Document Classification: Out Of Place vs. Cosine Distance Metrics

Dylan Visher

Abstract—**In this paper, we compare two different n-gram based methods for supervised content-based document classification. The two methods differ primarily in their distance metric. We compare the "out-of-place" distance metric proposed by Cavnar[?] with the cosine distance metric proposed by Damashek.[?] We will show that the cosine measure is a more accurate supervised, content-based n-gram distance measure for the categorization of English texts into literary certain genres. This paper shows that Damashek's treatment of documents as multidimensional vectors yields greater accuracy and precision than that of Cavnar's more naive, but still effective, distance metric. We also show that Damashek's is much better (relative to Cavnar) when trained on a sparse dataset.**

I. Introduction

Document Classification is a fundamental task in document processing, and appears in many places in our lives such as spam filtering, language identification, genre classification, etc. With the rise of the Internet and the proliferation of unclassified texts in the form of articles, blogs, etc. It is important to be able to classify documents into a genre for search filtering or for finding 'similar' texts. Thus, it is extremely important to be able to determine whether a document is of one category like romance or if it is of another category religion as these two categories could draw very different reader groups. Also, document classification necessitates high accuracy, because misclassifications of genres like romance novels as religious texts would probably lead to that romance novel going undiscovered by its intended audience.

In this paper, we compare two different supervised, content-based document classification schemes that compare n-gram profiles for texts against "average" n-gram profiles for a genre. We look at the "out-of-place" distance metric Cavnar proposes[?] as well as the cosine distance metric Damashek proposes.[?] We will analyze the differences in their methodology, and the differences in their results. We will compare these two methods using the Brown Corpus as defined by the NLTK Python package. Ultimately we show that, though they are comparable in terms of accuracy, Damashek's accuracy is on average higher. In addition, though they are similar when training on a large training set, with a reduced training set, Damashek proves to be a much more accurate method.

II. Research Question

This paper attempts to compare of the pros and cons of an "out-of-place" distance metric as defined by Cavnar in his paper[?] with the pros and cons of a "cosine-distance" metric as defined by Damashek in his paper.[?] Specifically we will see whether the cosine measure or the out-of-place measure provides a more useful character-based n-gram document classification distance metric for categorizing texts belonging to a certain genre. We will also show that Damashek improves in accuracy, relative to Cavnar, when trained on a shorter dataset.

III. Methodology

A. Cavnar

1) Overview: For Cavnar we create an n-gram profile for each genre and for each document we are attempting to classify. These n-gram profiles map n-grams of length 1 to 5 with their associated frequencies (generated from all texts of the same genre in the training set or from the text that is being classified). A text is classified by its n-gram profile being compared with each genre's n-gram profile using the "out-of-place" distance metric. The closest genre to the sample text is chosen to be that text's genre.

2) N-Gram Creation: To create the n-grams from a string (the concatenated string of multiple files, or the string that defines a file itself), we divide the string into individual words. Cavnar in his paper suggests padding the string with a single leading space, and pad the end of the word such that the each n-gram, of any size can be of just the final character and the following spaces. For this paper, however, we depart from this exact method because it gives an unjustified increased weight to the end of words. To solve this problem, we pad the beginning of words such that each n-gram, or any size can be of just the starting character preceded by spaces. This means that the n-gram profile generated by each word is symmetric, giving an equal weighting to both the start and end of the word. When tested with both options, this change significantly increases the performance of the Cavnar method.

3) Out-Of-Place Distance Metric: The out of place distance metric compares the n-gram profile of the sample text to be categorized and the n-gram profile of the template text (both sorted in decreasing order of absolute frequency). Cavnar chooses to ignore the first 300 n-grams because they seem to indicate language more than features. In fact, when we tested with and without the first 300 n-grams, we found that including the first 300 actually decreases performance, probably because of the noise they generate is greater than their value to profiling genre. After the two texts are aligned based off frequency, and the leading 300 n-grams are discarded, the metric works by summing the absolute value of the differences of the indices for each key in the profile. This is essentially saying, how far out of place from the template profile (what we know to be the true average) is each n-gram generated by sample text.

Cavnar does not explicitly address the problem of ngrams that are in the sample, but not in the template. To deal with this issue, when n-grams not in template we assume they were at the very end of the other frequency vector. Thus, we add to the distance the length of the template profile, as if they were about to be found.

