were the WordNet. This may be due to sampling error, but even for it to be competitive, given the dimensionality explosion as a result of ambiguity, is a pleasant surprise and is encouraging for further development of its use.

The success of WordNet according to the qualitative assessment and its apparent failure according to quantitative measures seems to indicate that the author should have used the two evaluative approaches side-by-side rather than sequentially. The augmentative approach seems to be promising, particularly when considering the development of a tool information architects might use to pick good document groupings and labels for them.

5.5. Current problems & potential solutions

The present study indicates the dangers inherent within using a highly heterogeneous web page corpus. Such collections are unsurprisingly very difficult to parse. As such, this system needs some significant refinements on the front end before it might be used as a high-quality classification and clustering research tool. The TCFICF will certainly come in handy.

Another place for improvement in the current system is with NLP-type features. One improvement would be to move from identifying words to identifying true terms by identifying phrases, such as “breast cancer” rather than “breast” and “cancer.” Another would be to perform POS-tagging up front; yet another would be to use a “perfect” stemmer, such as the Prefix stemmer.

POS tagging is not the only way to optimize use of WordNet. WordNet’s most powerful feature is the hypernymy data contained therein, followed by the meronymy. Exploiting these features along with POS tagging could actually make WordNet a powerful feature reduction tool.

It might prove useful to expand the competitive games approach to more features such as vector normalization or principle component analysis, and especially to the use of other algorithms. Simple K-means was selected for practical reasons, yet there are other algorithms that appear better-suited to the

present corpus, particularly hierarchical clustering algorithms or algorithms that allow for topic overlap. To wit:

> Clustering is a subjective process [….] This subjectivity makes the process of clustering difficult. This is because a single algorithm or approach is not adequate to solve every clustering problem. (Jain, 315).

Further improvements to the present study might be enhanced by better evaluation, namely the incorporation of purity and entropy statistics as well as more user-related qualitative data.

Finally, it would be instructive to apply the present study model to another heterogeneous web collection, perhaps one with a different order of magnitude in size, or one with a different topical focus. I suspect feature reduction performance may be highly specific to corpus size, heterogeneity, and the specific topics. Different topics may not only use different words but, more importantly, have broader or narrower distribution of features.

5.6. Future Questions

Can optimization of feature selection be automated? Can we use this competitive model to automatically select feature reps? Or are we going to always get the same factor levels? What makes feature selection performance

vary? Answering such questions requires better implementation of a text mining system—better integration, end-to-end—so that the problem takes a reasonable time to solve.

A much deeper issue lurks, one that the present author tried to briefly scratch at, but admittedly with a great deal of unease. When it comes to clustering, which is in no trivial sense a creative, generative process, what *is* optimal, anyway? Can we know what a "good" cluster is before we create one, define it rigorously, *functionally*, without resorting to "purity" and other conventionally measures that seem unindicative in an information architecture context? As with the present study, is context necessary to sort of set the "right bias"?

6. References

Banerjee, Arindam, and Langford, John. An objective evaluation criterion for clustering, Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining, August 22-25, 2004, Seattle, WA, USA

Dhillon, I., and Modha, D. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 2001, vol. 42, no. 1, pp. 143-175.

Efron, M., Elsas, J., Marchionini, G. and Zhang, J.. Machine Learning for Information Architecture. *Proceedings of the ACM & IEEE Joint Conference on Digital Libraries (JCDL 2004)*, 2004.

Fayyad, U., Piatetsky-Shapiro , G., and Smyth,P. The KDD process for extracting useful knowledge from volumes of data, *Communications of the ACM*, 1996, vol. 39, no. 11.

Frank, E., and Witten, I.E. *Data Mining: Practical machine learning tools with Java implementations.* Morgan Kaufmann, San Francisco, 2000.

Jain, A.K.,Murty, M.N., and Flynn, P.J. Data clustering: a review. *ACM Computing Surveys*, Sept. 1999, vol. 31 no. 3, pp. 264-323.

Jardine, N., and van Rijsbergen, C.J.. The use of hierarchic clustering in information retrieval. *Information Storage & Retrieval*, 1971, Vol. 7, pp. 217-240.

MacQueen, J.B. "Some Methods for classification and Analysis of Multivariate Observations, *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability"*, Berkeley, University of California Press, 1967, vol. 1, pp. 281-297.

Miller, George A. The Magical Number Seven, Plus or Minus Two: Some Limits in our Capacity for Processing Information. *The Psychological Revie*w, 1956, vol. 63 pp. 81-97.

Salton, G. *The SMART Retrieval System*. Prentice Hall, Englewood Cliffs, NJ, 1971.

