Quiz #2

- Out of 30 points
- High: 28.75
- Ave: 23
- Will drop lowest quiz
- I do not grade based on absolutes

Class feedback

- Thanks!
- Specific comments:
  - “Less/no Java)”
  - “tell us to get up more often and stretch and high-five”
  - “Drop lowest quiz grade”
  - “more labs”

Class presentations
Class presentations

- Presentations done in pairs (and one triplet)
- 25 minutes for presentation 10 min. for Q+A
- In the week following your presentation, come by and see me for 5-10 min. for feedback
- 5% of your grade is based on your presentation
  - I will also be looking for improvement from this presentation to your final project presentation
- If you are not presenting, you should spend at least 30 min. on each paper reading it before class

Class presentations

- 7 of you still haven’t e-mailed me preferences!
- If you e-mail me by 5pm today, I’ll take those into account
- I will post the assignments later today
  - I’ll try and give everyone their first choice

Other Admin

- Assignment 5 (last assignment!) will be posted soon and due next Friday (4/1)
- I will post final project deadlines, specifications, etc. soon
  - Groups 2-3 (possibly 4)
  - ~4 weeks of actual coding/writing
  - Start thinking about final projects
  - Project proposals will be due ~ April 4
- How many of you are seniors?
  - I will have to shift some things in the schedule since you’re grades are due early 😊

Text Similarity

- A common question in NLP is how similar are texts
  - \( \text{score: } \text{sim}(\text{, } \text{) } = ? \)
  - \( \text{rank: } \text{? } \text{How could these be useful? Applications?} \)
Text similarity: applications

- Information retrieval (search)
  query
  Data set (e.g. web)

Text similarity: applications

- Text classification

  These "documents" could be actual documents, for example using k-means or pseudo-documents, like a class centroid/average

Text similarity: applications

- Text clustering

Text similarity: applications

- Automatic evaluation

  human answer

  text to text
  (machine translation, summarization, simplification)

  sim

  output
Text similarity: application

- Word similarity
  \[ \text{sim}(\text{banana}, \text{apple}) = ? \]

- Word-sense disambiguation
  I went to the bank to get some money.

\[ \text{financial bank} \rightarrow \text{river bank} \]

Text similarity: applications

- Text similarity approaches
  \[ \text{sim}( \ , \ ) = ? \]

A: When the defendant and his lawyer walked into the court, some of the victim supporters turned their backs to him.

B: When the defendant walked into the courthouse with his attorney, the crowd turned their backs on him.

How can we do this?
The basics: text overlap

- Texts that have overlapping words are more similar.

A: When the defendant and his lawyer walked into the court, some of the victim supporters turned their backs to him.

B: When the defendant walked into the courthouse with his attorney, the crowd turned their backs on him.

Word overlap: a numerical score

- Idea 1: number of overlapping words

A: When the defendant and his lawyer walked into the court, some of the victim supporters turned their backs to him.

B: When the defendant walked into the courthouse with his attorney, the crowd turned their backs on him.

\[ \text{sim}(T_1, T_2) = 11 \]

Word overlap problems

- Doesn't take into word order
- Related: doesn't reward longer overlapping sequences

A: defendant his the When lawyer into walked backs him the court, of supporters and some the victim turned their backs to him.

B: When the defendant walked into the courthouse with his attorney, the crowd turned their backs on him.

\[ \text{sim}(T_1, T_2) = 11 \]

Word overlap problems

Doesn't take into account length

A: When the defendant and his lawyer walked into the court, some of the victim supporters turned their backs to him.

B: When the defendant walked into the courthouse with his attorney, the crowd turned their backs on him. I ate a large banana at work today and thought it was great!

\[ \text{sim}(T_1, T_2) = 11 \]
Word overlap problems

Doesn’t take into account synonyms

A: When the defendant and his lawyer walked into the court, some of the victim supporters turned their backs to him.

B: When the defendant walked into the courthouse with his attorney, the crowd turned their backs on him.

\[ \text{sim}(T_1, T_2) = 11 \]

Word overlap problems

Doesn’t take into account spelling mistakes

A: When the defendant and his lawyer walked into the court, some of the victim supporters turned their backs to him.

B: When the defendant walked into the courthouse with his attorney, the crowd turned their backs on him. I ate a large banana at work today and thought it was great!

\[ \text{sim}(T_1, T_2) = 11 \]

Word overlap problems

Treats all words the same

A: When the defendant and his lawyer walked into the court, some of the victim supporters turned their backs to him.

B: When the defendant walked into the courthouse with his attorney, the crowd turned their backs on him.

Word overlap problems

May not handle frequency properly

A: When the defendant and his lawyer walked into the court, some of the victim supporters turned their backs to him. I ate a banana and then another banana and it was good!

B: When the defendant walked into the courthouse with his attorney, the crowd turned their backs on him. I ate a large banana at work today and thought it was great!
Word overlap: sets

A: When the defendant and his lawyer walked into the court, some of the victim supporters turned their backs to him.

B: When the defendant walked into the courthouse with his attorney, the crowd turned their backs on him.

- What is the overlap, using sets?
  - $|A \cap B|$ the size of the intersection
- How can we incorporate length/size into this measure?

