Subject Classification in the Oxford English Dictionary

Zarrin Langari Frank Wm. Tompa

Department of Computer Science
University of Waterloo
Waterloo, ON
Canada
{zlangari, fwtompa}@db.uwaterloo.ca

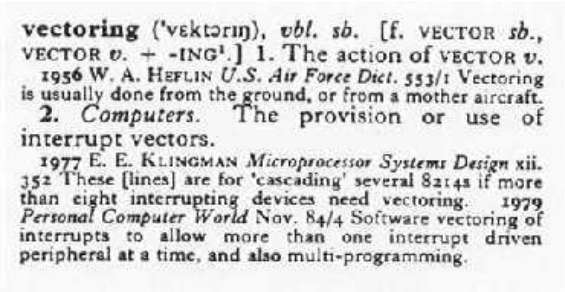
Abstract

TheOxford English Dictionary is a valuable source of lexical information and a rich testing ground for mining highly structured text. Each entry is organized into a hierarchy of senses, which include definitions, labels and cited quotations. Subject labels distinguish the subject classification of a sense, for example they signal how a word may be used in Anthropology, Music or Computing. Unfortunately subject labeling in the dictionary is incomplete. To overcome this incompleteness, we attempt to classify the senses (i.e., definitions) in the dictionary by their subjects, using the citations as an information guide. We report on four different approaches: k Nearest Neighbors, a standard classification technique; Term Weighting, an information retrieval method dealing with text; Naive Bayes, a probabilistic method; and Expectation Maximization, an iterative probabilistic method. Experimental performance of these methods is compared based on standard classification metrics.

1 Introduction

Dictionaries encode rich semantic structures organized as a sequence of entries. Each entry for a word in a typical monolingual dictionary includes some main definitions and additional information about those words that helps readers to understand usage more easily. In the *Oxford English Dictionary* [18] (the OED) an *entry*, such as the one show in Figure 1, contains various structural elements such as headword, pronunciation, part of speech, homonym number, usage labels, etymology and historical and current senses. Each *sense* contains a sense number, definition and cited quotations illustrating usage.

Some applications may need to distinguish those dictionary senses related to a chosen discipline. For example, we may wish to extract a historical *computing* dictionary, containing only those senses from the OED pertaining to computing. Unfortunately the dictionary's editors did not include labels everywhere they apply, nor are labels abbreviated consistently. For example, although the entry for **recursive** has senses labeled *Math* and *Logic*, *Linguistics*, *Computing,* and *Phonetics,* the entry for **recursion** includes no explicit subject labels at all. Thus the user must rely on other means to select all subject-specific word senses.



vectoring ('vɛktərɪŋ), *vbl. sb.* [f. VECTOR *sb.*, VECTOR *v.* + -ING¹.] 1. The action of VECTOR *v.*
1956 W. A. HEFLIN *U.S. Air Force Dict.* 553/1 Vectoring is usually done from the ground, or from a mother aircraft.
2. *Computers.* The provision or use of interrupt vectors.
1977 E. E. KLINGMAN *Microprocessor Systems Design* xii. 352 These [lines] are for 'cascading' several 8214s if more than eight interrupting devices need vectoring. 1979 *Personal Computer World* Nov. 84/4 Software vectoring of interrupts to allow more than one interrupt driven peripheral at a time, and also multi-programming.

Figure 1. Simple entry from the OED [18]

In an attempt to mine this information, we exploit the observation that senses related to computing typically include illustrative citations from the computing literature, and this holds for other subjects as well. Our approach is inspired by the earlier work of Jonathan Spencer, who realized several years ago that the collections of citations for a word in *Anthropology* often includes works from anthropology, and who suggested finding other words or senses in the field by finding those that cite similar sources [2]. Citations include date, author, and source of each illustrative quotation, but there are no subject labels attached to citations. Thus we must rely on the information implicit in each citation paragraph or in our own experience to know which citations arise from which subject areas. We have investigated

a *supervised* text classification task, in which we use the citations of other senses from the OED with an explicit subject label to induce a subject label for those senses without a label but with similar citations.

This paper summarizes our experience with extracting interesting lexical data from a 540 megabyte, highly structured text. Providing a data model for the dictionary senses, and considering the subjects as categories, we begin our categorization task by preprocessing the dictionary to make it more appropriate for our needs. Starting with the tagged dictionary, for example, we remove all unnecessary information from the entries. We also create suitable indexes over the labels and citation information to support our task. Thereafter we apply standard classification techniques and evaluate their effectiveness for this problem. We found that unlike many other categorization tasks, the problem size and the sparseness of the data were significant obstacles to address.

The remainder of this paper is organized as follows. In Section 2 we introduce the corpus (the OED) and our data model for classification. Section 3 gives a brief overview of the *k Nearest Neighbor*s, *Term Frequency * Inverse Document Frequency, Naive Bayes,* and *Expectation Maximization* methods as they apply to our problem. Section 4 summarizes performance evaluation metrics, and Section 5 describes the experimental comparison of the classification methods for our task. In conclusion, we summarize our experience and describe anticipated future work.

