N-Gram-Based Text Categorization

William B. Cavnar and John M. Trenkle
Environmental Research Institute of Michigan
P.O. Box 134001
Ann Arbor MI 48113-4001

Abstract

Text categorization is a fundamental task in document processing, allowing the automated handling of enormous streams of documents in electronic form. One difficulty in handling some classes of documents is the presence of different kinds of textual errors, such as spelling and grammatical errors in email, and character recognition errors in documents that come through OCR. Text categorization must work reliably on all input, and thus must tolerate some level of these kinds of problems.

We describe here an N-gram-based approach to text categorization that is tolerant of textual errors. The system is small, fast and robust. This system worked very well for language classification, achieving in one test a 99.8% correct classification rate on Usenet newsgroup articles written in different languages. The system also worked reasonably well for classifying articles from a number of different computer-oriented newsgroups according to subject, achieving as high as an 80% correct classification rate. There are also several obvious directions for improving the system's classification performance in those cases where it did not do as well.

The system is based on calculating and comparing profiles of N-gram frequencies. First, we use the system to compute profiles on training set data that represent the various categories, e.g., language samples or newsgroup content samples. Then the system computes a profile for a particular document that is to be classified. Finally, the system computes a distance measure between the document's profile and each of the category profiles. The system selects the category whose profile has the smallest distance to the document's profile. The profiles involved are quite small, typically 10K bytes for a category training set, and less than 4K bytes for an individual document.

Using N-gram frequency profiles provides a simple and reliable way to categorize documents in a wide range of classification tasks.

1.0 Introduction

Electronic documents come from a wide variety of sources. Many are generated with various word processing software packages, and are subjected to various kinds of automatic scrutiny, e.g., spelling checkers, as well as to manual editing and revision. Many other documents, however, do not have the benefit of this kind of scrutiny, and thus may contain significant numbers of errors of various kinds. Email messages and bulletin board postings, for example, are often composed on the fly and sent without even the most cursory levels of inspection and correction. Also, paper documents that are digitally scanned and run through an OCR system will doubtless contain at least some recognition errors. It is precisely on these kinds of documents, where further manual inspection and correction is difficult and costly, that there would be the greatest benefit in automatic processing.

One fundamental kind of document processing is text categorization, in which an incoming document is assigned to some pre-existing category. Routing news articles from a newswire is one application for such a system. Sorting through digitized paper archives would be another. These applications have the following characteristics:

- The categorization must work reliably in spite of textual errors.
- The categorization must be efficient, consuming as little storage and processing time as possible, because of the sheer volume of documents to be handled.
- The categorization must be able to recognize when a given document does *not* match any category, or when it falls *between* two categories. This is because category boundaries are almost never clear-cut.

In this paper we will cover the following topics:

- Section 2.0 introduces N-grams and N-gram-based similarity measures.
- Section 3.0 discusses text categorization using N-gram frequency statistics.
- Section 4.0 discusses testing N-gram-based text categorization on a language classification task.
- Section 5.0 discusses testing N-gram-based text categorization on a computer newsgroup classification task.
- Section 6.0 discusses some advantages of N-gram-based text categorization over other possible approaches.
- Section 7.0 gives some conclusions, and indicates directions for further work.

2.0 N-Grams

An N-gram is an N-character slice of a longer string. Although in the literature the term can include the notion of any co-occurring set of characters in a string (e.g., an N-gram made up of the first and third character of a word), in this paper we use the term for contiguous slices only. Typically, one slices the string into a set of overlapping N-grams. In our system, we use N-grams of several different lengths simultaneously. We also append blanks to the beginning and ending of the string in order to help with matching beginning-of-word and ending-of-word situations. (We will use the underscore character (“_”) to represent blanks.) Thus, the word “TEXT” would be composed of the following N-grams:

bi-grams:	_T, TE, EX, XT, T_
tri-grams:	_TE, TEX, EXT, XT_, T_ _
quad-grams:	_TEX, TEXT, EXT_, XT_ _, T_ _ _

In general, a string of length *k*, padded with blanks, will have *k*+1 bi-grams, *k*+1tri-grams, *k*+1 quad-grams, and so on.

N-gram-based matching has had some success in dealing with noisy ASCII input in other problem domains, such as in interpreting postal addresses ([1] and [2]), in text retrieval ([3] and [4]), and in a wide variety of other natural language processing applications[5]. The key benefit that N-gram-based matching provides derives from its very nature: since every string is decomposed into small parts, any errors that are present tend to affect only a limited number of those parts, leaving the remainder intact. If we count N-grams that are common to two strings, we get a measure of their similarity that is resistant to a wide variety of textual errors.