Jaccard index (Jaccard similarity coefficient)

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Dice’s coefficient

$$Dice(A,B) = \frac{2 |A \cap B|}{|A|+|B|}$$

How are these related?

Hint: break them down in terms of:

- $|A - B|$ words in $A$ but not $B$
- $|B - A|$ words in $B$ but not $A$
- $|A \cap B|$ words in both $A$ and $B$
### Word overlap: sets

\[
J(A, B) = \frac{|A \cap B|}{|A \cup B|}
\]

\[
= \frac{|A \cap B|}{|A - B| + |B - A| + |A \cap B|}
\]

**Dice's coefficient** gives twice the weight to overlapping words.

\[
Dice(A, B) = \frac{2|A \cap B|}{|A| + |B|}
\]

\[
= \frac{2|A \cap B|}{|A - B| + |B - A| + 2|A \cap B|}
\]

### Set overlap

- Our problems:
  - word order
  - length
  - synonym
  - spelling mistakes
  - word importance
  - word frequency

Set overlap measures can be good in some situations, but often we need more general tools.

### Bag of words representation

For now, let's ignore word order:

- Clinton said banana repeatedly last week on tv, "banana, banana, banana."

- (4, 1, 1, 0, 0, 0, 0, 0, ...)

Frequency of word occurrence

### Vector based word

A

- a₁ When 1
- a₂ the 2
- a₃ defendant 1
- a₄ and 1
- a₅ courthouse 0
- ...

Think of these as feature vectors.

B

- b₁ When 1
- b₂ the 2
- b₃ defendant 1
- b₄ and 0
- b₅ courthouse 1
- ...

How do we calculate the similarity based on these feature vectors?
**Vector based similarity**

- We have a $|V|$-dimensional vector space
- Terms are axes of the space
- Documents are points or vectors in this space
- Very high-dimensional
- This is a very sparse vector - most entries are zero

What question are we asking in this space for similarity?

**Distance measures**

- Euclidean (L2)
  
  \[ sim(A,B) = \sqrt{\sum (a_i - b_i)^2} \]

- Manhattan (L1)

  \[ sim(A,B) = \sum |a_i - b_i| \]

**Distance can be problematic**

Which \(d\) is closest to \(q\) using one of the previous distance measures?

Which do you think should be closer?
Distance can be problematic

The Euclidean (or L1) distance between \( q \) and \( d_2 \) is large even though the distribution of words is similar.

Use angle instead of distance

- Thought experiment:
  - take a document \( d \)
  - make a new document \( d' \) by concatenating two copies of \( d \)
  - “Semantically” \( d \) and \( d' \) have the same content
  - What is the Euclidean distance between \( d \) and \( d' \)?
  - What is the angle between them?
  - The Euclidean distance can be large
  - The angle between the two documents is 0

From angles to cosines

- Cosine is a monotonically decreasing function for the interval \([0^\circ, 180^\circ]\)
- decreasing angle is equivalent to increasing cosine

Cosine

How do we calculate the cosine between two vectors?
Cosine of two vectors

\[ A \cdot B = \|A\| \|B\| \cos \theta \]

\[ \cos \theta = \frac{A \cdot B}{\|A\| \|B\|} \]

Dot product

Just another distance measure, like the others:

\[ \text{sim}_{\cos}(A, B) = A \cdot B = \sum_{i=1}^{n} a_i b_i \]

Dealing with length

- Thought experiment, revisited:
  - take a document \( d \)
  - make a new document \( d' \) by concatenating two copies of \( d \)
- How does \( \text{sim}_{\cos}(d, d) \) relate to \( \text{sim}_{\cos}(d, d') \)?
- Does this make sense?

Length normalization

- A vector can be length-normalized by dividing each of its components by its length
- Often, we’ll use \( L_2 \) norm (could also normalize by other norms):
  \[ \|v\| = \sqrt{\sum v_i^2} \]
- Dividing a vector by its \( L_2 \) norm makes it a unit (length) vector
Unit length vectors

In many situations, normalization improves similarity, but not in all situations.

Normalized distance measures

- **Cosine**
  \[ \text{sim}_{\cos}(A,B) = \frac{A \cdot B}{\|A\| \cdot \|B\|} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2} \sqrt{\sum_{i=1}^{n} b_i^2}} \]

- **L2**
  \[ \text{sim}_{L2}(A,B) = \sqrt{\sum_{i=1}^{n} (a'_i - b'_i)^2} \]

- **L1**
  \[ \text{sim}_{L1}(A,B) = \sum_{i=1}^{n} |a'_i - b'_i| \]

\(a'\) and \(b'\) are length normalized versions of the vectors.

