Allison Obourn
CSE 573
12/9/12

Final Project – Displaying Unbiased Information in the Living Voters Guide

Goals

The Living Voters Guide is an online supplement to the standard voters guide. It provides detailed information about initiatives and encourages deliberation. Users are prompted to create a list of pros and cons about each initiative. The website provides a list, a couple points long, comprised of points other users have contributed, to help inspire users and inform them about considerations they may not have thought of. It is critical that these lists of points are non-biased so that the system will not be biased and will be trusted.

The eventual goal of my work is to be able to automatically choose a set of points to display that is both non-biased and topically diverse. However, that is too big a task to accomplish for this project. My smaller goal for this project is to use NLP to create clusters of topically similar points.

System Design and Algorithm Choices

My system uses Tf-idf to profile each of the points. It then uses cosine distance to find the distance between them. I chose Tf-idf because I wanted to see how a relatively simple bag of words technique would work with different versions of the same input text. I chose cosine distance because it works much better than linear distance and, from the reading I have done, appears to be a standard distance metric to use.

I tried altering the input text that I gave to Tf-idf in several different ways. I tried all combinations of removing stopwords, selecting nouns and verbs, selecting only nouns and selecting only verbs with including synonyms and not including synonyms. The synonyms were incorporated by altering Tf-idf to add an occurrence vote if there was a synonym of the looked for word, not just if the exact word was seen.

Interfaces

I do not have any usage screenshots as my project did not involve a front end or user interaction element.

I have included the code for a manual clustering interface that I wrote when it appeared that I would need it for evaluation. I did not evaluate it or include screenshots of it, as it is not for public use, it was simply built for my evaluation.

Experiments

Which text version of the points leads to clusterings closest to those a human would produce?

I used the B^3 metric to evaluate my output. B^3 generates two measurements, precision and recall. Precision is a measure of how many points are in a cluster with another point that they shouldn't be. It is the sum of the weight of each point times that point's precision. That point's precision is the number of other points that should end up in a cluster with that point and were clustered together with it in the output divided by the number of elements in the output cluster. Recall is a measure of how many times points that are supposed to end up in the same cluster end up together. It is the sum of the weight of each point times that point's recall. That point's recall is the number of other points that should end up in a cluster with that point and were clustered together with that point in the output divided by the number of elements that should have been clustered together with it. I weighted all points equally.

In order to compare against where points should go we need a ground truth. I used a clustered version of the points generated by a human for this. There is no single correct answer to clustering points; there are multiple good ways to cluster a group of points. There can be big differences in clusters between multiple people clustering the same dataset. Although a much larger number of different human clusterings would be needed to get meaningful data, I ran B^3 over two different human annotations to get a feeling for their agreement. I found they had precision of 0.65458 and recall of 0.69855. Both annotations were perfectly valid clusterings.

The results of the test of random are the averages of the scores over 1000 different random clusterings.