2 Dictionary and Data Model

As mentioned in the introduction, the *Oxford English Dictionary,* like any other dictionary, has a specific structure to express the meanings of words [3]. In a typical OED entry a headword (i.e., the word being described by the entry) together with pronunciation, part of speech, and homonym number (if needed) forms a "headword group." Besides the headword group, three other main elements that appear for most entries are definitions, quotations and labels. *Definitions* appear in a hierarchical structure of numbered *senses* to express:

- grammatical and semantic subdivisions of the word,
- parallel meanings and usages of a word over time, and
- the chronological order of the uses of various meanings of a word.

As supporting evidence for each sense, *quotations* are provided to illustrate how the word being described has been used by various authors over time. Together with the quoted text, each citation typically contains the author's name, the title of the work in which the quotation appeared, the publication date, and the location of the quotation within the work. *Labels* are related to various uses of a word, and they may appear in a headword group or with individual word senses. Among different types of labels, we are interested only in those that denote a subject. The entry for **vectoring** includes two senses, the first of which is unlabelled and has one supporting citation and the second of which is labelled *Computers* and has two citations.

Our model is based on extracted data from each entry, as illustrated for the entry for **lymphatic** in Figure 2. From the

lymphatic, *a. sb.*
A
I
1
BLOUNT, *Glossogr.* — SHAFTESB, *Charac.*
BAILEY — S. BURDER, *Orient. Lit.*
A
II
2
a *Phys. Anat.*
EVELYN, *Mem.* — BOYLE, *Usef. Exp. Nat. Philos.*
ARBUTHNOT, *Rules of Diet.* — *Astruc's Fevers.*
ABERNETHY, *Surg. Obs.* — R. KNOX, *Beclard's Anat.*
ROLLESTON, *Anim. Life.* — *Allbutt's Syst. Med.*
A
II
2
b *Bot.*
GREW, *Anat. Plants.* — LOUDON, *Encycl. Plants*
A
II
3
J. FORBES, *Laennec's Dis. Chest.*
HAWTHORNE, *Fr. & It. Note-Bks.*
G. MEREDITH, *R. Feverel.* — HUXLEY, *Physiol.*
Truth
B
1
SHAFTESBURY, *Charac.* — SHENSTONE, *Elegies*
B
2
Phil. Trans. — FLOYER, *Physic. Pulse-Watch*
HEWSON, *Phil. Trans.* — GOOD, *Bk. Nat.*
MIVART, *Cat*

Figure 2. Extracted information from the entry **lymphatic**.

headword group we extract the headword, part of speech, and homonym number and ignore any other components. For each sense in the hierarchy, we generate a sense locator based on the sense numbers (e.g. **A.I.1**). The other interesting information in each entry is contained in the labels, although not all labels denote subjects. For example, the subject label *Bot.* indicates that sense definition **A.II.2.b** pertains to botany. The final interesting elements for our clas-

sification task are citation components, namely author and title. For example, the title *Rules of Diet* in sense **A.II.2.a** appears for entry senses of **larynx** and **serus** that are also labelled *Anatomy*.

Our task is to classify the senses from the OED by choosing an appropriate label for each sense based on the evidence found in the corresponding citations' authors and titles. We restrict ourselves to *single-label* classification, in which only the top ranked category is assigned to each unlabeled document (so-called *1-per-doc* category assignment). We also apply *thresholding* [12], in which we assign a document to a category if its similarity or its probability of membership is above a user-defined amount.

After appropriate preprocessing the dictionary data can be considered as a pair of relations, one mapping sense identifiers to labels and the other mapping sense identifiers to cited authors and titles. These relations represent a *vector model* in which the authors and titles are the *terms* used as a documents' features, the vector space is r-dimensional with one dimension for each possible term, and a specific document $\overline{d_j} =< f_{j1}, ..., f_{jr} >$ is a vector of *term frequencies* in which the value for each component in the vector is the number of occurrences of that term in the document.

With this model, we can apply a supervised classification method, which uses pre-labeled data (in our case the labeled senses) as a *training set* and builds a classifier by learning from this set. To categorize a new document, its characteristics are compared to those of the members of classified sets. For simplicity, we assume that an author or a title can be used individually as a term that may be associated with a label, that all the terms are independent of each other, and that authors or titles of the same name refer to the same entity. We believe that each of these assumptions, while generally invalid, reflect the state of the data closely enough that they will not be too detrimental to the performance of our classification strategies.

As an overview of the size of our data, we note that there are about 523,000 senses in the dictionary, of which only 63,000 are labeled with one of 128 unique subject labels that appear more than 45 times each. There are 300,000 unique references (cited authors or titles) that occur in 4.2 million instances overall, and on average each sense includes 8 to 10 references. Thus we must accommodate a very large data space, 523,000 documents×300,000 terms, that is very sparsely populated.

3 Classification Methods

For our experiments, we investigated the performance of four techniques. The k Nearest Neighbors method (kNN) is a lazy learning method and a simple technique, often with high performance results. As a second approach, we investigated the tf*idf term weighting technique, which is commonly used in information retrieval applications. The last two techniques, Naive Bayes (NB) and Expectation Maximization (EM), are chosen from the probabilistic methods: both attempt to estimate the parameters of a generating model that creates documents in categories based on an unknown set of probabilities. More detailed descriptions of the methods can be found elsewhere [10].