7. Appendices

Appendix 1. Initial attempt to identify minimum and maximum term frequency levels

```
This file contains queries for the process of picking parameters for cluster data for my
knowledge discovery project.
Two representations are being examined & compared:
       webterm joined on the SPECIALIST lexicon (webterm_in_lragr)
       wordnet synsets of the above results


trying to set parameters...
word length 2, 3 characters
term frquency min 3
term freq max 900
minimum number of terms per document (selecting a value in the range 5 to 25)
examine the list of words eliminated by the maximum frequency and comparing them manually
to the subjects covered in the corpus
that range was approximately 250-900

pragmatic consideration: select a value in that range so that we might retain on the
order of 1500 documents out of the original 2400 or so documents that ere successfully
downloaded
min length of 2 was selected b/c some 2-length terms were eliminated that seemed
meaningful in a medical domain (e.g., pH)
capitalization was also preserved (due to the frequency of geographic and person names,
and again cases like pH vs. ph)
as for minimum frequency, I looked at 3 and 5; i assume that terms that

I ran a series of queries using variations of combinations of these values


finally, theset of documents available that meet these contstraints for both
representations were used

number filter
-------------
select distinct term, count(term)
from webterm_in_lragr
where term like '%1%'
or term like '%2%'
or term like '%3%'
or term like '%4%'
or term like '%5%'
or term like '%6%'
or term like '%7%'
or term like '%8%'
or term like '%9%'
or term like '%0%'
group by term
order by count(term);









------------------------
set pagesize 0
column term format a30
select distinct term, count(term)
from webterm_in_lragr
where length(term) > 2
and term not like '%1%'
or term not like '%2%'
```

```
or term not like '%3%'
or term not like '%4%'
or term not like '%5%'
or term not like '%6%'
or term not like '%7%'
or term not like '%8%'
or term not like '%9%'
or term not like '%0%'
and lower(term) not in
       (select *
       from stopwords)
group by term
having count(term) > 2
and count(term) < 901
order by count(term) desc;


set pagesize 0
column term format a30
select PMID,SECTIONID,PARAID,SENTID,WORDID,TERM
from webterm_in_lragr
where term in
       (select distinct term
       from webterm_in_lragr
       where length(term) > 1
       and term not like '%1%'
       or term not like '%2%'
       or term not like '%3%'
       or term not like '%4%'
       or term not like '%5%'
       or term not like '%6%'
       or term not like '%7%'
       or term not like '%8%'
       or term not like '%9%'
       or term not like '%0%'
       and lower(term) not in
              (select *
              from stopwords)
       group by term
       having count(term) > (5-1)
       and count(term) < (300+1))
ORDER BY PMID,SECTIONID,PARAID,SENTID,WORDID;


BEFORE STEMMING
---------------
1.  on webterm_in_lragr
A.
setting for stop words, min string length = 3, minTermFreq=3, max term frequency = 900
number of tokens = 217614
number of distinct terms = 6885
number of documents involved = 1715
of those, number of documents with less than 5 terms = 150
of those, number of documents with less than 10 terms = 299 (1416) <--
of those, number of documents with less than 25 terms = 549


B.
setting for stop words, min string length = 3, minTermFreq=8, max term frequency = 250
number of tokens = 142941
number of distinct terms = 3741
number of documents involved = 1688
of those, number of documents with less than 5 terms = 196
of those, number of documents with less than 10 terms = 411
of those, number of documents with less than 25 terms = 651


C.
setting for stop words, min string length = 2, minTermFreq=4, max term frequency = 575
number of tokens = 194820
number of distinct terms = 5786
```

```
number of documents involved = 1715
of those, number of documents with less than 5 terms = 156
of those, number of documents with less than 10 terms = 319
of those, number of documents with less than 25 terms = 571


min string length = 2; some units have meaning like pH or Ca
min term frequency = 6
max term frequency = 425

D.
setting for stop words, min string length = 2, minTermFreq=6, max term frequency = 425
number of tokens = 176961
number of distinct terms = 4552
number of documents involved = 1710
of those, number of documents with less than 5 terms = 172
of those, number of documents with less than 10 terms = 339
of those, number of documents with less than 25 terms = 596

E.
setting for stop words, min string length = 2, minTermFreq=6, max term frequency = 425
number of tokens = 176961
number of distinct terms = 4552
number of documents involved = 1710
of those, number of documents with less than 5 terms = 172
of those, number of documents with less than 10 terms = 339


F.
FINALLY, TRY
min string length = 2;
minTermFreq = 5;
maxTermFreq = 300
NO STEMMING!
number of tokens = 160,361
number of distinct terms = 4999
number of documents involved = 1694
of those, number of documents with less than 8 terms = 304  (1390)



********************************************************************************
********************
For webterm_in_lragr....
       PARAMETERS: min string length = 2, minTermFreq = 5, maxTermFreq = 300; minimum
tokens per document = 6,
       non-numeric strings only, NOT STEMMED, stop words removed
number of tokens = 159,617
number of distinct terms = 4998
number of documents involved = 1441
********************************************************************************
********************


Relevant queries:

TOTAL TOKENS:
set pagesize 0
column term format a30
select PMID,SECTIONID,PARAID,SENTID,WORDID,TERM
from webterm_in_lragr
where term in
       (select distinct term
       from webterm_in_lragr
       where length(term) > 1
       and term not like '%1%'
       or term not like '%2%'
       or term not like '%3%'
       or term not like '%4%'
       or term not like '%5%'
       or term not like '%6%'
```

```
        or term not like '%7%'
        or term not like '%8%'
        or term not like '%9%'
        or term not like '%0%'
        and lower(term) not in
                (select *
                from stopwords)
        group by term
        having count(term) > (5-1)
        and count(term) < (300+1))
and PMID in
        (select distinct PMID
        from webterm_in_lragr
        where term in
                (select distinct term
                from webterm_in_lragr
                where length(term) > 1
                and term not like '%1%'
                or term not like '%2%'
                or term not like '%3%'
                or term not like '%4%'
                or term not like '%5%'
                or term not like '%6%'
                or term not like '%7%'
                or term not like '%8%'
                or term not like '%9%'
                or term not like '%0%'
                and lower(term) not in
                        (select *
                        from stopwords)
                group by term
                having count(term) > (5-1)
                and count(term) < (300+1))
        group by PMID
        having count(PMID) > 5)
ORDER BY PMID,SECTIONID,PARAID,SENTID,WORDID;



UNIQUE TERMS:
set pagesize 0
column term format a30
select count(distinct term)
from webterm_in_lragr
where term in
        (select distinct term
        from webterm_in_lragr
        where length(term) > 1
