Assignment 7 WRITEUP.pdf

Ning Jiang

Does your program accurately identify the author for a small passage of text? What about a large passage of text? How do the different metrics (Euclidean, Manhattan, Cosine) compare against each other?

My program can accurately identify the author for a small passage text without any error.

Example: 1) Euclidean

```
njiang21@janny:~/cse13s/asgn7$ ./identify -d small.db -k 10 <texts/thomas-carlyle.txt
Top 10, metric: Euclidean distance, noise limit: 100
1) Thomas Carlyle [0.000000000000000]
2) Joseph Conrad [0.017166390487124]
3) Henry van Dyke [0.017738823972039]
4) H. G. Wells [0.018159389259155]
5) Henry James [0.018169885641644]
6) Henry Fielding [0.018863008966098]
7) Daniel Defoe [0.020333254684404]
8) Saxo Grammaticus [0.020424119118040]
9) Unknown [0.033337552670497]
10) Anonymous [0.078041085551897]
```

```
njiang21@janny:~/cse13s/resources/asgn7$ ./identify -d small.db -k 10 <t
exts/thomas-carlyle.txt
Top 10, metric: Euclidean distance, noise limit: 100
1) Thomas Carlyle [0.000000000000000]
2) Joseph Conrad [0.017166390487124]
3) Henry van Dyke [0.017738823972039]
4) H. G. Wells [0.018159389259155]
5) Henry James [0.018169885641644]
6) Henry Fielding [0.018863008966098]
7) Daniel Defoe [0.020333254684404]
8) Saxo Grammaticus [0.020424119118040]
9) Unknown [0.033337552670497]
10) Anonymous [0.078041085551897]
```

Example: 2) Manhattan

```
njiang21@janny:~/cse13s/asgn7$ ./identify -m -d small.db -k 10 <texts/henry-fielding.txt
Top 10, metric: Manhattan distance, noise limit: 100
1) Henry Fielding [0.000000000000000]
2) Daniel Defoe [0.931158367432455]
3) Thomas Carlyle [1.014699756233058]
4) Joseph Conrad [1.050932908542546]
5) H. G. Wells [1.085325934444574]
6) Henry James [1.096356995548563]
7) Saxo Grammaticus [1.097361520011068]
8) Henry van Dyke [1.107954523573068]
9) Unknown [1.281315067690451]
10) Anonymous [1.799725241728353]
```

```
njiang21@janny:~/cse13s/resources/asgn7$ ./identify -m -d small.db -k 10
 <texts/henry-fielding.txt
Top 10, metric: Manhattan distance, noise limit: 100
1) Henry Fielding [0.000000000000000]
2) Daniel Defoe [0.9311583674324 55]
3) Thomas Carlyle [1.0146997562330 58]
4) Joseph Conrad [1.0509329085425 46]
5) H. G. Wells [1.0853259344445 74]
6) Henry James [1.0963569955485 63]
7) Saxo Grammaticus [1.0973615200110 68]
8) Henry van Dyke [1.1079545235730 68]
9) Unknown [1.2813150676904 51]
10) Anonymous [1.7997252417283 53]
```

Example: 3) Cosine

```
njiang21@janny:~/cse13s/asgn7$ ./identify -c -d small.db -k 10 <texts/henry-van-dyke.txt
Top 10, metric: Cosine distance, noise limit: 100
1) Henry van Dyke [0.999409129341257]
2) Daniel Defoe [0.999517113427802]
3) H. G. Wells [0.999549924026520]
4) Joseph Conrad [0.999561319593377]
5) Henry James [0.999573813218979]
6) Henry Fielding [0.999576807013006]
7) Unknown [0.999585372887743]
8) Thomas Carlyle [0.999590996912274]
9) Saxo Grammaticus [0.999645817311929]
10) Anonymous [0.999891762831797]
```

```
njiang21@janny:~/cse13s/resources/asgn7$ ./identify -c -d small.db -k 10
 <texts/henry-van-dyke.txt
Top 10, metric: Cosine distance, noise limit: 100
1) Henry van Dyke [0.999409129341257]
2) Daniel Defoe [0.999517113427802]
3) H. G. Wells [0.999549924026520]
4) Joseph Conrad [0.999561319593377]
5) Henry James [0.999573813218979]
6) Henry Fielding [0.999576807013006]
7) Unknown [0.999585372887743]
8) Thomas Carlyle [0.999590996912274]
9) Saxo Grammaticus [0.999645817311929]
10) Anonymous [0.999891762831797]
```

My program can accurately identify the author for a medium passage text without any error. Takes about one minutes and 20 seconds.