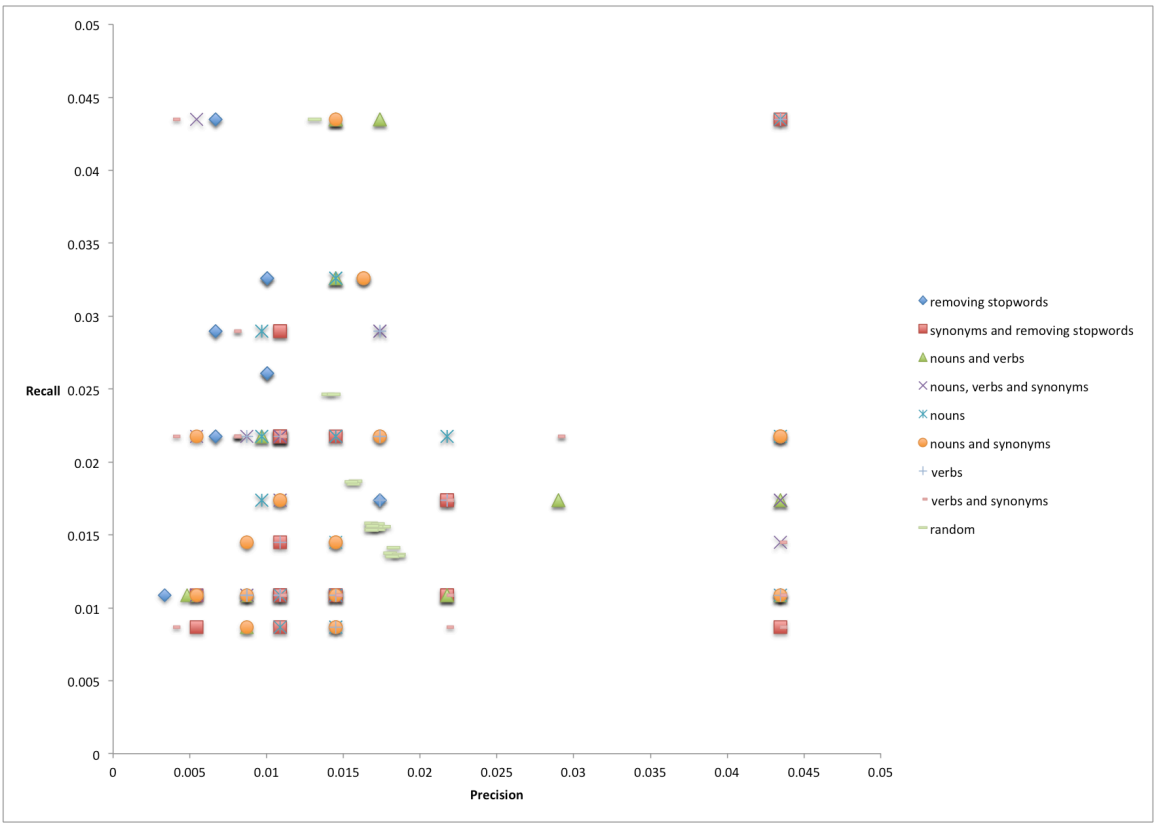

Figure 1 – Scores for each point in each type of input text

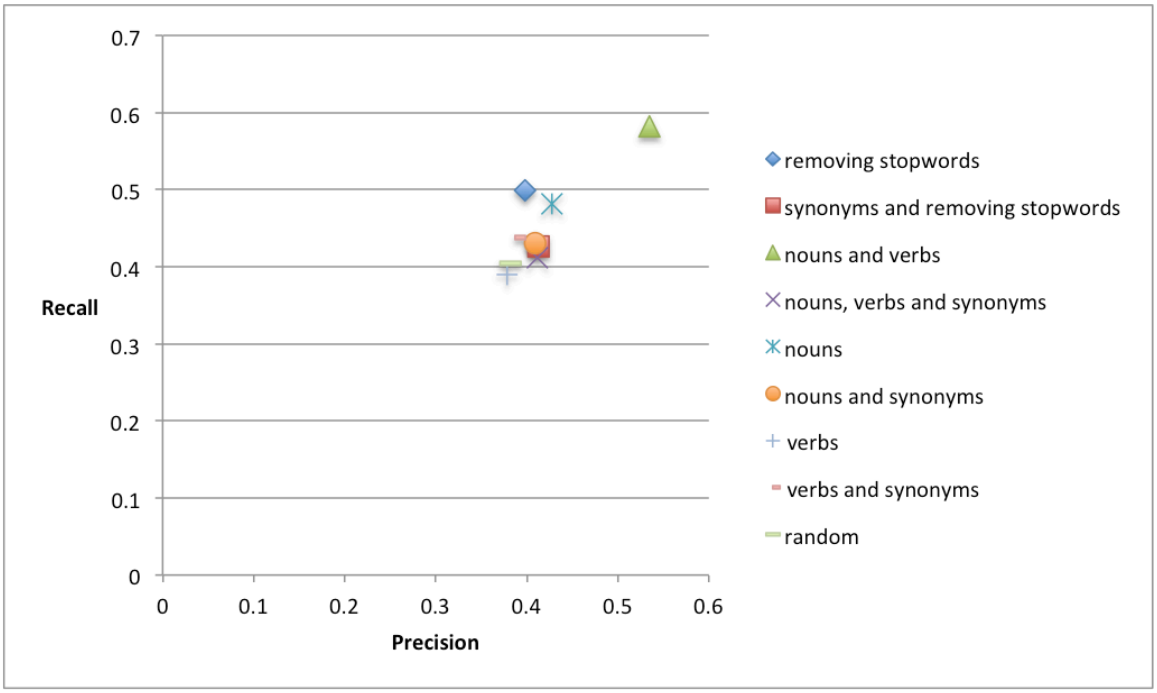

Figure 2 – Total score of all points combined per method.

Conclusions and Future Work

My results show that clustering only considering the nouns and verbs and not taking synonyms into account is the most accurate of the options I tried. I was very surprised by this result. It does not match my hypothesis that considering synonyms would improve accuracy. I think the reason for this is that the synonym lists that I was using do not have the domain specific terminology that is needed. I used the NLTK synonyms, which are very general. Creating a corpus of domain specific words and retesting this is left as future work. I think that, if it was done and combined with just using nouns and verbs, it would boost result accuracy.

I was not surprised to find that considering nouns and verbs together performed better than eliminating stopwords. Nouns and verbs form the core meaning of sentences, the other words add specificity, which is unnecessary here. I was surprised to find that just selecting nouns performed worse than nouns and verbs. Since I was looking for topical diversity I thought selecting only nouns would be the best way to only select topics.

Although the precision and recall of all my methods are better than that of random clustering they are not significantly better. I think nouns, verbs and no synonyms is the only one that is better. I tried to pick a topic label for each cluster my algorithms generated but I had a difficult time. Using nouns and verbs, I could pick a few labels. All of the other methods generated completely random looking clusters so I couldn't pick any I felt were correct. One factor that probably contributes to the poorness of the clustering is the length of the points. They are no more than 5 sentences long and most are much shorter. With such a short length, there aren't many words to cluster upon. The political jargon that was not caught by the synonym comparison is probably another reason. A third big reason may be spelling. Another future improvement could be to add automatic spelling correction.

The B^3 score of nouns, verbs and no synonyms is almost halfway between the B^3 scores of random clustering and human clustering. Although it does not generate good enough clusters for my application, hopefully with the future work described above, it can be made to.

Appendices:

Outside Code

I used NLTK to assist with all of my code. I used its implementation of Tf-idf and extended the same version of Tf-idf to create a version that takes synonyms into consideration. I also used it for synonym lists, stopword lists, a POS tagger and clustering.

Use Instructions

The readme with instructions on how to run the program is located in the zip of the code in README.txt.

The main program requires NLTK to run and is located in syns.py. It takes 3 arguments. The first is the mode to run it in. There are nine options.

1: no alteration to the input data except minimal stopword filtering
2: synonyms added to the input data and minimal stopword filtering
3: only verbs and nouns considered
4: synonyms added and only nouns and verbs considered
5: only nouns considered
6: synonyms added and only nouns considered
7: only verbs considered
8: synonyms added and only verbs considered
9: 1000 random clusterings
10: compares two human generated clusterings

It also takes the names of two files. The first is the input file, which contains the points, and the second is the output file where it outputs the matrix. When the program runs it will print out the precision and recall for each point.

A working version of the web interface can be found at:
http://homes.cs.washington.edu/~aeobourn/nlp/cluster.php