4) Catagorizing a Text:

```
function CavnarDistance(template, sample map[string]double) double
    tsorted := Sort(template, by=value, reverse=True)
    ssorted := Sort(sample, by=value, reverse=True)
    for k in sample not in template do
        tsorted.append((k, 0))
    distance := 0
    for i := range ssorted do
        j := find(ssorted[i], tsorted)
        distance += abs(i-j)
    return distance
function GenerateNGrams(category string) (ngrams map[string]double)
    ngrams = nil
    words := tokenize(string)
    for w := range words do
        // Pad the word on either side
        word = " "*len( LongestNgram - 1 )+word+" "*len( LongestNgram - 1)
        stringsoffset := make([]string)
        wordgrams := All Ngrams in word from SmallestNgram to LongestNgram Inclusive
        for g in wordgrams do
            if g in ngrams then
                ngrams[g]++
            else
                ngrams[g] = 0
    return ngrams
function CategorizeText(string smp)
    // Generate Profiles for Each Genre
    Templates := map[string]double
    for c := range categories do
        TemplateProfiles[c] := GenerateNGrams(Append(AllTextsIn(c)))
    sample := GenerateNGrams[smp]
    minsofar := infinity
    category string := "None"
    for c := range categories do
        d := CavnarDistance(TemplateProfiles[c], sample)
        if d < minsofar then
            minsofar := d
            category := c
    return category
```

B. Damashek

1) Overview: For the purpose of being concise we will be focusing on the differences between Damashek and Cavnar. The main difference is how the distance metric is computed. Damashek looks at the cosine distance between the two $absolute$ frequency profiles of the template and the sample. Damashek sees the n-gram profiles as vectors in multidimensional space.

2) N-Gram Creation: Unlike Cavnar, Damashek only looks at n-grams of length 5 (vs 1-5). Damashek also does not pad the word with spaces on either side, as n-grams of smaller length are more indicative of language, not of any higher level categories (which we are trying to do).

3) Cosine Distance Metric: Damashek looks at the template and sample profiles as vectors in n dimensional space. Thus he normalizes the template and sample to have the same n-gram keys (defaulting ones not in there to 0). Then he turns the relative frequency vectors into absolute frequency vectors, sorted in key order (thus projecting them onto the same n dimensional hyperplane). Then he compares the two by using a normalized cosine distance.

4) Catagorizing a Text:

```
function DamashekDistance(template, sample map[string]double) double
    tsorted := Sort(template, by=key)
    ssorted := Sort(sample, by=key)
    for k in sample not in template do
        tsorted.append((k, 0))
    for k in template not in sample do
        tsorted.append((k, 0))
    tfreqs := [tsorted[1]/sum(tsorted)]
    sfreqs := [ssorted[1]/sum(ssorted)]
    mu1 := len(tfreqs)
    mu2 := len(sfreqs)
    numerator := sum([(i-mu1) *(j - mu2) for i in tfreqs for j in sfreqs])
    xxs := sum([(x-mu1) ** 2 for x in tfreqs])
    yys := sum([(y-mu2) **2 for y in sfreqs])
    return numerator / (sqrt(xxs * yys)
function GenerateNGrams(category string) (ngrams map[string]double)
    ngrams = nil
    words := tokenize(string)
    for w := range words do
        // Pad the word on either side
        stringsoffset := make([]string) wordgrams := All grams of size 5 in w
        for g in wordgrams do
            if g in ngrams then
                ngrams[g]++
            else
                ngrams[g] = 0
    return ngrams
function CategorizeText(string smp)
    // Generate Profiles for Each Genre
    Templates := map[string]double
    for c := range categories do
        TemplateProfiles[c] := GenerateNGrams(Append(AllTextsIn(c)))
    sample := GenerateNGrams[smp]
    minsofar := infinity
```