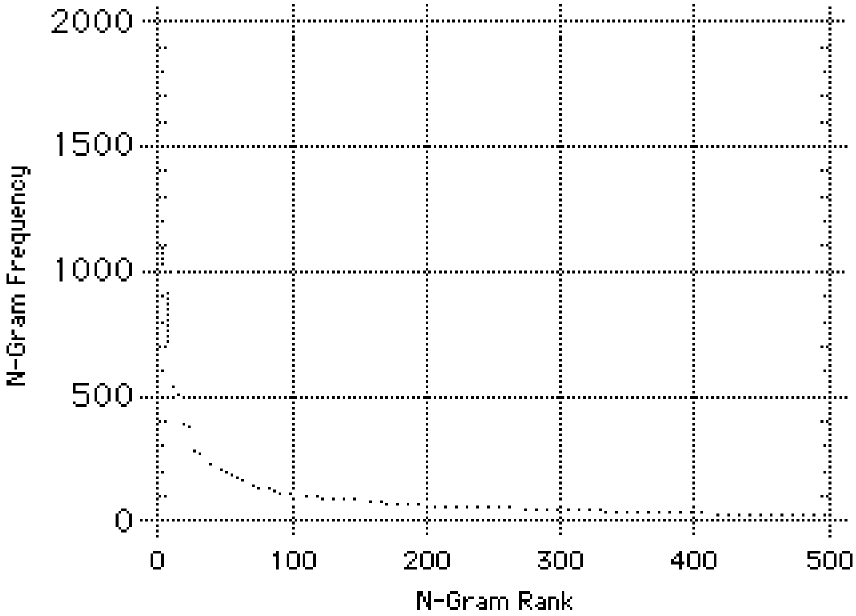
3.0 Text Categorization Using N-Gram Frequency Statistics

Human languages invariably have some words which occur more frequently than others. One of the most common ways of expressing this idea has become known as Zipf’s Law [6], which we can re-state as follows:

> The *n*th most common word in a human language text occurs with a frequency inversely proportional to *n*.

The implication of this law is that there is always a set of words which dominates most of the other words of the language in terms of frequency of use. This is true both of words in general, and of words that are specific to a particular subject. Furthermore, there is a smooth continuum of dominance from most frequent to least. The smooth nature of the frequency curves helps us in some ways, because it implies that we do not have to worry too much about specific frequency thresholds. This same law holds, at least

FIGURE 1. N-Gram Frequencies By Rank In A Technical Document



approximately, for other aspects of human languages. In particular, it is true for the frequency of occurrence of N-grams, both as inflection forms and as morpheme-like word components which carry meaning. (See Figure 1 for an example of a Zipfian distribution of N-gram frequencies from a technical document.) Zipf's Law implies that classifying documents with N-gram frequency statistics will not be very sensitive to cutting off the distributions at a particular rank. It also implies that if we are comparing documents from the same category they should have similar N-gram frequency distributions.

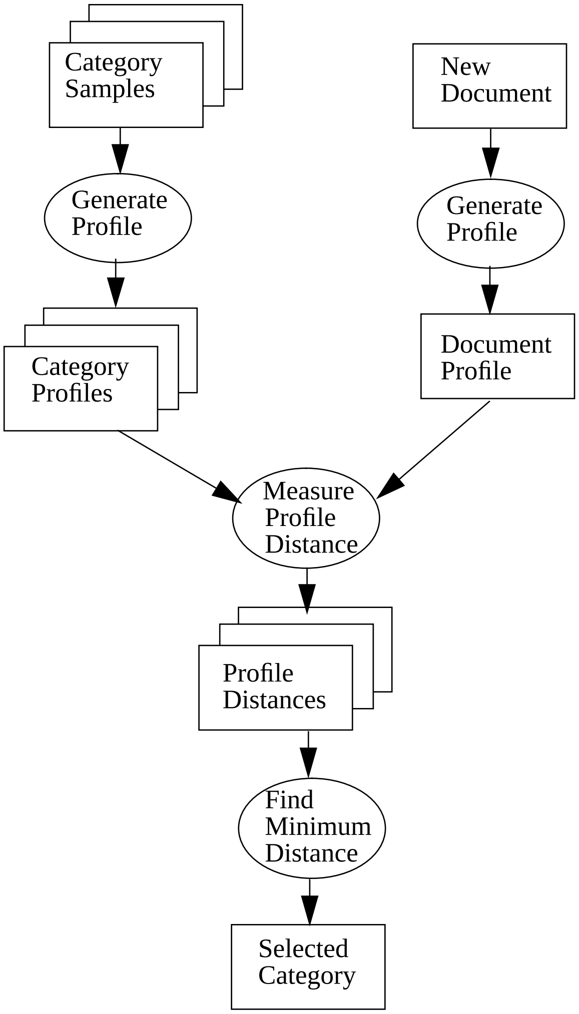
We have built an experimental text categorization system that uses this idea. Figure 2 illustrates the overall data flow for the system. In this scheme, we start with a set of pre-existing text categories (such as subject domains) for which we have reasonably sized samples, say, of 10K to 20K bytes each. From these, we would generate a set of N-gram frequency profiles to represent each of the categories. When a new document arrives for classification, the system first computes its N-gram frequency profile. It then compares this profile against the profiles for each of the categories using an easily calculated distance measure. The system classifies the document as belonging to the category having the smallest distance.