        and term not like '%1%'
        or term not like '%2%'
        or term not like '%3%'
        or term not like '%4%'
        or term not like '%5%'
        or term not like '%6%'
        or term not like '%7%'
        or term not like '%8%'
        or term not like '%9%'
        or term not like '%0%'
        and lower(term) not in
                (select *
                from stopwords)
        group by term
        having count(term) > (5-1)
        and count(term) < (300+1))
and PMID in
        (select distinct PMID
        from webterm_in_lragr
        where term in
                (select distinct term
                from webterm_in_lragr
                where length(term) > 1
```

```
                and term not like '%1%'
                or term not like '%2%'
                or term not like '%3%'
                or term not like '%4%'
                or term not like '%5%'
                or term not like '%6%'
                or term not like '%7%'
                or term not like '%8%'
                or term not like '%9%'
                or term not like '%0%'
                and lower(term) not in
                        (select *
                        from stopwords)
                group by term
                having count(term) > (5-1)
                and count(term) < (300+1))
        group by PMID
        having count(PMID) > 5);


TOTAL PAGES:
set pagesize 0
column term format a30
select count(distinct PMID)
from webterm_in_lragr
where term in
        (select distinct term
        from webterm_in_lragr
        where length(term) > 1
        and term not like '%1%'
        or term not like '%2%'
        or term not like '%3%'
        or term not like '%4%'
        or term not like '%5%'
        or term not like '%6%'
        or term not like '%7%'
        or term not like '%8%'
        or term not like '%9%'
        or term not like '%0%'
        and lower(term) not in
                (select *
                from stopwords)
        group by term
        having count(term) > (5-1)
        and count(term) < (300+1))
and PMID in
        (select distinct PMID
        from webterm_in_lragr
        where term in
                (select distinct term
                from webterm_in_lragr
                where length(term) > 1
                and term not like '%1%'
                or term not like '%2%'
                or term not like '%3%'
                or term not like '%4%'
                or term not like '%5%'
                or term not like '%6%'
                or term not like '%7%'
                or term not like '%8%'
                or term not like '%9%'
                or term not like '%0%'
                and lower(term) not in
                        (select *
                        from stopwords)
                group by term
                having count(term) > (5-1)
                and count(term) < (300+1))
        group by PMID
        having count(PMID) > 5);
```

```
********************************************************************************************
******************
********************************************************************************************
******************
2.  on wt_lr_wn:

match the equivalent of the results above for parameters....
first create a table view of webterm_in_lragr limited to parameters
call it wt_lr_cluster_view:

create table wt_lr_cluster_view as
       select PMID,SECTIONID,PARAID,SENTID,WORDID,TERM
       from webterm_in_lragr
       where term in
              (select distinct term
              from webterm_in_lragr
              where length(term) > 1
              and term not like '%1%'
              or term not like '%2%'
              or term not like '%3%'
              or term not like '%4%'
              or term not like '%5%'
              or term not like '%6%'
              or term not like '%7%'
              or term not like '%8%'
              or term not like '%9%'
              or term not like '%0%'
              and lower(term) not in
                     (select *
                     from stopwords)
              group by term
              having count(term) > (5-1)
              and count(term) < (300+1))
       and PMID in
              (select distinct PMID
              from webterm_in_lragr
              where term in
                     (select distinct term
                     from webterm_in_lragr
                     where length(term) > 1
                     and term not like '%1%'
                     or term not like '%2%'
                     or term not like '%3%'
                     or term not like '%4%'
                     or term not like '%5%'
                     or term not like '%6%'
                     or term not like '%7%'
                     or term not like '%8%'
                     or term not like '%9%'
                     or term not like '%0%'
                     and lower(term) not in
                            (select *
                            from stopwords)
                     group by term
                     having count(term) > (5-1)
                     and count(term) < (300+1))
              group by PMID
              having count(PMID) > 5)
       ORDER BY PMID,SECTIONID,PARAID,SENTID,WORDID;


restrict only to above documents & to min string length & non-numeric stop-cleared
strings, but apply minSynset_idFreq, maxSynset_idFreq, minimum tokens to synsets,

       PARAMETERS: min string length = 2, minSynset_idFreq = 5, maxSynset_IDFreq = 300;
       non-numeric strings only, NOT STEMMED, stop words removed


select *
from wt_lr_wn
where PMID in
```

```
        (select distinct PMID
        from wt_lr_cluster_view)
and str in
        (select distinct str
        from wt_lr_wn
        where length(str) > 1
        and str not like '%1%'
        or str not like '%2%'
        or str not like '%3%'
        or str not like '%4%'
        or str not like '%5%'
        or str not like '%6%'
        or str not like '%7%'
        or str not like '%8%'
        or str not like '%9%'
        or str not like '%0%'
        and lower(str) not in
                (select *
                from stopwords))
and synset_id in
        (select distinct synset_id
        from wt_lr_wn
        group by synset_id
        having count(synset_id) > (5-1)
        and count(synset_id) < (300+1)
        )
order by PMID,SECTIONID,PARAID,SENTID,WORDID,SYNSET_ID;


********************************************************************************************
***************
number of tokens as unique location-synset_id pairs = 575,899
number of distinct synset_ids = 12094
number of documents involved = 1441?  1432???
********************************************************************************************
***************



create table tmp00 as
select distinct PMID
from wt_lr_cluster_view;

create table tmp01 as
select distinct str
from wt_lr_wn
where length(str) > 1
and str not like '%1%'
or str not like '%2%'
or str not like '%3%'
or str not like '%4%'
or str not like '%5%'
or str not like '%6%'
or str not like '%7%'
or str not like '%8%'
or str not like '%9%'
or str not like '%0%'
and lower(str) not in
        (select *
        from stopwords);

create table tmp02 as
select distinct synset_id
from wt_lr_wn
group by synset_id
having count(synset_id) > (5-1)
and count(synset_id) < (300+1);

select distinct synset_id
from wt_lr_wn
where PMID in
```

```
        (select *
        from tmp00)
and str in
        (select *
        from tmp01)
and synset_id in
        (select *
        from tmp02);
```