3.1 k Nearest Neighbors

Using the kNN method [19],[20], for each unlabeled data vector, we find the k nearest training data vectors in the multi-dimensional vector space, judging proximity by the often-used *Cosine* similarity measure:

$$sim(\overline{d_x}, \overline{d_y}) = \frac{\sum_{t=1}^{r} f_{xt} f_{yt}}{\sqrt{\sum_{t=1}^{r} f_{xt}^2} \sqrt{\sum_{t=1}^{r} f_{yt}^2}} \quad (1)$$

where r is the number of dimensions, $\overline{d_x}$ is a labeled document vector, $\overline{d_y}$ is an unlabeled document vector, and f_{xt} and f_{yt} are the term frequencies of these two documents [9]. We assign to the new object the category of the majority of its neighbors.

As well as choosing a similarity function, specific applications of kNN require settings for various thresholds. The first is the value of k, indicating how many neighbors are to be considered for picking a category. This threshold is usually determined experimentally by a validation set. Other parameters are minimal thresholds for voting: how many neighbors must be in the chosen category, how similar must those neighbors be to the new object, and so forth.

3.2 Term Frequency * Inverse Document Frequency

Selection of tf*idf, first developed for information retrieval [1][8][16], as the second method is intended to overcome the weaknesses of depending on *one-to-one* comparisons for very small (sparse) documents. If each instance of unlabeled data is compared against a set of senses from the labeled data, we can find more common references and thus potentially improve the results.

To apply this approach to classification, we consider the concatenation of the training objects in each category to be the documents in our collection (i.e., one aggregated document per category) and the data object to be classified as a "query." In matching a query to candidate documents, the term weighting system gives higher value to a term that occurs often in a document if it occurs relatively *infrequently* in the document collection as a whole. Following Salton's suggestions [16], we must also account for differences in category size and query length. Specifically, the term weight for the t^{th} term in category c_i is computed as:

$$w_{it} = \frac{f_{it} \log \frac{N}{n_t}}{\sqrt{\sum_{h=1}^{r} (f_{ih} \log \frac{N}{n_h})^2}} \quad (2)$$

where again r is the number of terms, f_{it} is the frequency of term t in category i, N is the number of categories, and n_t is the number of categories containing at least one occurrence of term t. The term weight for the t^{th} term in a data object to be classified (our "query") is computed as :

$$d'_{yt} = [0.5 + \frac{0.5 f_{yt}}{F_y}] \log \frac{N}{n_t} \quad (3)$$

where f_{yt} is the frequency of term t in the query and F_y is the highest term frequency among all the query object's terms. With these definitions of weights, the similarity of a category c_i to an unlabeled document $\overline{d_y}$ is then calculated as $sim(c_i, \overline{d_y})$ using the Cosine measure in Equation 1 above, but with these term weights substituted for the simple term frequencies that are used in kNN. We finally classify the test data object by assigning it the label of the most similar category.

It is important to keep in mind that whereas kNN compares an unlabeled document to individual labeled documents, tf*idf compares an unlabeled document to the set of labeled documents in each category taken as a whole. Nevertheless, as for kNN, tf*idf requires settings for thresholds for choosing the appropriate category: how similar the closest category is to the new object and how much more similar it is than the other categories.

3.3 Naive Bayes

The idea underlying the last two classifiers is to use probabilities of categories and category-dependent joint probabilities of terms occurring in documents to find the probability of a category given a document [15]. The probabilistic model is "naive" in its assumption of term independence, i.e., the conditional probability of a term given a category is assumed to be independent of the conditional probabilities of other terms given that category [13][20]. Because citations are occasionally repeated within a single sense, we are interested in *multinomial* Bayesian classifiers, for which term frequencies, rather than merely presence or absence of terms, are used in computing the joint probabilities.

The metrics for these methods are based on $P(c_i|\overline{d_j})$, the probability of document $\overline{d_j}$ belonging to class c_i, for all the classes $i = 1..N$:

$$P(c_i|\overline{d_j}) = \frac{P(c_i) * \prod_{t=1}^{r} P(T_{jt}|c_i)^{f_{jt}}}{\sum_{q=1}^{N} (P(c_q) * \Pi_{t=1}^{r} P(T_{jt}|c_q)^{f_{jt}})} \quad (4)$$

where $\overline{d_j}$ is again a vector of term frequencies f_{jt}, $P(c_i)$ is the prior probability that a randomly picked document falls within category c_i, $P(T_{jt}|c_i)$ is the probability of generating the t^{th} term for document j if it is in the i^{th} class, and r is again the number of distinct terms in the document space.