3.1 Generating N-Gram Frequency Profiles

The bubble in Figure 2 labelled "Generate Profile" is very simple. It merely reads incoming text, and counts the occurrences of all N-grams. To do this, the system performs the following steps:

- Split the text into separate tokens consisting only of letters and apostrophes. Digits and punctuation are discarded. Pad the token with sufficient blanks before and after.
- Scan down each token, generating all possible N-grams, for N=1 to 5. Use positions that span the padding blanks, as well.
- Hash into a table to find the counter for the N-gram, and increment it. The hash table uses a conventional collision handling mechanism to ensure that each N-gram gets its own counter.
- When done, output all N-grams and their counts.
- Sort those counts into reverse order by the number of occurrences. Keep just the N-grams themselves, which are now in reverse order of frequency.

FIGURE 2. Dataflow For N-Gram-Based Text Categorization



The resulting file is then an N-gram frequency profile for the document. When we plot the frequencies in this profile by rank, we get a Zipfian distribution graph very similar to that in Figure 2. We can make the following informal observations from an inspection of a number of different N-gram frequency profiles for a variety of different category samples:

- The top 300 or so N-grams are almost always highly correlated to the language. That is, a long English passage about compilers and a long English passage about poetry would tend to have a great many N-grams in common in the top 300 entries of their respective profiles. On the other hand, a long passage in French on almost any topic would have a very different distribution of the first 300 N-grams.
- The very highest ranking N-grams are mostly uni-grams (N=1), and simply reflect the distribution of the letters of the alphabet in the document's language. After that come N-grams that comprise function words (such as determiners) and very frequent prefixes and suffixes. There is, of course, a long tail to the distribution of language-specific N-grams, and it goes well past 300.
- Starting around rank 300 or so, an N-gram frequency profile begins to show N-grams that are more specific to the subject of the

document. These represent terms and stems that occur very frequently in documents about the subject.

- There is nothing special about rank 300 itself, since Zipf's law gives us in fact a very smooth distribution curve. Rather, we arrived at this number mostly by inspection. Doubtless, one could do more elaborate statistics and choose an optimal cutoff rank for a particular application.

We should note that these observations apply mostly to shorter documents, such as those from newsgroups. If documents were longer, the shift from language-specific N-grams to subject-specific N-grams would likely occur at a later rank.

3.2 Comparing and Ranking N-Gram Frequency Profiles

The bubble in Figure 2 labelled "Measure Profile Distance" is also very simple. It merely takes two N-gram profiles and calculates a simple rank-order statistic we call the "out-of-place" measure. This measure determines how far out of place an N-gram in one profile is from its place in the other profile. Figure 3 gives a simple example of this calculation using a few N-grams. For each N-gram in the document profile, we find its counterpart in the category profile, and then calculate how far out of place it is. For example, in Figure 3, the N-gram "ING" is at rank 2 in the document, but at rank 5 in the category. Thus it is 3 ranks out of place. If an N-gram (such as "ED" in the figure) is not in the category profile, it takes some maximum out-of-place value. The sum of all of the out-of-place values for all N-grams is the distance measure for the document from the category. We could also use other kinds of statistical measures for ranked lists (such as the Wilcoxin rank sum test). However, the out-of-place score provides a simple and intuitive distance measure that seems to work well enough for these proof-of-concept tests.

Finally, the bubble labelled "Find Minimum Distance" simply takes the distance measures from all of the category profiles to the document profile, and picks the smallest one.