Cosine similarity with 3 documents

<table>
<thead>
<tr>
<th>Term</th>
<th>SaS</th>
<th>PoP</th>
<th>WH</th>
</tr>
</thead>
<tbody>
<tr>
<td>affection</td>
<td>115</td>
<td>58</td>
<td>20</td>
</tr>
<tr>
<td>jealous</td>
<td>10</td>
<td>7</td>
<td>11</td>
</tr>
<tr>
<td>gossip</td>
<td>2</td>
<td>0</td>
<td>6</td>
</tr>
</tbody>
</table>

Term frequencies (counts)

Length normalized

<table>
<thead>
<tr>
<th>Term</th>
<th>SaS</th>
<th>PoP</th>
<th>WH</th>
</tr>
</thead>
<tbody>
<tr>
<td>affection</td>
<td>0.99</td>
<td>0.99</td>
<td>0.84</td>
</tr>
<tr>
<td>jealous</td>
<td>0.08</td>
<td>0.1</td>
<td>0.46</td>
</tr>
<tr>
<td>gossip</td>
<td>0.02</td>
<td>0.0</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Often becomes much clearer after length normalization.
Our problems

- Which of these have we addressed?
  - word order
  - length
  - synonym
  - spelling mistakes
  - word importance
  - word frequency

Word overlap problems

Treats all words the same

A: When the defendant and his lawyer walked into the court, some of the victim supporters turned their backs to him.

B: When the defendant walked into the courthouse with his attorney, the crowd turned their backs on him.

Word importance

- Include a weight for each word/feature

A

<table>
<thead>
<tr>
<th></th>
<th>w_1</th>
<th>w_2</th>
<th>w_3</th>
<th>w_4</th>
<th>w_5</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a.</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a.</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a.</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

B

<table>
<thead>
<tr>
<th></th>
<th>w_1</th>
<th>w_2</th>
<th>w_3</th>
<th>w_4</th>
<th>w_5</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>b.</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b.</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b.</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b.</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

13
Distance + weights

- We can incorporate the weights into the distances
- Think of it as either (both work out the same):
  - preprocessing the vectors by multiplying each dimension by the weight
  - incorporating it directly into the similarity measure

\[
\text{sim}_{w}(A,B) = \frac{\sum_{i} w_a i w_b i}{\sqrt{\sum_{i} (w_a i)^2} \sqrt{\sum_{i} (w_b i)^2}}
\]

Idea: use corpus statistics

- document frequency (df) is one measure of word importance
- Terms that occur in many documents are weighted less, since overlapping with these terms is very likely
  - In the extreme case, take a word like the that occurs in EVERY document
- Terms that occur in only a few documents are weighted more

Document frequency

- The overall frequency of a word is the number of occurrences in a dataset, counting multiple occurrences
- Example:

<table>
<thead>
<tr>
<th>Word</th>
<th>Overall frequency</th>
<th>Document frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>insurance</td>
<td>10440</td>
<td>3997</td>
</tr>
<tr>
<td>try</td>
<td>10422</td>
<td>8760</td>
</tr>
</tbody>
</table>

- Which word is a better search term (and should get a higher weight)?

Document vs. overall frequency
Document frequency

<table>
<thead>
<tr>
<th>Word</th>
<th>Collection frequency</th>
<th>Document frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>insurance</td>
<td>10440</td>
<td>3997</td>
</tr>
<tr>
<td>try</td>
<td>10422</td>
<td>8760</td>
</tr>
</tbody>
</table>

Document frequency is often related to word importance, but we want an actual weight. Problems?

\[
sim_{\text{sim}}(A, B) = \frac{\sum_i (w_i^A w_i^B)}{\sqrt{\sum_i (w_i^A)^2} \sqrt{\sum_i (w_i^B)^2}}
\]

From document frequency to weight

<table>
<thead>
<tr>
<th>Word</th>
<th>Collection frequency</th>
<th>Document frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>insurance</td>
<td>10440</td>
<td>3997</td>
</tr>
<tr>
<td>try</td>
<td>10422</td>
<td>8760</td>
</tr>
</tbody>
</table>

- weight and document frequency are inversely related
- higher document frequency should have lower weight and vice versa
- document frequency is unbounded
- document frequency will change depending on the size of the data set (i.e. the number of documents)

Inverse document frequency

\[
idf_w = \log \frac{N}{\text{df}_w}
\]

- \(\text{idf}_w\) is inversely correlated with \(\text{df}\)
- higher \(\text{df}\) results in lower idf
- \(N\) incorporates a dataset dependent normalizer
- \(\log\) dampens the overall weight

idf example, suppose \(N = 1\) million

<table>
<thead>
<tr>
<th>term</th>
<th>(\text{df}_t)</th>
<th>(\text{idf}_t)</th>
</tr>
</thead>
<tbody>
<tr>
<td>calpurnia</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>animal</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>sunday</td>
<td>1,000</td>
<td></td>
</tr>
<tr>
<td>fly</td>
<td>10,000</td>
<td></td>
</tr>
<tr>
<td>under</td>
<td>100,000</td>
<td></td>
</tr>
<tr>
<td>the</td>
<td>1,000,000</td>
<td></td>
</tr>
</tbody>
</table>

What are the idfs assuming log base 10?
There is one idf value/weight for each word.

What if we didn't use the log to dampen the weighting?

TF-IDF

- One of the most common weighting schemes
- TF = term frequency
- IDF = inverse document frequency
  \[ a'_i = a_i \times \log \frac{N}{df_i} \]

We can then use this with any of our similarity measures!