Example: 1) Euclidean

```
njiang21@janny:~/cse13s/asgn7$ ./identify -d medium.db -k 10 <texts/henry-james.txt
Top 10, metric: Euclidean distance, noise limit: 100
1) Henry James [0.000000000000000]
2) Various [0.015940282434850]
3) Joseph Conrad [0.017647495633110]
4) Thomas Carlyle [0.018169885641648]
5) Henry van Dyke [0.018188000064887]
6) Kate Douglas Wiggin [0.019289204823720]
7) H. G. Wells [0.019709696125277]
8) Charles Dickens [0.019997635145445]
9) Robert Louis Stevenson and Lloyd Osbourne [0.020825962859932]
10) Henry Fielding [0.021112901136087]
```

```
njiang21@janny:~/cse13s/resources/asgn7$ ./identify -d medium.db -k 10 <
texts/henry-james.txt
Top 10, metric: Euclidean distance, noise limit: 100
1) Henry James [0.000000000000000]
2) Various [0.015940282434850]
3) Joseph Conrad [0.017647495633110]
4) Thomas Carlyle [0.018169885641648]
5) Henry van Dyke [0.018188000064887]
6) Kate Douglas Wiggin [0.019289204823720]
7) H. G. Wells [0.019709696125277]
8) Charles Dickens [0.019997635145445]
9) Robert Louis Stevenson and Lloyd Osbourne [0.020825962859932]
10) Henry Fielding [0.021112901136087]
```

Example: 2) Manhattan

```
njiang21@janny:~/cse13s/asgn7$ ./identify -m -d medium.db -k 10 <texts/ernest-thompson-seton.txt
Top 10, metric: Manhattan distance, noise limit: 100
1) Ernest Thompson Seton [0.000000000000000]
2) Jack London [0.893735008049811]
3) Zane Grey [0.910053345817283]
4) Various [0.925495919629609]
5) Joseph A. Altsheler [0.937463372863063]
6) Henry van Dyke [0.953656482756709]
7) Francis Parkman, Jr. [0.965351064688882]
8) Joseph Conrad [0.986804436292095]
9) Stewart Edward White [0.994024606654091]
10) B. M. Bower [1.010355791499966]
```

```
njiang21@janny:~/cse13s/resources/asgn7$ ./identify -m -d medium.db -k 1
0 <texts/ernest-thompson-seton.txt
Top 10, metric: Manhattan distance, noise limit: 100
1) Ernest Thompson Seton [0.000000000000000]
2) Jack London [0.893735008049811]
3) Zane Grey [0.910053345817283]
4) Various [0.925495919629609]
5) Joseph A. Altsheler [0.937463372863063]
6) Henry van Dyke [0.953656482756709]
7) Francis Parkman, Jr. [0.965351064688882]
8) Joseph Conrad [0.986804436292095]
9) Stewart Edward White [0.994024606654091]
10) B. M. Bower [1.010355791499966]
```

Example:
3)Cosine

```
njiang21@janny:~/cse13s/asgn7$ ./identify -c -d medium.db -k 10 <texts/robert-louis-stevenson-an
d-lloyd-osbourne.txt
Top 10, metric: Cosine distance, noise limit: 100
1) Robert Louis Stevenson and Lloyd Osbourne [0.999332537358980]
2) Edwin Arlington Robinson [0.999386865562879]
3) Booth Tarkington [0.999408472425765]
4) Henry Lawson [0.999429268962652]
5) J. M. Synge [0.999448390530159]
6) Charles Dickens [0.999460958228609]
7) Oscar Wilde [0.999506474049548]
8) B. M. Bower [0.999509821982014]
9) The American Tract Society [0.999512821368626]
10) Robert W. Service [0.999513491477593]
njiang21@janny:~/cse13s/asgn7$
```

```
njiang21@janny:~/cse13s/resources/asgn7$ ./identify -c -d medium.db -k 1
0 <texts/robert-louis-stevenson-and-lloyd-osbourne.txt
Top 10, metric: Cosine distance, noise limit: 100
1) Robert Louis Stevenson and Lloyd Osbourne [0.999332537358980]
2) Edwin Arlington Robinson [0.999386865562879]
3) Booth Tarkington [0.999408472425765]
4) Henry Lawson [0.999429268962652]
5) J. M. Synge [0.999448390530159]
6) Charles Dickens [0.999460958228609]
7) Oscar Wilde [0.999506474049548]
8) B. M. Bower [0.999509821982014]
9) The American Tract Society [0.999512821368626]
10) Robert W. Service [0.999513491477593]
```