4.0 Testing N-Gram-Based Text Categorization on Language Classification

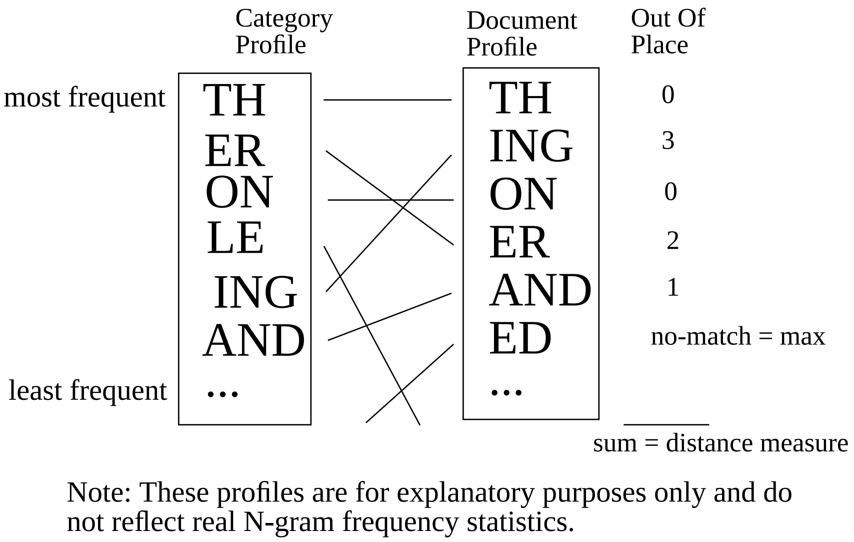
Most writing systems support more than one language. For example, nearly all of the languages from the former Soviet Union use the Cyrillic script. Given a text that uses a particular writing system, it is necessary to determine the language in which it is written before further processing is possible.

There are several broad approaches to the language classification problem. One obvious technique is to keep a lexicon for each possible language, and then to look up every word in the sample text to see in which lexicon it falls. The lexicon that contains the most words from the sample indicates which language was used.

However, building or obtaining representative lexicons is not necessarily easy, especially for some of the lesser-used languages. Furthermore, if the language is highly inflected, that is, using many different forms for each word to indicate case, tense or other attributes, then either the lexicons must become several times larger to get the necessary word inclusion, or one must develop some language-specific morphological processing to reduce different forms to their stems. Finally, if the text is the result of an OCR process, there may be recognition errors due to poor image quality, and these will disrupt the lexicon lookup process.

Another approach to language classification involves the use of N-gram analysis. The basic idea is to identify N-grams whose occurrence in a document gives strong evidence for or against identification of a text as belonging to a particular language. Although this has been done before, it makes a good test case for our text categorization method. We can use the N-gram frequency profile technique to classify documents according to their language without building a lexicon or a set of morphological processing rules. Instead, we need merely obtain modestly sized sample texts (10K to 20K bytes), calculate the N-gram frequency profiles, and use those to classify the documents.

FIGURE 3. Calculating The Out-Of-Place Measure Between Two Profiles



4.1 Language Classification Testing Procedure

In this test, our N-gram-based text categorization system very reliably identified the language of electronic mail messages taken from some of the Usenet newsgroups. These messages came in a variety of languages, but were all presented in standard ASCII, with a few typographical conventions to handle such things as diacritical markings. The classification procedure was as follows:

- Obtained training sets (category samples) for each language to be classified. Typically, these training sets were on the order of 20K to 120K bytes in length. There was no particular format requirement, but each training set did not contain samples of any language other than the one it was supposed to represent.
- Computed N-gram frequency profiles on the training sets as described above.
- Computed each article's N-gram profile as described above. The resulting profile was on the order of 4K in length.
- Computed an overall distance measure between the sample's profile and the category profile for each language using the out-of-place measure, and then picked the category with the smallest distance.

Such a system has modest computational and storage requirements, and is very effective. It requires no semantic or content analysis apart from the N-gram frequency profile itself.

4.2 Language Classification Test Data

For this test, we collected 3713 language samples from the soc.culture newsgroup hierarchy of the Usenet. These newsgroups are devoted to discussions about topics relevant to particular countries or cultures. Generally, those discussions were in the language of the particular country/culture, although some articles were partly or wholly in English. Table 1 gives a breakdown of the number of samples for each group, the supposed principal language for the group, the number of non-English articles, the number of English articles, the number of mixed language articles, the number of articles that contain junk (i.e., not a body of recognizable text), and the number of usable articles (pure English or pure non-English) for the test.