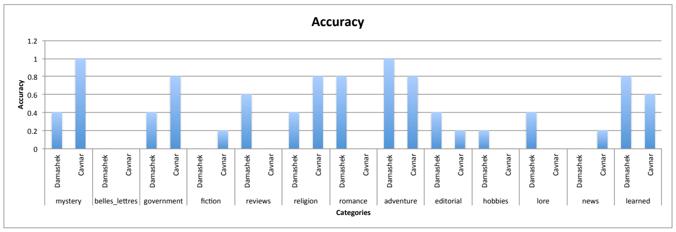

Fig. 1. Accuracy/Recall of Damashek vs. Cavnar By Category

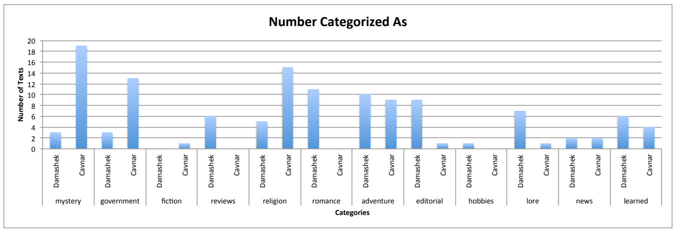

Fig. 2. Number Categorized As Genre Belles-Lettres Removed

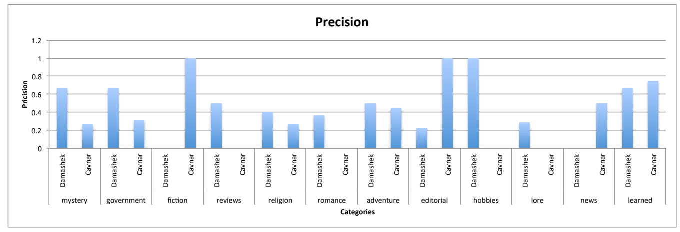

Fig. 3. Precision of Damashek vs. Cavnar Belles-Lettres Removed

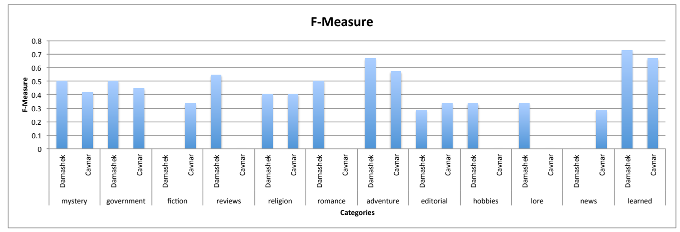

Fig. 4. F-Measure of Damashek vs. Cavnar Belles-Lettres Removed

```
category string := "None"
for c := range categories do
    d := DamashekDistance(TemplateProfiles[c], sample)
    if d < minsofar then
        minsofar := d
        category := c
return category
```

IV. Results

In this experiment we trained both Cavnar and Damashek's algorithm on the first 10 documents in each category of python's nltk Brown Corpus[1]. We then tested both Cavnar and Damashek's algorithm on next 5 documents. We then compared the algorithm's outputs against the know category of each text to find the Accuracy, Precision, Recall, and F-measure of the algorithm for each category. On the given test set the overall accuracy of Cavnar is .354 and the overall accuracy of Damashek is .415.

In the genre by genre analysis of the accuracies (also known as the recall) both methods performs similarly. Cavnar performs significantly better classifying mysteries and religion and government. Whereas Damashek performs better categorizing reviews, romance, adventure, lore, and learned. Both techniques have categories where they excel and both have categories where they fail completely. Some categories like belles lettres, fiction, hobbies, and news seem to be pathological cases for both techniques with accuracy rates of 20% or less (essentially random). Both methods perform worse than random on classifying belles lettres. Since no text was classified as belles lettres, the category is removed or the following figures.

Figure 2 shows the number of texts that were categorized as being in a certain category. In many categories there is dramatic over classification (mystery, government, religion and adventure for Cavnar). In the ideal case, only 5 texts would be categorized into each genre, since the test set only contains 5 texts in each genre. Damashek experiences much less over categorization as adventure, romance, and editorial are over classified containing only 2 times the ideal, whereas Cavnar's mystery category is almost 4 times larger than the ideal.

Figure 3 shows the precisions of the two methods. Both methods experience high precision only on the categories that have low accuracy. This indicates that higher accuracy is also accompanied with over classification. The categories with the highest accuracy for Damashek have a precision of about 50%.

F-Measure is a measure for a more true accuracy which takes into account both recall and precision. This makes the two approaches much more consistent with F-Measure accuracy levels mostly hovering around 50% except for in the pathological categories. Cavnar has more pathological categories than Damashek (reviews, romance, hobbies, and lore vs. fiction and news) but even ignoring these categories Damashek does consistently better than Cavnar (with the exception of editorial).