To calculate these probabilities, we need to estimate the probability of a term given a specific class. The maximum likelihood value is the total number of times that a term occurs in that class (for training data) divided by the total number of terms in that class:

$$P(T_{jt}|c_i) = \frac{\sum_{j=1}^{|D_l|} f_{jt} * B_{ji}}{\sum_{s=1}^{r} \sum_{j=1}^{|D_l|} f_{js} * B_{ji}} \quad (5)$$

where B_{ji} is 1 if and only if document $\overline{d_j}$ belongs to category c_i and $|D_l|$ is the number of training documents. The class prior probability is also similarly estimated:

$$P(c_i) = \frac{\sum_{j=1}^{|D_l|} B_{ji}}{|D_l|} \quad (6)$$

In practice, however, we need to account for the situation in which a term does not occur in some class in any of the training data but could appear in that class in some unseen data. In these situations, we must avoid assigning zero probability for that term given that class. Following best practices, we use *Simple Good-Turing* to "smooth" the probabilities by distributing a small portion of the probability mass from observed data to terms that do not appear in the training set [5].

In summary, with the NB method, we classify a document by first computing maximum likelihood values for the class prior probability $P(c_i)$ for all the classes and for the (smoothed) probability of document terms given each class $P(T_{jt}|c_i)$. We then calculate the posterior probability of each class given a document that comprises specific terms $P(c_i|\overline{d_j})$. We finally choose the class with the highest probability, subject once again to passing thresholds corresponding to those used in the other approaches.

3.4 Expectation Maximization

In his original approach to labeling OED senses based on citations, Spencer included iterative steps to incorporate additional evidence introduced as unlabeled senses were incrementally classified [2]. A similar iterative algorithm has also been shown to be useful in extracting author-title pairs from the World Wide Web [4]. Following these approaches, we decided to evaluate the performance of Expectation Maximization (EM), in which we use the terms that appear in unlabeled documents to strengthen the evidence of a term occurring in a class. The iterative behavior of the EM method provides a form of *many-to-many* basis of comparison, since all unlabeled senses are simultaneously compared to all labeled senses.

For the first "expectation" step, we calculate $P(c_i|\overline{d_j})$ using the NB method described for each unlabeled document over all the classes. Then instead of using the labeled data only, in next "maximization" step we use tentatively classified data as well as the labeled data to calculate revised probabilities $P(c_i)$ for each class and $P(T_{jt}|c_i)$ for each term:

$$P(T_{jt}|c_i) = \frac{\sum_{j=1}^{|D|} f_{jt} * P(c_i|\overline{d_j})}{\sum_{s=1}^{s=r} \sum_{j=1}^{|D|} f_{js} * P(c_i|\overline{d_j})} \quad (7)$$

$$P(c_i) = \frac{\sum_{j=1}^{|D|} P(c_i|\overline{d_j})}{|D|} \quad (8)$$

where $P(c_i|\overline{d_j})$ are the probabilities for all documents (and thus $P(c_i|\overline{d_j})= B_{ji}$ for labeled documents). These are used in turn to revise the classification probabilities $P(c_i|\overline{d_j})$, and this cycle can be iterated until the probabilities $P(c_i|\overline{d_j})$ converge. At this stage, all the unlabeled data has been considered simultaneously, and all their probabilities for membership in each category have been estimated. Using these probabilities, we assign classes using the same thresholds we use for NB.

Experience has shown that EM improves over the NB method in applications where the assumptions are valid, i.e., the generative mixture model reflects the application as a whole and the probability models for each category adequately reflect the data objects, including the document length distribution and the assumption of term independence [15].

4 Evaluation

Before describing the details of the experiments, we need to determine how to measure the *effectiveness* of each method in classifying the dictionary senses. Recall and precision are two well-known metrics from information retrieval. In the context of text classification, they are defined as [7][11]:

$$Pr = \frac{\#\ of\ documents\ correctly\ assigned\ to\ category\ i}{\#\ of\ documents\ assigned\ to\ category\ i}$$

$$Re = \frac{\#\ of\ documents\ correctly\ assigned\ to\ category\ i}{\#\ of\ documents\ that\ should\ be\ assigned\ to\ category\ i}$$

which show how to compute the precision and recall over one category. There are two approaches to compute the overall precision and recall:

- *macroaveraging*: Overall precision and recall are computed by averaging the results of the values for each category, which therefore gives equal weight to each category. If m is the number of categories then:

$$Pr = \frac{\sum_{i=1}^{m} Pr_i}{m} \quad Re = \frac{\sum_{i=1}^{m} Re_i}{m}$$

- *microaveraging*: First, for each category the number of correctly classified documents a_i, incorrectly classified ones b_i, and incorrectly rejected documents c_i are computed. Then, giving equal weight to each document, overall precision and recall is computed based on the totals of these values in all categories:

$$Pr = \frac{\sum_{i=1}^{m} a_i}{\sum_{i=1}^{m} (a_i + b_i)} \quad Re = \frac{\sum_{i=1}^{m} a_i}{\sum_{i=1}^{m} (a_i + c_i)}$$