The sample articles ranged in size from a single line of text to as much as 50K bytes, with the

TABLE 1. Breakdown of Articles From Newsgroups

Newsgroup	Language	# Art.	Non-Engl	Engl.	Mixed	Junk	Usable
australia	English	104	0	104	0	0	104
brazil	Portuguese	86	46	10	13	17	56
britain	English	514	0	509	0	5	508
canada	English	257	0	251	3	3	251
celtic	English	347	0	345	0	2	345
france	French	294	200	73	17	4	273
germany	German	505	73	408	13	11	481
italy	Italian	336	293	23	13	7	316
latinamerica	Spanish	275	92	133	5	45	225
mexico	Spanish	288	197	66	7	18	263
netherlands	Dutch	255	184	51	15	5	235
poland	Polish	127	92	25	7	3	117
portugual	Portuguese	97	68	27	0	2	95
spain	Spanish	228	176	33	12	7	209
Totals		**3713**	**1421**	**2058**	**105**	**129**	**3478**

average around 1700 bytes. The sample extraction program also removed the usual header information, such as subject and keyword identification, leaving only the body of the article. This prevented any matches that were too strongly influenced by standard header information for the newsgroup (e.g., the newsgroup name or other lengthy identification phrases). For each language, we also assembled from manually selected and edited newsgroup articles an independent training set of 20K to 120K bytes in length. The N-gram frequency files for these training sets become the language profiles used by the classification procedure.

We determined the true classification for each test sample semi-automatically. First, we assumed that each sample was in fact in the language corresponding to the dominant language for the newsgroup it came from. For example, we would expect that a sample from the france newsgroup would be in French. This produced a default classification for each sample. Then we classified the sample with the procedure outlined earlier. We compared the resulting classification to the default one. If there was a discrepancy, that is, if the classification procedure identified the sample as being from some language other than the default, we then manually inspected the sample and gave it a corrected classification, if necessary. We also determined by this process articles which had mixed languages (e.g., interspersed passages in English and Portuguese) or junk (no recognizable body of text) and removed them from the test set. The resulting test set consisted of 3478 usable articles consisting of reasonably pure samples of a single language.

4.3 Language Classification Results

We have categorized the results along several dimensions. First, we kept track of whether the original article was over or under 300 bytes in length. Our initial hypothesis was that the system would have more problems classifying shorter messages because there would be a smaller amount of text from which to compute N-gram frequencies. On the whole, the system was only slightly sensitive to length. Second, we also varied the number of the N-gram frequencies available in the profile for the match, by limiting it to

TABLE 2. Percent Correct Classification

Article Length	**≤ 300**	**≤ 300**	**≤ 300**	**≤ 300**	**> 300**	**> 300**	**> 300**	**> 300**
Profile Length	**100**	**200**	**300**	**400**	**100**	**200**	**300**	**400**
Newsgroup								
australia	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
brazil	70.0	80.0	90.0	90.0	91.3	91.3	95.6	95.7
britain	96.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0
canada	100.0	100.0	100.0	100.0	100.0	*99.6	100.0	100.0
celtic	100.0	100.0	100.0	100.0	99.7	100.0	100.0	100.0
france	90.0	95.0	100.0	*95.0	99.6	99.6	*99.2	99.6
germany	100.0	100.0	100.0	100.0	98.9	100.0	100.0	100.0
italy	88.2	100.0	100.0	100.0	91.6	99.3	99.6	100.0
latinamerica	91.3	95.7	*91.3	95.7	97.5	100.0	*99.5	*99.0
mexico	90.6	100.0	100.0	100.0	94.8	99.1	100.0	*99.5
netherlands	92.3	96.2	96.2	96.2	96.2	99.0	100.0	100.0
poland	93.3	93.3	100.0	100.0	100.0	100.0	100.0	100.0
portugual	100.0	100.0	100.0	100.0	86.8	97.6	100.0	100.0
span	81.5	96.3	100.0	100.0	90.7	98.9	98.9	99.45
Overall	**92.9**	**97.6**	**98.6**	**98.3**	**97.2**	**99.5**	**99.8**	**99.8**

Note: Asterisks indicate combinations of test variables that did worse than similar combinations using shorter profiles.

statistics for 100, 200, 300 or 400 N-grams. This variable did have an impact on match performance, although by the 400 N-gram level language classification was almost perfect. Table 2 gives the classification percent correct for each combination of test variables, while Table 3 gives the ratio of errors committed to samples processed.

These results show some interesting patterns:

- The classification procedure works a little better for longer articles, but not quite as much as we expected.
- For the most part, the classification procedure works better the longer the category profile it has to use for matching. However, there were some interesting anomalies, indicated by the cells with asterisks on Table 2. These represent combinations of test variables that did worse than similar combinations with shorter profiles. In other words, for these cases, using more N-gram frequencies actually decreased classification performance. Post mortem examination of the problematic articles showed that at least part of the difficulty was that, in spite of the manual truthing efforts to remove mixed text, some articles still had passages from two languages. The interfering passages were mostly in the so-called signature blocks which are customary at the end of Usenet articles. In mixed language situations, this system, which used a forced choice selection, had no good mechanism for dealing with two language profiles with very similar distance measures from the article. In this case, adding statistics for more N-grams may then push one distance measure slightly ahead of the other in a hard-to-predict fashion.