My program can accurately identify the author for a large passage text without any error. Take about four minutes.

Example: 1) Euclidean

```
njiang21@janny:~/cse13s/asgn7$ ./identify -k 10 <texts/christopher-marlowe.txt
Top 10, metric: Euclidean distance, noise limit: 100
1) Christopher Marlowe [0.000000000000000]
2) William Shakespeare [0.036663891718514]
3) Dante Alighieri [0.038187354084974]
4) William D. McClintock [0.038603095623566]
5) Edgar Allan Poe [0.038677849554845]
6) Lew Wallace [0.039963951144447]
7) Johann Wolfgang von Goethe [0.040020525022406]
8) George Rawlinson [0.040156479668630]
9) Henry Timrod [0.040196656007310]
10) John Fiske [0.040264038199767]
```

```
njiang21@janny:~/cse13s/resources/asgn7$ ./identify -k 10 <texts/christo
pher-marlowe.txt
Top 10, metric: Euclidean distance, noise limit: 100
1) Christopher Marlowe [0.000000000000000]
2) William Shakespeare [0.036663891718514]
3) Dante Alighieri [0.038187354084974]
4) William D. McClintock [0.038603095623566]
5) Edgar Allan Poe [0.038677849554845]
6) Lew Wallace [0.039963951144447]
7) Johann Wolfgang von Goethe [0.040020525022406]
8) George Rawlinson [0.040156479668630]
9) Henry Timrod [0.040196656007310]
10) John Fiske [0.040264038199767]
```

Example: 2) Manhattan

```
njiang21@janny:~/cse13s/asgn7$ ./identify -m -k 10 <texts/nathaniel-h.-bishop.txt
Top 10, metric: Manhattan distance, noise limit: 100
1) Nathaniel H. Bishop [0.000000000000000]
2) Various [1.066843996488571]
3) David Livingstone [1.070318936173724]
4) Richard Henry Dana [1.086042553357048]
5) Jules Verne [1.086099923299065]
6) James Fenimore Cooper [1.100278891125577]
7) Ida Pfeiffer [1.102345006320986]
8) Henry van Dyke [1.102530328285066]
9) Thomas Bailey Aldrich [1.105153564447951]
10) Samuel Smiles [1.116303213080148]
```

```
njiang21@janny:~/cse13s/resources/asgn7$ ./identify -m -k 10 <texts/nath
aniel-h.-bishop.txt
Top 10, metric: Manhattan distance, noise limit: 100
1) Nathaniel H. Bishop [0.000000000000000]
2) Various [1.066843996488571]
3) David Livingstone [1.070318936173724]
4) Richard Henry Dana [1.086042553357048]
5) Jules Verne [1.086099923299065]
6) James Fenimore Cooper [1.100278891125577]
7) Ida Pfeiffer [1.102345006320986]
8) Henry van Dyke [1.102530328285066]
9) Thomas Bailey Aldrich [1.105153564447951]
10) Samuel Smiles [1.116303213080148]
```