the number of files in the two representations do not match, so I'm going to force the two representations to deal with the exact same set of papers

```
create table wt_lr_cluster_view_2 as
select *
from wt_lr_cluster_view
where PMID in
        (select distinct PMID
        from wt_lr_wn
        where PMID in
                (select *
                from tmp00)
        and str in
                (select *
                from tmp01)
        and synset_id in
                (select *
                from tmp02));

drop table wt_lr_cluster_view;
create table wt_lr_cluster_view as
select * from wt_lr_cluster_view_2;
```

Now I need to get the new wt_lr_cluster_view stats:

```
********************************************************************************************
***********
For webterm_in_lragr....
        PARAMETERS: min string length = 2, minTermFreq = 5, maxTermFreq = 300; minimum
tokens per document = 6,
        non-numeric strings only, NOT STEMMED, stop words removed
number of tokens = 159,617
number of distinct terms = 4998
number of documents involved = 1441

But not all of these documents could be included by the synset representation...

For webterm_in_lragr
        PARAMETERS: min string length = 2, minTermFreq = 5, maxTermFreq = 300; minimum
tokens per document = 6,
        non-numeric strings only, NOT STEMMED, stop words removed
all captured in table wt_lr_cluster_view
number of tokens = 159,525
number of distinct terms = 4998
number of documents involved = 1432

********************************************************************************************
***********
number of tokens as unique location-synset_id pairs = 575,899
number of distinct synset_ids = 12094
number of distinct terms = 4113
number of documents involved = 1432

restricted only to documents included in the above restricted representation of
webterm_in_lragr, also to min string length & non-numeric stop-cleared strings, but
minSynset_idFreq, maxSynset_idFreq.  Minimum synsets was not tested.
```

```
        PARAMETERS: min string length = 2, minSynset_idFreq = 5, maxSynset_IDFreq = 300;
        non-numeric strings only, NOT STEMMED, stop words removed


Captured in table wt_lr_synset_cluster_view
********************************************************************************
**********


This "final" representation led to the problem of "monster" clusters for all values
except k=2 for simple K-means.

So I need to tweak my factors
de-capitalize
increase minimum tokens per document from 6 to
minTermFreq from 5 to
maxTermFreq from 300 to
min string length from 2 to 3
frequency to tfidf

phrasing:
1. join tables SPECIALIST and wordnet
search for words joined by '_'

select w.word
from wn_synset w, lragr s
where w.word in
 (select word from wn_synset
 where word like '%\_%')
and lower(w.word)=lower(s.str);


select word from wn_synset
where word like '%\_%'
ESCAPE '\';


select REGEXP_REPLACE(str, 'a', 'b')
from lragr
where str like '% %'



ESCAPE '\';

create table wt_lr_synset_cluster_view as
select distinct *
from wt_lr_wn
where PMID in
        (select *
        from tmp00)
and str in
        (select *
        from tmp01)
and synset_id in
        (select *
        from tmp02);


//find the number of documents with less than 5, 10, 25, 50 terms
Set pagesize 0
column term format a30
select distinct PMID, count(PMID)
from webterm_in_lragr
where term in
        (select distinct term
        from webterm_in_lragr
        where length(term) > 1
        and term not like '%1%'
        or term not like '%2%'
        or term not like '%3%'
```

```
        or term not like '%4%'
        or term not like '%5%'
        or term not like '%6%'
        or term not like '%7%'
        or term not like '%8%'
        or term not like '%9%'
        or term not like '%0%'
        and lower(term) not in
               (select *
               from stopwords)
        group by term
        having count(term) > (5-1)
        and count(term) < (300+1))
group by PMID
having count(PMID) < 8;

ORDER BY SECTIONID,PARAID,SENTID,WORDID;
```

Appendix 2. Second/final attempt to define minimum and maximum term frequencies

```
examining the following representations
baseline:
webterm
webtermtwo
webtermtwostemmed

features:
webtermtwolragrstemmed
wtlrposstemmed
wtlrposwnsyn


picking parameters again

1. selecting minimum number of terms per document: 10
I selected 10 because html pages (present data set included) may frequently have very
little content.

In fact I made sure that the set of documents chosen would be consistent across all
experiments, and so the documents must have a mimimum of 10 terms/stems; the
wtlragrposwnsyn table has 1499 documents with a minimum of 10 terms; this is the set of
documents selected

2.  minimum term frequency:  2, 5
terms that occur once don't provide any information that might cluster, but what about 3,
or 5 or 10?  let's see what happens when we lose approximately 50% ot 60% of the
terms/stems.  Out of the six tables to be evaluated, it looks like a minimum term
frequency of 4 puts us there


3.  maximum term frequency: 1950
set by querying each table for terms/stems happening over 500x; apparently non-trivial
terms that accord well with potential groupings should not be cut out (e.g., breast,
cancer), but trivial terms should (e.g., health, home, center)


set pagesize 0
column term format a30
column stem format a30

select term, frequency from (select distinct term, count(term) as frequency from webterm
group by term) where frequency>500 order by frequency
suggests a cutoff of 1100
select term, frequency from (select distinct term, count(term) as frequency from
webtermtwo group by term) where frequency>500 order by frequency
SUGGESTS A CUTOFF OF 1900
select stem, frequency from (select distinct stem, count(stem) as frequency from
webtermtwostemmed group by stem) where frequency>500 order by frequency
suggests a cutoff of 1950
select stem, frequency from (select distinct stem, count(stem) as frequency from
webtermtwolragrstemmed group by stem) where frequency>500 order by frequency
suggests a cutoff of 1950 again
select stem, frequency from (select distinct stem, count(stem) as frequency from
wtlrposstemmed group by stem) where frequency>500 order by frequency
1950


tables for running weka experiments on:

create table webterm_min2_max1950 as
select * from webterm
where term in
(select term
from (select distinct term, count(term) as frequency from webterm group by term)
where frequency>1 and
frequency<1951)
```

```
and pmid in (select pmid from kddnchipmidlist);

create table webterm_min5_max1950 as
select * from webterm
where term in
(select term
from (select distinct term, count(term) as frequency from webterm group by term)
where frequency>4 and
frequency<1951)
and pmid in (select pmid from kddnchipmidlist);

create table webtermtwo_min2_max1950 as
select * from webtermtwo
where term in
(select term
from (select distinct term, count(term) as frequency from webtermtwo group by term)
where frequency>1 and
frequency<1951)
and pmid in (select pmid from kddnchipmidlist);

create table webtermtwo_min5_max1950 as
select * from webtermtwo
where term in
(select stem
from (select distinct term, count(term) as frequency from webtermtwo group by term)
where frequency>4 and
frequency<1951)
and pmid in (select pmid from kddnchipmidlist);

create table webtermtwostemmed_min2_max1950 as
select * from webtermtwostemmed
where stem in
(select stem
from (select distinct stem, count(stem) as frequency from webtermtwostemmed group by
stem)
where frequency>1 and
frequency<1951)
and pmid in (select pmid from kddnchipmidlist);

create table webtermtwostemmed_min5_max1950 as
select * from webtermtwostemmed
where stem in
(select stem
from (select distinct stem, count(stem) as frequency from webtermtwostemmed group by
stem)
where frequency>4 and
frequency<1951)
and pmid in (select pmid from kddnchipmidlist);

create table webtermtwolragrstemmed_min2_max1950 as
select * from webtermtwolragrstemmed
where stem in
(select stem
from (select distinct stem, count(stem) as frequency from webtermtwolragrstemmed group by
stem)
where frequency>1 and
frequency<1951)
and pmid in (select pmid from kddnchipmidlist);

create table webtermtwolragrstemmed_min5_max1950 as
select * from webtermtwolragrstemmed
where stem in
(select stem
from (select distinct stem, count(stem) as frequency from webtermtwolragrstemmed group by
stem)
where frequency>4 and
frequency<1951)
and pmid in (select pmid from kddnchipmidlist);

create table wtlrposstemmed_min2_max1950 as
select * from wtlrposstemmed
```

```
where stem in
(select stem
from (select distinct stem, count(stem) as frequency from wtlrposstemmed group by stem)
where frequency>1 and
frequency<1951)
and pmid in (select pmid from kddnchipmidlist);

create table wtlrposstemmed_min5_max1950 as
select * from wtlrposstemmed
where stem in
(select stem
from (select distinct stem, count(stem) as frequency from wtlrposstemmed group by stem)
where frequency>4 and
frequency<1951)
and pmid in (select pmid from kddnchipmidlist);


create table wtlrposwnsyn_min2_max1950 as
select * from wtlrposwnsyn
where synset_ID in
(select synset_ID
from (select distinct synset_ID, count(synset_ID) as frequency from wtlrposwnsyn group by
synset_ID)
where frequency>1 and
frequency<1951)
and pmid in (select pmid from kddnchipmidlist);

create table wtlrposwnsyn_min5_max1950 as
select * from wtlrposwnsyn
where synset_ID in
(select synset_ID
from (select distinct synset_ID, count(synset_ID) as frequency from wtlrposwnsyn group by
synset_ID)
where frequency>4 and
frequency<1951)
and pmid in (select pmid from kddnchipmidlist);



webterm_min2_max1950
webterm_min5_max1950
webtermtwo_min2_max1950
webtermtwo_min5_max1950
webtermtwostemmed_min2_max1950
webtermtwostemmed_min5_max1950
webtermtwolragrstemmed_min2_max1950
webtermtwolragrstemmed_min5_max1950
wtlrposstemmed_min2_max1950
wtlrposstemmed_min5_max1950
wtlrposwnsyn_min2_max1950
wtlrposwnsyn_min5_max1950

document list:

where PMID in kddnchipmidlist

create table kddnchipmidlist as
select distinct pmid from
(select distinct * from
(select pmid, type, sectionid, paraid, supplid, sentid, wordid, term
from wtlrposwnsyn))
group by pmid having count(pmid) > 10;



webterm
webtermtwo
webtermtwostemmed
webtermtwolragrstemmed
wtlrposstemmed
```