Overall, the system yielded its best performance at a profile length of 400 N-grams. At this

TABLE 3. Ratio of Incorrect Classifications To Total Possible Classifications

Article Length	**≤ 300**	**≤ 300**	**≤ 300**	**≤ 300**	**> 300**	**> 300**	**> 300**	**> 300**
Profile Length	**100**	**200**	**300**	**400**	**100**	**200**	**300**	**400**
Newsgroup								
australia	0/12	0/12	0/12	0/12	0/92	0/92	0/92	0/92
brazil	3/10	2/10	1/10	1/10	4/46	4/46	2/46	2/46
britain	1/32	0/32	0/32	0/32	0/476	0/476	0/476	0/476
canada	0/19	0/19	0/19	0/19	0/232	1/232	0/232	0/232
celtic	0/18	0/18	0/18	0/18	1/327	0/327	0/327	0/327
france	2/20	1/20	0/20	1/20	1/253	1/253	2/253	1/253
germany	0/32	0/32	0/32	0/32	5/449	0/449	0/449	0/449
italy	2/17	0/17	0/17	0/17	25/299	2/299	1/299	0/299
latinamerica	2/23	1/23	2/23	1/23	5/202	0/202	1/202	2/202
mexico	3/32	0/32	0/32	0/32	12/231	2/231	0/231	1/231
netherlands	2/26	1/26	1/26	1/26	8/209	2/209	0/209	0/209
poland	1/15	1/15	0/15	0/15	0/102	0/102	0/102	0/102
portugual	0/12	0/12	0/12	0/12	11/83	2/83	0/83	0/83
span	5/27	1/27	0/27	0/27	17/182	2/182	2/182	1/182
Overall	**21/295**	**7/295**	**4/295**	**4/295**	**89/3183**	**16/3183**	**8/3183**	**7/3183**

level, the system misclassified only 7 articles out of 3478, yielding an overall classification rate of 99.8%.

5.0 Testing N-Gram-Based Text Categorization on Subject Classification

The same text categorization approach easily extends to the notion of using N-gram frequency to measure subject similarity for documents that are in the same language. Indeed, the approach extends to a multi-language database where both the language and the content of the document are of interest in the retrieval process. In order to test this approach, we used this classification system to identify the appropriate newsgroup for newsgroup articles. The articles for this experiment came from some of the Usenet newsgroups. We wished to see how accurately the system would identify which newsgroup each message *originally* came from. The classification procedure was as follows:

- Obtained training sets for each newsgroup. For this purpose, we used articles known as frequently-asked-question (FAQ) lists. Many newsgroups regularly publish such FAQs as a way of reducing traffic in the group by answering questions or discussing issues that come up a lot in the group. In this sense, then, the FAQ for a newsgroup tries to define what the newsgroup is (and is not) about, and as such contains much of the core terminology for the group. The FAQs we have collected are between 18K and 132K in length. There is no particular format requirement, but the FAQ should provide adequate coverage for the subject matter of the newsgroup.
- Computed N-gram frequencies on the newsgroup’s FAQ. These are exactly the same as the other kinds of N-gram fre-

quency profiles mentioned earlier. The resulting profiles are quite small, on the order of 10K bytes or less.

- Computed an article's N-gram profile in a fashion similar to that for computing the profile for each FAQ. The articles averaged 2K in length and the resulting article profiles were on the order of 4K in length.
- Computed an overall distance measure between the article's profile and the profile for each newsgroup's FAQ. The FAQ profile with the smallest distance measure from the article's profile determined which newsgroup to classify the sample as.
- Compared the selected newsgroup from the actual one the article came from.

5.1 Subject Classification Test Data

To test this system, we collected article samples from five Usenet newsgroups. These newsgroups are shown in Table 4. We chose these five because they were all subfields of computer science, and thus would provide an opportunity for testing how the system might confuse newsgroups that were somewhat closely related. The article extraction program also removed the usual header information such as subject and keyword identification, leaving only the body of the article. This prevented any matches that were too strongly influenced by standard header information for the newsgroup (e.g., the newsgroup name). For the profiles, we chose the FAQs shown in Table 5. Notice that there is some, but not perfect, overlap with the selected newsgroups for the experiment:

- There are FAQs for rec.games.go and comp.robotics, but no articles from either group.
- There are two FAQs related to compression, covering slightly different areas.
- There are articles for comp.graphics, but no FAQ.