Example: 3)Cosine **takes longer!**

```
njiang21@janny:~/cse13s/asgn7$ ./identify -c -k 10 <texts/william-shakespeare.txt
Top 10, metric: Cosine distance, noise limit: 100
1) Elizabeth Barrett Browning [0.998922776695346]
2) William Shakespeare [0.998928467671442]
3) John Webster [0.999151810753424]
4) Richard Brinsley Sheridan [0.999209862746223]
5) Christopher Marlowe [0.999217109132166]
6) John Dryden [0.999282258234486]
7) Johann Wolfgang von Goethe [0.999321061927905]
8) Mary Rowlandson [0.999353291314523]
9) Howard Pyle [0.999357128425179]
10) Algernon Charles Swinburne [0.999358562358709]
njiang21@janny:~/cse13s/asgn7$
```

```
njiang21@janny:~/cse13s/resources/asgn7$ ./identify -c -k 10 <texts/will
iam-shakespeare.txt
Top 10, metric: Cosine distance, noise limit: 100
1) Elizabeth Barrett Browning [0.998922776695346]
2) William Shakespeare [0.998928467671442]
3) John Webster [0.999151810753424]
4) Richard Brinsley Sheridan [0.999209862746223]
5) Christopher Marlowe [0.999217109132166]
6) John Dryden [0.999282258234486]
7) Johann Wolfgang von Goethe [0.999321061927905]
8) Mary Rowlandson [0.999353291314523]
9) Howard Pyle [0.999357128425179]
10) Algernon Charles Swinburne [0.999358562358709]
```

You observe about your program’s behavior as you tune the number of noise words that are filtered out and the amount of text that you feed your program.

1）

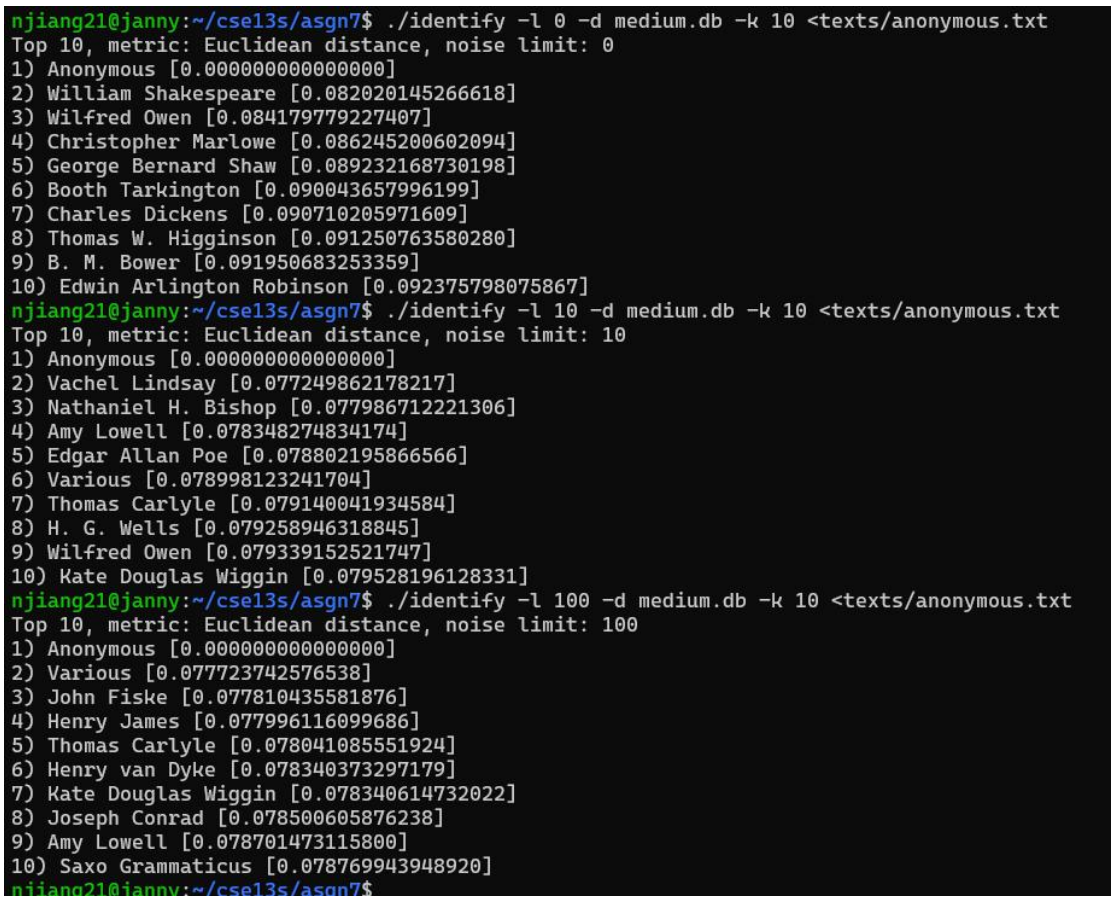
the distance different when the noise word different. We can see that when noiselimit change grom 0->10->100, the distance for each text becomes higher.

```
njiang21@janny:~/cse13s/asgn7$ ./identify -l 0 -d small.db -k 10 <texts/thomas-carlyle.txt
Top 10, metric: Euclidean distance, noise limit: 0
1) Thomas Carlyle [0.000000000000000]
2) H. G. Wells [0.023094613434560]
3) Daniel Defoe [0.023275524479483]
4) Henry Fielding [0.030697071020574]
5) Henry van Dyke [0.032031843261263]
6) Joseph Conrad [0.032804199322987]
7) Unknown [0.036322058016885]
8) Saxo Grammaticus [0.036450321857481]
9) Henry James [0.037190106360369]
10) Anonymous [0.093145977339713]
njiang21@janny:~/cse13s/asgn7$ ./identify -l 10 -d small.db -k 10 <texts/thomas-carlyle.txt
Top 10, metric: Euclidean distance, noise limit: 10
1) Thomas Carlyle [0.000000000000000]
2) Henry van Dyke [0.020454616885638]
3) Daniel Defoe [0.021408806705770]
4) H. G. Wells [0.022556521329783]
5) Henry James [0.023860150368918]
6) Joseph Conrad [0.026156221153175]
7) Henry Fielding [0.028284245796482]
8) Unknown [0.030373424639263]
9) Saxo Grammaticus [0.030961411707830]
10) Anonymous [0.079140041934604]
njiang21@janny:~/cse13s/asgn7$ ./identify -l 100 -d small.db -k 10 <texts/thomas-carlyle.txt
Top 10, metric: Euclidean distance, noise limit: 100
1) Thomas Carlyle [0.000000000000000]
2) Joseph Conrad [0.017166390487124]
3) Henry van Dyke [0.017738823972039]
4) H. G. Wells [0.018159389259155]
5) Henry James [0.018169885641644]
6) Henry Fielding [0.018863008966098]
7) Daniel Defoe [0.020333254684404]
8) Saxo Grammaticus [0.020424119118040]
9) Unknown [0.033337552670497]
10) Anonymous [0.078041085551897]
```