Most researchers believe that for classification purposes microaveraging is a better indicator, because each categorization decision is weighted equally, but some prefer macroaveraging in which frequent categories are not weighted more heavily than infrequent ones [17]. Finally, in order to have a single measure of performance, we chose to use a simple *F-measure* that balances precision and recall [17]:

$$F = \frac{2PrRe}{(Pr + Re)}$$

5 Experiments

For our experiments we used the 63,000 senses that were labeled by any of the 128 labels occurring in the dictionary more than 45 times each. These senses served as both the training data and the test data, using *3-fold cross-validation*. We partitioned the whole set into three subsets of equal size, randomly assigning documents to subsets, to obtain three instances of training and test data with about $\frac{1}{3}$ of data in each training set and $\frac{2}{3}$ in each test set. After performing categorization with each of the four methods, two different thresholds were applied: the first one is a threshold to pick the best category (majority number for kNN, similarity for tf*idf, and probability for NB and EM methods) and the second is the threshold for the difference between the best and the second-best category. Threshold values were chosen to maximize the F-measures as explained above, yielding the results shown in Table 1.

	kNN	tf*idf	NB	EM
threshold for best	0	0.01	0.7	0.5
threshold for 2nd-best	1	0.01	10^{-8}	10^{-8}
Macro F-measure	0.28	0.41	0.43	0.41
Micro F-measure	0.48	0.49	0.57	0.54

Table 1. Maximal F-measures and corresponding thresholds for best and second-best categories

For the kNN method, the value chosen for *k* was 100 to achive the best performance, although using any values of

k between 15 and 200 produced very similar results. As seen in the table, the best result was achieved for kNN by not imposing a threshold to pick the nearest neighbor of an unlabeled sense, which means that at least one neighbor belonging to a class is enough to select the category. The other threshold, however, indicates that the class with the highest representation should have at least one more member than the class with the next highest value; when two different classes have the same number of representatives within the nearest *k* neighbors, we do not assign either one.

For tf*idf to pick the most similar category, the best performance was achived using a threshold of 0.01 (that is, if the most similar category to the unlabeled document has a similarity measure less than 0.01, no category is assigned). Furthermore the most similar category should have a similarity measure that is at least 0.01 greater than the next closest one for it to be chosen. Interestingly, even though tf*idf uses consolidated information from all senses in a category to evaluate the similarity of a test sense, its F-measure based on microaveraging is comparable to that of kNN for this task.

NB and EM evaluate the fit of a test sense in terms of probabilities. Best performance was achieved for NB using 0.7 as the minimal probability for choosing a class for a given document. For EM best performance was achieved when the probability of a document to be assigned to a category was at least $\frac{1}{2}$. We found that for both probabilistic approaches, the most probable category and the next most probable one should have a difference in probability of at least 10^{-8} for best results.

Comparing the F-measures in Table 1, we note that Naive Bayes has the best performance results. We expected NB to outperform kNN since it uses consolidated information from all senses in a category, but we were surprised that it also outperformed EM. Apparently tuning the model through iterations causes it to diverge further from reality! The reason may be that the assumptions made for the probability model, primarily term independence, do not hold for our data. Perhaps the specific form of generating model may itself not match this data.

Table 2 shows the microaverage-based F-measures for each individual category, sorted by the best performance achieved across all four methods (shown in bold face in the table). The same data is presented graphically in Figures 3a, 3b, and 3c, which compare the performance of NB to each of the other three methods. For almost all classes NB outperforms the other methods. In 34 classes tf*idf performs slightly better than the others, and EM has somewhat better performance results in 25 classes. Notably kNN always underperforms the other methods, especially in some classes (for example, kNN has F-measure 0.01 for *Herb* whereas the other methods' performances are around 0.40); there are also five classes for which kNN has a zero F-measure (i.e. *no* documents were correctly classified).

Interestingly the classes with highest F-measures have very different sizes and correspond to subjects covering a variety of disciplines. We conclude that kNN is not a good classification method when each document is very small and the number of common terms is very low. tf*idf and EM perform somewhat better, especially for some of the smaller classes, but for some classes they are also poor classifiers.

Unfortunately, we have not been able to detect a pattern that distinguishes in which classes any of the methods perform well. As shown in Figure 3d, the performances are slightly higher in very large categories, but the size of category alone is not a good criterion for predicting performance. We note that the dictionary was created by many people over a long period, but we know of no peculiarities in the methods of assigning labels or in choosing which works to cite that can explain the deviations in performance. For example, as can be seen in Table 1, there are three different *Law* categories with good F-measures, but *Civil Law* (with 50 documents) has maximal F-measure of 0.11; *Chemistry* and *Physics* could be well-classified, *Biology* and *Physical Chemistry* could not; *Cricket* and *Golf* worked well, but *Football* and *Racing* did not.

6 Conclusion

In this work we defined the subject classification problem for the *Oxford English Dictionary* and expressed the theory behind four different text classification methods to be used for this kind of data, namely *k* Nearest Neighbors, Term Frequency * Inverse Document Frequency, Naive Bayes and Expectation Maximization. We presented a data model for the senses in the OED that is suitable for classifying dictionary senses based on citation information. Our implementation of this model was designed to accommodate the vastness and sparseness of the data.