wtlrposwnsyn

4. k value

5, 6, 7

it may be the case that I might only get two

Appendix 3: Complete distinct term count statistics on preliminary feature representations

WEBTERM

before stemming & stoplist

table name: webterm

number of unique terms	number of documents	no of terms w/term frequency=1	no of terms w/term frequency=2	no of terms w/term frequency=3	no of terms w/term frequency=4	no of terms w/term frequency=5	no of terms w/term frequency <10	no of terms w/term frequency <20	no of terms w/term frequency <30	no of terms w/term frequency <50	no of terms w/term frequency <100	no of terms w/term frequency > 250	no of terms w/term frequency > 500	no of terms w/term frequency > 750	no of terms w/term frequency > 1000	most frequent term
45078	1854	21558	6670	3213	1948	1319	37563	40743	42049	43147	44115	359	159	102	76	SYM_comma (36689)
100%	100%	48%	15%	7%	4%	3%	83%	90%	93%	96%	98%	1%	0%	0%	0%	

after lower cased, stoplisted, & stripped of strings containing non-alpha characters

table name: webtermtwo

number of unique terms	number of documents	number of documents w/ >10 terms	no of terms w/term frequency=1	no of terms w/term frequency=2	no of terms w/term frequency=3	no of terms w/term frequency=4	no of terms w/term frequency=5	no of terms w/term frequency <10	no of terms w/term frequency <20	no of terms w/term frequency <30	no of terms w/term frequency <50	no of terms w/term frequency <100	no of terms w/term frequency > 250	no of terms w/term frequency > 500	no of terms w/term frequency > 750	no of terms w/term frequency > 1000	most frequent term
24888	1822	1614	11032	3540	1718	1172	831	20112	21996	22779	23534	24203	235	95	52	36	health (6569)
55%	98%	87%	44%	14%	7%	5%	3%	81%	88%	92%	95%	97%	1%	0%	0%	0%	

after stemming

table name: webtermtwostemmed

number of unique stems/terms	number of documents	number of documents w/ >10 stems	no of stems w/stem frequency=1	no of stems w/stem frequency=2	no of stems w/stem frequency=3	no of stems w/stem frequency=4	no of stems w/stem frequency=5	no of stems w/stem frequency <10	no of stems w/stem frequency <20	no of stems w/stem frequency <30	no of stems w/stem frequency <50	no of stems w/stem frequency <100	no of stems w/stem frequency > 250	no of stems w/stem frequency > 500	no of stems w/stem frequency > 750	no of stems w/stem frequency > 1000	most frequent stem
19213	1822	1614	8753	2669	1257	851	578	15417	16738	17316	17896	18490	278	120	72	44	health (6577)
43%	98%	87%	46%	14%	7%	4%	3%	80%	87%	90%	93%	96%	1%	1%	0%	0%	

number of unique terms not stemmed: 584

WTLRAGR

before stemming & stoplist

table name: webtermtwolragr

number of unique terms	number of documents	number of documents w/ >10 terms	no of terms w/term frequency=1	no of terms w/term frequency=2	no of terms w/term frequency=3	no of terms w/term frequency=4	no of terms w/term frequency=5	no of terms w/term frequency <10	no of terms w/term frequency <20	no of terms w/term frequency <30	no of terms w/term frequency <50	no of terms w/term frequency <100	no of terms w/term frequency > 250	no of terms w/term frequency > 500	no of terms w/term frequency > 750	no of terms w/term frequency > 1000	most frequent term
14392	1820	1604	4394	1984	1115	816	611	10322	11809	12496	13149	13745	221	92	51	35	health (6569)
100% (58% of webtermtwo)	100%	87%	31%	14%	8%	6%	4%	72%	82%	87%	91%	96%	2%	1%	0%	0%	