Given this setup, we ran the classification procedure outlined above for all 778 newsgroup articles against the 7 selected FAQs. Our results are shown in Table 6. In the table, we can see the following strong results:

- The security FAQ provides 77% coverage of alt.security.
- The compilers FAQ provides 80% coverage of comp.compilers.
- The compression and jpeg_compression FAQs together provide 78% coverage of comp.compression.
- The go FAQ picked up only 3 articles altogether, indicating that its coverage is almost completely disjoint from the five selected newsgroups.

There are also these somewhat weaker results:

- The robotics FAQ picked up 11 ai articles and 23 graphics articles. This is probably because of the relative proximity of these subfields to robotics.
- The ai FAQ provides only 30% coverage of the comp.ai group. Noticing that the ai FAQ is nearly twice as large as the next largest FAQ, we can speculate that it may in fact cover too much material, thus throwing off the statistical nature of the N-gram frequency measure. This may also reflect the fact that comp.ai really consists of several related but distinct subgroups (expert systems, connectionism/neural networks, vision systems, theorem provers, etc.) that happen to share the same newsgroup.
- The articles from comp.graphics were distributed among the other FAQs. This is not unexpected since we did not include the FAQ from comp.graphics for the articles to match to. It is interesting that the strongest matching FAQ for these articles was jpeg_compression, which covers a compression standard for graphical data, and thus was a strong plausible contender for the match. It earned a 44% coverage of comp.graphics.

Overall, the system works quite well given the somewhat noisy nature of the newsgroups,

TABLE 4. Article Samples

Group	Abbrev.	#Articles	Covers
alt.security	security	128	computer security issues
comp.ai	ai	145	general artificial intelligence issues
comp.compilers	compilers	66	programming language compilers and interpreters
comp.compression	compression	187	techniques and programs for data compression
comp.graphics	graphics	252	general computer graphics issues

TABLE 5. Frequently Asked Question Articles

FAQ	Size	Origin
security	49K	FAQ from alt.security
ai	132K	FAQ from comp.ai
compilers	18K	FAQ from comp.compilers
compression	75K	basic FAQ from comp.compression
jpeg_compression	52K	special FAQ from comp.compression devoted to the JPEG standard for compressing graphics data
robotics	51K	FAQ from comp.robotics
go	21K	FAQ from rec.games.go (the game of go)

TABLE 6. Classification Results

Best-Matching	**Articles from Original Groups**				
FAQ	security	ai	compilers	compression	graphics
security	99	69	2	29	63
ai	3	44	7	1	13
compilers	4	11	53	7	19
compression	14	5	1	65	21
jpeg_compression	8	4	1	81	113
robotics	0	11	2	2	23
go	0	1	0	2	0
Total	128	145	66	187	252

and the necessarily incomplete nature of the FAQ lists. Although we do not analyze it here, cursory manual examination of the results showed that when the system matched an article against the incorrect FAQ, the correct FAQ was generally the second choice. Another thing to keep in mind is that we did not determine the actual contents of each article to see if it rightly belonged to the group it appeared in. In Usenet newsgroups, spurious cross-posting of irrelevant articles (e.g., conference announcements for other slightly related research areas) does happen on occasion, and some of those are present in our samples. Also, it is entirely possible for articles to be truly interdisciplinary, e.g., an article on using advanced AI techniques for detecting hacker intrusion patterns could appear in alt.security. Such an article might match strongly to two groups simultaneously.

6.0 Advantages of the N-Gram Frequency Technique

The primary advantage of this approach is that it is ideally suited for text coming from noisy sources such as email or OCR systems. We originally developed N-gram-based approaches to various document processing operations to use with very low-quality images such as those found in postal addresses. Although one might hope that scanned documents that find their way into text collections suitable for retrieval will be of somewhat higher quality, we expect that there will be a large amount of variability in the document database. This variability is be due to such factors as scanner differences, original document printing quality, low quality photocopies, and faxes, as well as preprocessing and character recognition differences. Our N-gram-based scheme provides robust access in the face of such errors. This capability may make it acceptable to use a very fast but low quality character recognition module for similarity analysis.

It is possible that one could achieve similar results using whole word statistics. In this approach, one would use the frequency statistics for whole words. However, there are several possible problems with this idea. One is that the system becomes much more sensitive to OCR problems—a single misrecognized character throws off the statistics for a whole word. A second possible difficulty is that short passages (such as Usenet articles) are simply too short to get representative subject word statistics. By definition, there are simply more N-grams in a given passage than there are words, and there are consequently greater opportunities to collect enough N-grams to be significant for matching. We hope to directly compare the performance of N-gram-based profiling with whole-word-based profiling in the near future.