when noiselimit change grom 100->1000, the distance for each text becomes lower.

```
njiang21@janny:~/cse13s/asgn7$ ./identify -l 100 -d small.db -k 10 <texts/thomas-carlyle.txt
Top 10, metric: Euclidean distance, noise limit: 100
1) Thomas Carlyle [0.000000000000000]
2) Joseph Conrad [0.017166390487124]
3) Henry van Dyke [0.017738823972039]
4) H. G. Wells [0.018159389259155]
5) Henry James [0.018169885641644]
6) Henry Fielding [0.018863008966098]
7) Daniel Defoe [0.020333254684404]
8) Saxo Grammaticus [0.020424119118040]
9) Unknown [0.033337552670497]
10) Anonymous [0.078041085551897]
njiang21@janny:~/cse13s/asgn7$ ./identify -l 1000 -d small.db -k 10 <texts/thomas-carlyle.tx
t
Top 10, metric: Euclidean distance, noise limit: 1000
1) Thomas Carlyle [0.000000000000000]
2) Joseph Conrad [0.019030156655914]
3) H. G. Wells [0.019190497802083]
4) Henry Fielding [0.019510508041132]
5) Saxo Grammaticus [0.019771874156100]
6) Henry van Dyke [0.020065495366754]
7) Henry James [0.020287774481508]
8) Daniel Defoe [0.022072913542191]
9) Unknown [0.036337391077500]
10) Anonymous [0.084095984905416]
```

2) When we use larger amount of text that you feed the program, the order of the order can be changed.



But there is always a distance which is the author of this passage, and it’s order is 1 and it’s distance is always 0.0000000000.

3) I write two files by myself: a big one and a small one and large one. If the file size is really small, this program out put are too small and the difference between each file are really small.