We evaluated the performance of the four methods experimentally, using the labeled senses of the dictionary for both training and test data. The evaluation metric we used to compare the above methods is the F-measure, a combination of precision and recall. The results show that the kNN method has the lowest results, which we attribute to the lack of common citations between two small "sense documents." With tf*idf microaverage-based F-measures were marginally improved, but still below .5; perhaps an alternative weighting system for "category documents" or a different similarity measure would improve performance. NB and EM, the two methods based on probability models, show higher results, but surprisingly EM is worse than NB. This reinforces the recent results from other researchers, who found that "Naive Bayes usually performs classification well despite violations of its assumptions...but (in EM)

label	size	kNN	tf*idf	n. Bayes	EM	label	size	kNN	tf*idf	n. Bayes	EM
Her	1293	0.73	0.83	**0.84**	0.83	*Cinemat*	75	0.19	**0.44**	0.42	0.36
Computers	316	0.67	0.83	0.83	**0.83**	*Bookbinding*	106	0.25	**0.44**	0.39	0.28
Min	2444	0.74	0.74	**0.82**	0.81	*Hist*	3519	0.38	0.26	**0.43**	0.39
Surfing	61	0.73	0.80	**0.82**	0.74	*Coalmining*	158	0.09	**0.42**	0.29	0.34
Music	1852	0.66	0.76	**0.80**	0.80	*Theol*	413	0.19	0.36	0.40	**0.41**
Chem	4447	0.69	0.71	**0.79**	0.77	*Herb*	178	0.01	0.31	0.39	**0.40**
Mountaineering	57	0.69	**0.79**	0.77	0.73	*Gr Antiq*	164	0.34	0.36	**0.40**	0.39
Bridge	127	0.54	0.77	**0.78**	0.73	*Prosody*	214	0.18	0.36	**0.40**	0.39
Linguistics	365	0.68	0.71	0.74	**0.76**	*Painting*	127	0.25	0.38	**0.40**	0.36
Cricket	522	0.48	0.75	**0.76**	0.74	*Ophthalm*	56	0.06	0.31	**0.40**	0.34
Naut	3039	0.69	0.69	**0.76**	0.71	*Physical Chem*	86	0.16	0.36	**0.40**	0.29
Bot	4521	0.66	0.65	**0.75**	0.72	*Geom*	535	0.19	0.37	0.39	**0.39**
Psychol	483	0.55	0.66	0.72	**0.73**	*Biol*	1382	0.25	0.27	**0.39**	0.33
Logic	521	0.63	0.71	**0.73**	0.72	*Boxing*	57	0.19	0.36	**0.38**	0.32
Cookery	237	0.48	**0.70**	0.68	0.69	*Hunting*	186	0.25	0.35	**0.38**	0.28
Photogr	265	0.49	0.68	0.68	**0.69**	*Agric*	134	0.13	0.34	0.37	**0.38**
Statistics	122	0.28	0.58	**0.68**	0.65	*Philol*	197	0.12	0.29	0.34	**0.38**
Chess	129	0.43	**0.68**	0.66	0.64	*Theatr*	181	0.19	**0.37**	0.34	0.32
Scots Law	550	0.66	0.67	**0.68**	0.67	*Physical Geogr*	100	0.08	0.31	**0.36**	0.32
Aeronaut	277	0.46	**0.66**	0.65	0.65	*Ecol*	627	0.14	0.29	0.33	**0.35**
Roman Law	144	0.55	0.62	**0.66**	0.59	*Med*	1939	0.26	0.25	**0.34**	0.31
Law	2508	0.62	0.60	0.66	**0.66**	*Genetics*	120	0.08	**0.32**	0.31	0.27
Dentistry	77	0.54	0.64	0.64	**0.65**	*Brewing*	54	0.17	0.27	**0.32**	0.23
Math	1378	0.54	0.50	**0.65**	0.60	*Petrol*	64	0.04	**0.32**	0.32	0.31
Printing	446	0.55	0.63	**0.65**	0.63	*Rugby Football*	63	0.05	0.28	**0.32**	0.22