Another related idea is that by using N-gram analysis, we get word stemming essentially for free. The N-grams for related forms of a word (e.g., ‘advance’, ‘advanced’, ‘advancing’, ‘advancement’, etc.) automatically have a lot in common when viewed as sets of N-grams. To get equivalent results with whole words, the system would have to perform word stemming, which would require that the system have detailed knowledge about the particular language that the documents were written in. The N-gram frequency approach provides language independence for free.

Other advantages of this approach are the ability to work equally well with short and long documents, and the minimal storage and computational requirements.

7.0 Conclusions And Future Directions

The N-gram frequency method provides an inexpensive and highly effective way of classifying documents. It does so by using samples of the desired categories rather than resorting to more complicated and costly methods such as natural language parsing or assembling detailed lexicons. Essentially this approach defines a “categorization by example” method. Collecting samples and building profiles can even be handled in a largely automatic way. Also, this system is resistant to various OCR problems, since it depends on the statistical properties of N-gram occurrences and not on any particular occurrence of a word.

Although the existing system already has demonstrated good performance, there is considerable room for further work:

- Currently the system uses a number of different N-grams, some of which ultimately are more dependent on the language of the document than the words comprising its content. By omitting the statistics for those N-grams which are extremely common because they are essentially features of the language, it may be possible to get better discrimination from those statistics that remain. It is also possible that the system should include some additional statistics for rarer N-grams, thus gaining further coverage.

- It seems clear that the quality of the document set affects the subject categorization performance. We would like to experiment with document sets that have a higher overall coherence and quality. For example, it would be interesting to test this technique on a set of technical abstracts for several different areas. By splitting the set for each area into training and testing portions, then computing the profile for each area from the training set, we could repeat this experiment in a more controlled way.
- In a related issue, the quality of the training set in general greatly affects matching performance. Although the FAQs were easy to obtain and work with, other training sets might have produced better results, even for these newsgroups. Of necessity, a FAQ lags the group it covers, since new "hot" topics of discussion have not yet made it into the FAQ. To test this, it would be interesting to compare the FAQ-based profiles with profiles derived from a separate set of articles from the appropriate newsgroups.
- The raw match scores the system produces are largely useless by themselves except for imposing an overall relative ordering of matches for the various profiles. To correct this, we must devise a good normalization scheme, which would produce some sort of absolute measure of how good a particular match really is. This would allow the system to reject some documents on the grounds that their normalized scores were so low that the documents did not have good matches at all. Normalized scores would also let the system determine if a particular document lay between two classifications because of its interdisciplinary nature. A related idea would be to see how well the system could predict which articles get cross-posted to different groups precisely because of their interdisciplinary content.
- This type of document similarity measure is ideally suited for document filtering and routing. All that a user needs to do is collect a representative set of documents that cover the relevant topics, then compute an overall profile. From that point on, it is simple and cheap to compute the profile of every incoming document, match it against the user's overall profile, and accept those whose match scores are sufficiently good.
- This system currently handles only languages that are directly representable in ASCII. The emerging ISO-6048/UNICODE standard opens up the possibility of applying the N-gram frequency idea to all of the languages of the world, including the ideographic ones.

Acknowledgments

The authors gratefully acknowledge the very useful remarks and suggestions for this paper by David Lewis, of AT&T.

References

[1] Cavnar, William B. and Vayda, Alan J., "Using superimposed coding of N-gram lists for Efficient Inexact Matching", *Proceedings of the Fifth USPS Advanced Technology Conference*, Washington D.C., 1992.

[2] Cavnar, William B. and Vayda, Alan J., "N-gram-based matching for multi-field database access in postal applications", *Proceedings of the 1993 Symposium On Document Analysis and Information Retrieval*, University of Nevada, Las Vegas.

[3] Cavnar, William B., "N-Gram-Based Text Filtering For TREC-2," to appear in the *proceedings of The Second Text REtrieval*

Conference (TREC-2), ed. by, Harman, D.K., NIST, Gaithersburg, Maryland, 1993.

[4] Kimbrell, R.E., “Searching for Text? Send and N-gram!,” *Byte*, May 1988, pp. 297-312.

[5] Suen, Ching Y., “N-Gram Statistics for Natural Language Understanding and Text Processing,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-1, No. 2, April 1979, pp.164-172.

[6] Zipf, George K., Human Behavior and the Principle of Least Effort, an Introduction to Human Ecology, Addison-Wesley, Reading, Mass., 1949.