after stemming

table name: webtermtwolragrstemmed

number of unique stems	number of documents	number of documents w/ >10 stems	no of stems w/stem frequency=1	no of stems w/stem frequency=2	no of stems w/stem frequency=3	no of stems w/stem frequency=4	no of stems w/stem frequency=5	no of stems w/stem frequency <10	no of stems w/stem frequency <20	no of stems w/stem frequency <30	no of stems w/stem frequency <50	no of stems w/stem frequency <100	no of stems w/stem frequency > 250	no of stems w/stem frequency > 500	no of stems w/stem frequency > 750	no of stems w/stem frequency > 1000	most frequent stem
9241	1820	1604	2547	1186	665	502	364	6164	7108	7564	8041	8558	263	117	71	43	health (6576)
100%	100%	88%	28%	13%	7%	5%	4%	67%	77%	82%	87%	93%	3%	1%	1%	0%	

The join on lragr sheds 10496 terms. Almost all of the lost terms are either 1) proper names(~ 60%);
2) obscure organizational acronyms (~10%); 3)spanish words (~10%), 4)misspelled words (~15%), or
5) odd words with non-alpha characters inside (typically two terms separated by ellipses or typos ~3-5%);
try SQL>select * from lostermtmp sample(0.3); to evaluate

WTLRAGRPOS

before stemming & stoplist

table name: wtlrpos

number of unique terms	number of documents	number of documents w/ >10 terms	no of terms w/term frequency=1	no of terms w/term frequency=2	no of terms w/term frequency=3	no of terms w/term frequency=4	no of terms w/term frequency=5	no of terms w/term frequency <10	no of terms w/term frequency <20	no of terms w/term frequency <30	no of terms w/term frequency <50	no of terms w/term frequency <100	no of terms w/term frequency > 250	no of terms w/term frequency > 500	no of terms w/term frequency > 750	no of terms w/term frequency > 1000	most frequent term
12707	1792	1593	3912	1733	972	716	536	9098	10406	11007	11594	12116	206	86	49	33	health (6569)
100% (88% of webtermt wolragr)	100%	89%	31%	14%	8%	6%	4%	72%	82%	87%	91%	95%	2%	1%	0%	0%	

after stemming

table name: wtlrposstemmed

number of unique stems	number of documents	number of documents w/ >10 stems	no of stems w/stem frequency=1	no of stems w/stem frequency=2	no of stems w/stem frequency=3	no of stems w/stem frequency=4	no of stems w/stem frequency=5	no of stems w/stem frequency <10	no of stems w/stem frequency <20	no of stems w/stem frequency <30	no of stems w/stem frequency <50	no of stems w/stem frequency <100	no of stems w/stem frequency > 250	no of stems w/stem frequency > 500	no of stems w/stem frequency > 750	no of stems w/stem frequency > 1000	most frequent stem
7723	1792	1593	2129	964	543	412	295	5086	5868	6241	6655	7096	249	111	69	41	health (6576)
100% (61% of wtlrpos)	100%	89%	28%	12%	7%	5%	4%	66%	76%	81%	86%	92%	3%	1%	1%	1%	

WTLRPOSWNSYN

table name: wtlrposwnsyn

number of unique terms	number of documents	number of documents w/ >10 terms	no of terms w/term frequency=1	no of terms w/term frequency=2	no of terms w/term frequency=3	no of terms w/term frequency=4	no of terms w/term frequency=5	no of terms w/term frequency <10	no of terms w/term frequency <20	no of terms w/term frequency <30	no of terms w/term frequency <50	no of terms w/term frequency <100	no of terms w/term frequency > 250	no of terms w/term frequency > 500	no of terms w/term frequency > 750	no of terms w/term frequency > 1000	most frequent term
8796	1792	1499	2516	1120	649	495	363	6026	6971	7427	7892	8323	174	78	45	31	health (6569)
	100%	84%	29%	13%	7%	6%	4%	69%	79%	84%	90%	95%	2%	1%	1%	0%	

number of unique IDs	number of documents	number of documents w/ >10 IDs	no of IDs w/ID frequency=1	no of IDs w/ID frequency=2	no of IDs w/ID frequency=3	no of IDs w/ID frequency=4	no of IDs w/ID frequency=5	no of IDs w/ID frequency <10	no of IDs w/ID frequency <20	no of IDs w/ID frequency <30	no of IDs w/ID frequency <50	no of IDs w/ID frequency <100	no of IDs w/ID frequency > 250	no of IDs w/ID frequency > 500	no of IDs w/ID frequency > 750	no of IDs w/ID frequency > 1000	most frequent ID
25634	1792	1737	4658	2483	1513	1303	921	13440	16580	18360	20404	22698	1206	553	337	232	113628836 (as terms 'health' and 'wellness' occurs 6799 times)
	100%	97%	18%	10%	6%	5%	4%	52%	65%	72%	80%	89%	5%	2%	1%	1%	

Appendix 4. 90 Initial clustering experiments on 30 candidate representations

BINARIZED

k5

wttlr

	none	rp3pct	rp25pct	rp50	rp150		
1	1230	864	788	875	869		
2	218	471	378	441	474		
3	30	130	226	151	127		
4	20	23	74	26	24		
5	1	11	33	6	5		
stddev	527.33405	366.226296	304.98721	365.4513648	369.8536197	386.771	
	78.93%	54.82%	45.65%	54.70%	55.36%	57.89%	

wtlrpos

	none	rp3pct	rp25pct	rp50	rp150		
1	1240	855	891	847	850		
2	207	412	344	365	408		
3	30	186	179	175	184		
4	21	36	67	99	50		
5	1	10	18	13	7		
stddev	532.069262	349.056156	353.43698	332.3510193	344.9524605	382.373	
FACTO	79.64%	52.25%	52.90%	49.75%	51.63%	57.23%	