Billiards	86	0.44	**0.65**	0.63	0.49	*Nuclear Physics*	74	0.11	**0.31**	0.29	0.29
Fencing	117	0.54	**0.64**	0.60	0.56	*Cytology*	103	0.06	0.28	**0.30**	0.25
Archit	1100	0.53	0.57	**0.63**	0.60	*Fortif*	136	0.07	0.27	**0.28**	0.27
Geol	1226	0.53	0.55	**0.63**	0.59	*Optics*	129	0.18	0.24	**0.28**	0.25
Mining	854	0.60	0.55	**0.61**	0.57	*Typogr*	66	0.07	**0.28**	0.26	0.25
Cards	201	0.54	**0.61**	0.60	0.58	*Carpentry*	80	0.04	0.16	**0.27**	0.21
Astron	697	0.43	0.58	0.61	**0.61**	*Pottery*	47	0.05	0.24	**0.27**	0.19
Golf	154	0.35	**0.60**	0.59	0.50	*Football*	86	0.14	**0.27**	0.18	0.14
Farriery	112	0.31	0.40	**0.60**	0.49	*Geomorphol*	60	0.04	**0.26**	0.24	0.24
Astrol	267	0.49	0.57	**0.59**	0.57	*Shipbuilding*	84	0.00	0.22	**0.26**	0.19
Archaeol	135	0.30	0.53	**0.57**	0.56	*Hort*	108	0.11	**0.25**	0.19	0.23
Forestry	52	0.47	0.51	**0.57**	0.51	*Rom Antiq*	167	0.07	0.24	0.25	**0.25**
Angling	150	0.53	0.52	**0.57**	0.50	*Surg*	579	0.07	0.22	0.24	**0.24**
Falconry	117	0.43	**0.55**	0.50	0.48	*Palaeont*	189	0.02	0.22	0.22	**0.24**
Rhet	212	0.49	0.50	**0.55**	0.38	*Nat Hist*	460	0.02	0.16	0.23	**0.23**
Zool	2936	0.44	0.40	**0.55**	0.51	*Ecol Hist*	356	0.07	**0.23**	0.23	0.23
Ornith	474	0.40	0.51	0.54	**0.55**	*Electr Engin*	48	0.10	0.21	0.21	**0.23**
Entom	752	0.46	0.49	**0.54**	0.54	*Conch*	101	0.02	0.18	**0.21**	0.20
Anat	1718	0.41	0.46	**0.54**	0.54	*Mech*	624	0.18	0.20	**0.20**	0.18
Baseball	124	0.28	0.52	**0.54**	0.51	*Comm*	328	0.10	0.19	**0.19**	0.19
Physics	806	0.40	0.43	**0.54**	0.51	*Physiol*	693	0.07	**0.19**	0.18	0.19
Cryst	174	0.41	0.48	**0.54**	0.49	*Naval*	57	0.07	0.17	**0.19**	0.19
Ecol	91	0.35	0.51	**0.54**	0.49	*Art*	104	0.06	**0.18**	0.16	0.14
Philos	421	0.42	0.46	0.52	**0.53**	*Eng Hist*	99	0.01	**0.18**	0.15	0.17
Pharm	417	0.30	0.47	**0.53**	0.50	*Founding*	80	0.06	**0.17**	0.12	0.13
Meteorol	82	0.17	0.47	**0.50**	0.43	*Ichthyol*	175	0.02	0.13	**0.17**	0.11
Path	2156	0.45	0.45	**0.50**	0.49	*Metaph*	106	0.01	**0.15**	0.11	0.14
Phonetics	207	0.30	0.49	**0.50**	0.47	*Radio*	46	0.04	**0.14**	0.13	0.10
Electronics	235	0.39	0.48	0.49	**0.49**	*Gunnery*	55	0.06	0.11	**0.13**	0.08
Econ	115	0.31	0.48	**0.48**	0.45	*Building*	126	0.04	0.11	0.10	**0.12**
Biochem	480	0.20	0.46	**0.48**	0.45	*Stock Exchange*	54	0.08	**0.12**	0.12	0.10
Weaving	107	0.25	0.41	**0.48**	0.33	*Antiq*	256	0.03	**0.12**	0.10	0.10
Jazz	64	0.20	0.47	0.47	**0.48**	*Sport*	90	0.01	**0.11**	0.07	0.06
Arch	54	0.35	0.30	**0.47**	0.29	*RC Ch*	114	0.05	**0.11**	0.04	0.06
Metallurgy	163	0.25	**0.47**	0.44	0.46	*Civil Law*	50	0.00	0.09	**0.11**	0.10
Electr	530	0.31	0.41	**0.47**	0.46	*Racing*	97	0.03	**0.11**	0.09	0.10
Curling	58	0.34	0.37	**0.46**	0.40	*Politics*	61	0.00	0.07	0.10	**0.10**
Mil	1191	0.38	0.40	**0.46**	0.41	*Sporting*	86	0.00	0.09	**0.10**	0.05
Anthrop	47	0.18	**0.44**	0.35	0.38	*Acad*	297	0.00	0.02	0.03	**0.03**