V. Discussion

When looking at the recall for both methods (Figure 1), both methods are classify categories with well over 50 percent accuracy. In his paper, Cavnar boasted an accuracy of greater than 99% in most cases for language detection. However, the task of classifying documents written in the same language into different genres is inherently much more difficult. Genre's necessitate inner language differentiation, and therefore more specific In addition, we notice that there are a few pathological genres that are responsible for most of the loss of accuracy. Upon analysis of the nature of these categories it becomes apparent why our methods failed.

Belles Lettres is by far the worst genre for classification is defined as fine or beautiful writing and is a genre that encompasses all other genres, poetry, fiction, drama, even

[1]All categories in the brown corpus were used other than science fiction and humor

essays, the only requisites being that they are ”beautiful”. A writing’s beauty has nothing to do with the words that an author uses. This is a classic case of humans propensity for fuzzy logic and multiple classifications messing up the discrete logic of computers and classifications. It is unclear whether a ”beautiful” fiction piece be classified as belles lettres or fiction, in fact it would be best for it to be classified as both. Similarly fiction could be classified more specifically as lore, adventure, some might even say religion should fall under that category. The boundaries between genres is not nearly as clear as classifying a text as french or english. As a text often can have many genres, but most of the time it only has one language.

This reasoning is extended further when viewing other genres such as lore, adventure, romance, and mystery. All of these genres are specifically a subset of fiction. It would be equally valid to classify a romantic fiction document as either romance or fiction. It is evident from the over classification of romance and adventure that Damashek’s method sees many fiction categories as romance or adventure specifically. This is probably because works in fiction will tend to bias in a certain direction (romance, adventure, lore, ...) rather than being the perfectly average fiction.

Hobbies are often also miscategorized often as adventure or editorial. This is because many write about hobbies in editorials and many people have hobbies that involve adventure. People are prone to describing their hobbies with more adventurous vocabulary. Writings about hobbies are a subset of adventure and editorial, it is no surprise that they are miscategorized as such.

The news genre poses an entirely new problem. Not only can the government, reviews, or editorial genre be classified as news, but news also naturally assumes the language of whatever it is describing. Its vocabulary can change dramatically with depending on subject. If writing about the current crisis in Ukraine words like ’Russia’, ’Ukraine’, ’Crimea’, ’USSR’, and many other associated words will occur with a very high frequency, whereas on average these words will fall into the noise of celebrity gossip, elections, financial crises, etc. This inconsistency leads to the template vector (the one created during training on the news corpus) to be very sparse and to not contain much of the vocabulary that is in the sample articles.

Figure 2 and 3, the number categorized in each genre and the precisions, indicate that both these methods higher accuracy is also coupled with higher over classification. But Damashek tends to have much lower over classification coupled with its high accuracy.

Upon removal of the pathological, umbrella categories (belles lettres, news, fiction, hobbies) the overall accuracy of the methods improve dramatically with Cavnar attaining 49% Accuracy and Damashek 62% accuracy. This improvement indicates that much of the miscategorization of the methods (especially Damashek) has to do with Damashek’s propensity to favor the more specific. All of these pathological umbrella categories can be categorized as a more specific category as most texts are not average but rather they are nuanced, leaning in multiple directions.

If we want to improve Damashek further, we can incorporate Cavnar’s approach for looking at n-grams of varying length. If Damashek uses n-grams of length 4-5 (keeping the pathological genres in the dataset) it can attain an accuracy of 45.7%, a significant improvement over the 41.4% accuracy without these improvements.

When Damashek is run on n-grams of length 4 and 5 rather than just 5 and trains on only half the items (but still all the genres), it is still able to achieve the 41.5% accuracy rate of the algorithm with fixed length accuracy. Thus Damashek proves to be superior when using a sparse training set. This probably has to do with the cosine distance inherently penalizing less than the out of order distance metric for words that were not in the template profile.

Ultimately Damashek offers a distance metric with lower risk for over-classification and pathological categories, which excels at sparse training sets. Damashek’s method, however, always favors the more specific categorization whereas Cavnar’s allows more frequently a more general categorization.

VI. Conclusion

Both techniques are comparable with each other achieving similar results on this dataset. Damashek, however, is far less susceptible to pathological categories and gross over categorization. Damashek makes much more modest over classifications than Cavnar because the cosine distances of word vectors give a much more specific analysis of categories (though this analysis does not always align with human analysis).