wnsyn

	none	rp3pct	rp25pct	rp50	rp150		
1	1026	838	845	795	782		
2	250	450	380	386	431		
3	205	158	148	207	160		
4	16	44	98	91	100		
5	1	8	27	19	25		
stddev	420.908898	347.457623	332.3632	309.7221335	310.1246524	344.115	
	63.00%	52.01%	49.75%	46.36%	46.42%	51.51%	371.08633
							55.54%
all k5	493.437403	354.246692	330.26247	335.8415058	341.6435775		
	73.86%	53.02%	49.43%	50.27%	51.14%		
rppct vs fixed	493.437403	342.254579		338.7425417			
	73.86%	51.23%		50.70%			
none vs rp)	493.437403	340.49856					
	73.86%	50.97%					

k7

wttlr

	none	rp3pct	rp25pct	rp50	rp150	
1	1	802	771	828	818	
2	43	149	162	141	98	
3	205	354	350	362	402	
4	19	5	22	22	10	
5	1205	29	88	5	23	
6	16	157	93	135	144	
7	10	3	13	6	4	
stddev	442.646747	287.299164	270.56449	298.4065619	300.1979506	319.823
	78.50%	50.95%	47.98%	52.92%	53.24%	56.72%

wtlrpos

	none	rp3pct	rp25pct	rp50	rp150	
1	1	755	701	819	752	
2	70	168	148	159	157	
3	187	387	317	349	373	
4	22	147	57	124	63	
5	1194	1	22	13	27	
6	16	39	161	7	25	
7	9	2	93	28	2	
stddev	436.907095	273.907788	234.91022	292.4667665	275.3812076	302.715
	77.48%	48.57%	41.66%	51.87%	48.84%	53.68%

wnsyn

	none	rp3pct	rp25pct	rp50	rp150		
1	953	841	652	774	758		
2	1	453	313	353	431		
3	226	62	123	93	164		
4	9	5	15	3	9		
5	151	116	96	82	103		
6	1	3	282	168	8		
7	157	18	17	25	25		
stddev	338.192746	318.741274	226.15776	273.0518876	282.2611084	287.681	
	59.97%	56.52%	40.11%	48.42%	50.06%	51.02%	303.40618
							53.80%
k7	405.915529	293.316075	243.87749	287.975072	285.9467555		
	71.98%	52.02%	43.25%	51.07%	50.71%		

rppct vs fixed	405.915529	268.596782	286.9609138	
	71.98%	47.63%	50.89%	
none vs rp)	405.915529	277.778848		
	71.98%	49.26%		

k9

wttlr

	none	rp3pct	rp25pct	rp50	rp150	
1	1	820	759	761	816	
2	29	109	125	98	96	
3	213	394	304	360	402	
4	20	3	8	20	10	
5	1156	27	79	5	23	
6	10	138	107	118	145	
7	9	3	3	6	4	
8	36	2	31	46	2	
9	25	3	83	85	1	
stddev	376.726396	276.055751	239.80936	248.0887498	275.8360524	283.303
	75.85%	55.58%	48.28%	49.95%	55.53%	57.04%

wtlrpos

	none	rp3pct	rp25pct	rp50	rp150	
1	1	738	706	748	735	
2	55	165	143	154	150	
3	221	325	251	305	322	
4	22	67	67	93	64	
5	1127	1	15	6	31	
6	14	34	141	4	23	
7	9	2	120	28	2	
8	25	165	23	151	169	
9	25	2	33	10	3	
stddev	366.459449	239.975577	215.87966	239.5214558	237.4084619	259.849
	73.78%	48.31%	43.46%	48.22%	47.80%	52.32%

wnsyn

	none	rp3pct	rp25pct	rp50	rp150			
1	954	835	620	735	751			
2	1	440	333	419	432			
3	227	61	108	165	169			
4	7	13	19	26	9			
5	152	101	85	95	95			
6	1	3	250	1	6			
7	151	35	9	19	25			
8	2	6	9	12	6			
9	3	4	65	26	5			
stddev	307.871772	286.545856	203.74378	251.1374679	259.7306832	261.806		
	61.98%	57.69%	41.02%	50.56%	52.29%	52.71%	268.31936	
							54.02%	314.271
								54.54%
								3%
k9	350.352539	267.525728	219.81093	246.2492245	257.6583991			
	70.54%	53.86%	44.25%	49.58%	51.87%			

rppct vs fixed	350.352539	243.66833		251.9538118
	70.54%	49.06%		50.73%

none vs rp)	350.352539	247.811071
	70.54%	49.89%

Summary statistics, BINARIZED, k=5,7,9

k5-9	none	rp3pct	rp25pct	rp50	rp150
SD	416.5685	305.0295	264.6503	290.02193	295.08291
%MAX	72.13%	52.97%	45.65%	50.30%	51.24%
+/-	8.08%	3.38%	4.40%	2.56%	3.22%

	none	rppct
rppct vs fixed	416.568491	284.839897
	72.13%	49.31%

	rpC
	292.5524224
	50.77%

none vs rp)	none	rp
SD	416.568491	288.69616
%MAX	72.13%	50.04%
+/-	8.08%	4.36%

	wttlr	wtlrpos	wtlrposwn
	329.97	314.98	297.87
	57.21%	54.41%	51.74%

k5	k7	k9
371.09	303.41	268.32
0.56	0.54	0.54

TF

k5

wttlr

	none	rp3pct	rp25pct	rp50	rp150		1309.821
1	1	1442	1446	1444	1482		
2	2	1	7	9	5		
3	1470	42	32	40	9		
4	25	8	6	1	1		
5	1	6	8	5	2		
stddev	654.24208	638.715273	640.837109	639.81302	660.877	646.897	
	97.93%	95.60%	95.92%	95.77%	98.92%	96.83%	

wtlrpos

	none	rp3pct	rp25pct	rp50	rp150	
1	1	1478	1413	1482	1448	
2	2	1	1	4	1	
3	1470	9	78	2	37	
4	25	6	4	9	7	
5	1	5	3	2	6	
stddev	654.24208	658.6400383	623.153031	660.87609	642.02	647.786
FACTOR	97.93%	98.58%	93.27%	98.92%	96.10%	96.96%