Table 2. F-measures for all 128 categories

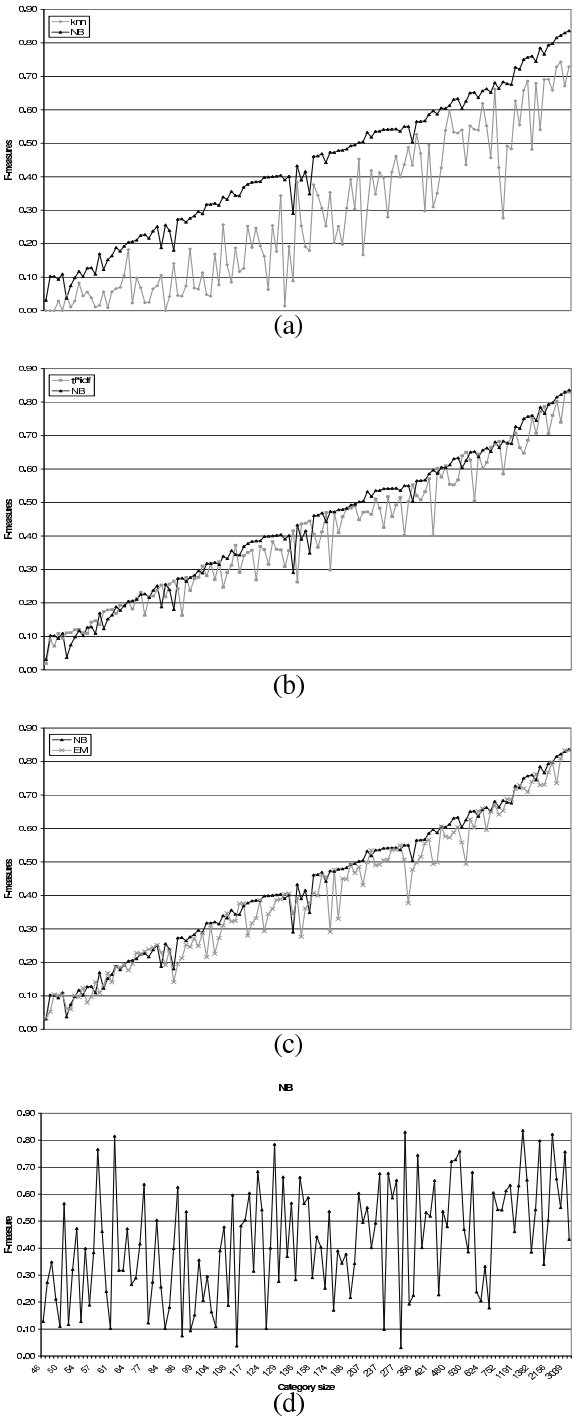


Figure 3. (a), (b) and (c) represent the comparisons of F-measures in Naive Bayes and the other three methods. (d) shows Naive Bayes performance *vs.* category size.

in some data sets, when there are a lot of labeled and a lot of unlabeled documents, this is not the case...and the incorporation of unlabeled data decreases, rather than increases, classification accuracy."[15] Nevertheless, perhaps a better probability model for our data could improve performance [14].

The original purpose of classifying OED entries by subject is to provide the editors of the dictionary with a tool that can help them to improve the labeling of subject-specific senses. As a next step, then, we plan to carry out experiments that involve the unlabeled data, using the 63,000 labeled senses for training. More specifically, we need to evaluate whether our thresholds to choose *not* to categorize a sense can serve to identify the general vocabulary that should remain unlabeled. Furthermore, we should also evaluate multi-labeling techniques (i.e. *n-per-doc* category assignment), so as to provide editors with several ranked choices for selecting the right subject label for a sense; this will also support the few instances where one sense really should be assigned several subject labels. Finally we wish to measure the performance of the approaches based on citations against other schemes to create taxonomies based on natural language processing of the definition text (or of the cited quotation itself) [6]. We may also find that combining evidence from citations with evidence from the definitions yields even better results.

Acknowledgments

We are grateful to Dale Schuurmans, Reem Al-Halimi, Charlie Clarke, and Fuchun Peng for their many insights into approaches to addressing this research problem. Financial support was provided by grants from Bell Canada through Bell University Labs, the MITACS Centre of Excellence, the Natural Sciences and Engineering Research Council of Canada, and the University of Waterloo.

References

[1] R. BaezaYates and B. RibeiroNeto. *Modern Information Retrievel*. Addison-Wesley, ACM Press, 1999.

[2] D. L. Berg. The research potential of the electronic OED2 database at the University of Waterloo: A guide for scholars. Technical report, UW Centre for the New Oxford English Dictionary, 1989.

[3] D. L. Berg. *A Guide To The Oxford English Dictionary*. Oxford University Press, 1993.

[4] S. Brin. Extracting patterns and relations from the World Wide Web. In *Proceedings of WebDB Workshop at EDBT'98*, Valencia, Spain, March 1998.

[5] W. A. Gale. Good-Turing smoothing without tears. Technical report, AT&T Bell Laboratories, 1994.

[6] N. Ide and J. Véronis. Knowledge extraction from machine-readable dictionaries: An evaluation. In *Machine Translation and the Lexicon*, pages 19–34. Springer-Verlag, 1994.

[7] M. Iwayama and T. Tokunaga. Cluster-based text categorization: A comparison of category search strategies. In *Proceedings of ACM SIGIR Conference (SIGIR)*, pages 273–280, Seattle, WA USA, 1995.

[8] T. Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *Proceedings of the 14th International Conference on Machine Learning ICML97*, 1997.

[9] R. Korfhage. *Information Storage and Retrieval*. Wiley Computer Publication, 1997.

[10] Z. Langari. Subject classifications in the Oxford English Dictionary. Master's thesis, University of Waterloo, Department of Computer Science, 2001.

[11] D. Lewis. Evaluating text categorization. In *Proceedings of speech and Natural Language Workshop*, pages 312–318, Asilomar, Feb. 1991.

[12] D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 37–50, 1992.

[13] A. McCallum and K. Nigam. A comparison of event models for naive Bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[14] A. McCallum and K. Nigam. Text classification by bootstrapping with keywords, EM and shrinkage. In *ACL '99 Workshop for Unsupervised Learning in Natural Language Processing*, 1999.

[15] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

[16] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[17] F. Sebastiani. Machine learning in automated text categorization. Technical report, Istituto di Elaborazione dell'Informazione, 1999.

[18] J. A. Simpson and E. S. C. Weiner, editors. *The Oxford English Dictionary*. Oxford University Press, second edition, 1989.

[19] Y. Yang. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1(1/2):67–88, 1999.

[20] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 42–49, 1999.