wnsyn

	none	rp3pct	rp25pct	rp50	rp150		
1	1479	1414	1427	1453	1362		
2	1	64	10	8	98		
3	13	13	55	32	31		
4	4	3	4	3	5		
5	1	4	2	2	2		
stddev	659.32299	623.4751799	630.609467	644.88549	595.155	630.69	
	98.69%	93.32%	94.39%	96.53%	89.08%	94.40%	641.7909
							96.06%
all k5	655.93572	640.2768304	631.533202	648.52487	632.684		
	98.18%	95.84%	94.53%	97.07%	94.70%		
rppct vs fixed	655.93572	635.9050163		640.60445			
	98.18%	95.18%		95.88%			
none vs rp)	655.93572	638.2547321					
	98.18%	95.53%					

k7

wttlr

	none	rp3pct	rp25pct	rp50	rp150	
1	1	1180	1138	1226	1435	
2	2	1	40	7	5	
3	1473	274	277	227	42	
4	18	1	6	1	7	
5	1	6	8	5	2	
6	1	7	8	4	1	
7	3	30	22	29	7	
stddev	555.13795	437.3143143	418.93135	453.60351	538.535	**480.704**
	98.45%	77.55%	74.29%	80.44%	95.50%	**85.25%**

wtlrpos

	none	rp3pct	rp25pct	rp50	rp150	
1	1	1440	1414	1441	1432	
2	2	41	1	33	11	
3	1456	8	65	8	41	
4	35	3	4	11	8	
5	1	3	2	1	3	
6	1	3	5	2	3	
7	3	1	8	3	1	
stddev	547.74946	540.7360504	529.580157	541.10517	537.201	**539.274**
	97.14%	95.89%	93.91%	95.96%	95.27%	**95.63%**

wnsyn

	none	rp3pct	rp25pct	rp50	rp150		
1	1484	1459	1341	1367	1435		
2	1	19	106	90	5		
3	6	5	1	28	42		
4	4	3	4	2	7		
5	1	3	35	1	2		
6	1	8	9	8	1		
7	1	1	2	2	7		
stddev	560.02083	549.0261075	498.376029	509.42353	538.535	**531.076**	
	99.31%	97.36%	88.38%	90.34%	95.50%	**94.18%**	**517.0183**
							91.69%
k7	**554.30275**	**509.0254907**	**482.295846**	**501.3774**	**538.09**		
	98.30%	**90.27%**	**85.53%**	**88.91%**	**95.42%**		

rppct vs fixed	**554.30275**	**495.6606682**	**519.73371**	
	98.30%	**87.90%**	**92.17%**	
none vs rp)	**554.30275**	**507.6971876**		
	98.30%	**90.03%**		

k9

wttlr

	none	rp3pct	rp25pct	rp50	rp150	
1	1	1169	1221	1168	1279	
2	1	4	4	19	1	
3	323	274	231	275	185	
4	3	1	2	1	5	
5	1	4	6	5	2	
6	1	7	9	3	1	
7	4	32	21	21	15	
8	1128	3	4	1	4	
9	37	5	1	6	7	
stddev	375.5736	386.184901	402.357152	385.85234	421.414	394.276
	75.61%	77.75%	81.01%	77.68%	84.84%	79.38%

wtlrpos

	none	rp3pct	rp25pct	rp50	rp150	
1	1	1438	1409	1207	1201	
2	1	3	8	5	11	
3	318	42	66	238	250	
4	5	3	4	14	7	
5	1	2	2	1	4	
6	1	1	6	4	2	
7	4	1	2	3	1	
8	1133	3	1	2	1	
9	35	6	1	25	22	
stddev	376.8979	476.9709402	466.381848	397.60254	396.218	422.814
	75.88%	96.03%	93.90%	80.05%	79.77%	85.12%

wnsyn

	none	rp3pct	rp25pct	rp50	rp150			
1	1482	1343	1176	1256	1389			
2	1	95	98	30	73			
3	6	23	130	130	18			
4	4	18	4	5	1			
5	1	6	34	6	6			
6	1	3	18	35	3			
7	1	5	26	26	4			
8	1	3	2	4	2			
9	1	2	10	6	2			
stddev	493.33663	442.1928626	381.186015	410.48998	459.036	437.248		
	99.32%	89.03%	76.74%	82.64%	92.42%	88.03%	418.113	
							84.18%	525.641
								91.22%
								6%
k9	415.26938	435.1162346	416.641671	397.98162	425.556			
	83.61%	87.60%	83.88%	80.13%	85.68%			

rppct vs fixed	415.26938	425.878953	411.76883	
	83.61%	85.74%	82.90%	
none vs rp)	415.26938	418.8238914		
	83.61%	84.32%		

Summary statistics, TF, k=5,7,9

	none	rp3pct	rp25pct	rp50	rp150
k5-9	541.83595	528.1395186	510.156906	515.9613	532.11
	93.36%	91.24%	87.98%	88.70%	91.93%

	none	rppct	rpC
rppct vs fixed	541.83595	519.1482125	524.03566
	93.36%	89.61%	90.32%

	none	rp
none vs rp)	541.83595	521.591937
	93.36%	89.96%

wttlr	wtlrpos	wtlrposwn
507.29	536.62	533.00
87.15%	92.57%	92.20%

k5	k7	k9
641.79	517.02	418.11
0.96	0.92	0.84

SUMMARY STATISTICS, OVERALL BINARIZED VS TF

Overall

		bin	tf
binarized	*SD*	314.270626	525.641
vs tf	*%MAX*	51.54%	91%
	+/-	+/- 3%	+/- 6%

	none	rp
none vs rp	479.2022186	405.14405
	82.74%	70.00%

	rp pct	rp C
rp pct vs rp c	401.9940549	408.29404
	69.46%	70.54%

	rp 3pct	rp 25 pct	rp50	rp150
rp 3pct vs 25pct	416.5845085	387.4036	402.9916	413.6
rp 50 vs rp 150	72%	67%	70%	72%

		k5	k7	k9
k5, k7, k9		506.4386289	410.212	343.22
		76%	73%	69%

	wtt	wtlrpos	wtlrposwn
wtt, wtpos, wtwn	418.63	425.80	415.44
	72%	73%	72%