Assignments 4

Quiz #2 Thursday
- Same rules as quiz #1
- First 30 minutes of class
- Open book and notes

Assignment 5 out on Thursday

Quiz #2 Topics
- Linguistics 101
- Parsing
  - Grammars, CFGs, PCFGs
  - Top-down vs. bottom-up
  - CKY algorithm
  - Grammar learning
  - Evaluation
  - Improved models
- Text similarity
  - Will also be covered on Quiz #3, though

Text Similarity
A common question in NLP is how similar are texts

score: \( \text{sim}( \quad , \quad ) = ? \)

rank: ?
Bag of words representation

For now, let’s ignore word order:

Obama said bananas repeatedly last week on TV, “banana, banana, banana.”

$\langle 4, 1, 1, 0, 1, 0, 0, \ldots \rangle$

“Bag of words representation”: multi-dimensional vector, one dimension per word in our vocabulary

Frequency of word occurrence

Vector based word

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>When</td>
<td>b1</td>
</tr>
<tr>
<td>a2</td>
<td>the</td>
<td>b2</td>
</tr>
<tr>
<td>a3</td>
<td>defendant</td>
<td>b3</td>
</tr>
<tr>
<td>a4</td>
<td>and</td>
<td>b4</td>
</tr>
<tr>
<td>a5</td>
<td>courthouse</td>
<td>b5</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Multi-dimensional vectors, one dimension per word in our vocabulary

How do we calculate the similarity based on these vectors?

Normalized distance measures

Cosine

\[
sim_{cos}(A, B) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum a_i b_i}{\sqrt{\sum a_i^2} \sqrt{\sum b_i^2}}
\]

L2

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

L1

\[
dist_{L1}(A, B) = \sum |a_i - b_i|
\]

$a'$ and $b'$ are length normalized versions of the vectors

Our problems

So far...

- word order
- length
- synonym
- spelling mistakes
- word importance
- word frequency

For now, let’s ignore word order:

Obama said bananas repeatedly last week on TV, “banana, banana, banana.”

$\langle 4, 1, 1, 0, 1, 0, 0, \ldots \rangle$
Word importance

Include a weight for each word/feature

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1 (w_1)</td>
<td>1 (w_1)</td>
</tr>
<tr>
<td>b</td>
<td>2 (w_2)</td>
<td>2 (w_2)</td>
</tr>
<tr>
<td>c</td>
<td>1 (w_3)</td>
<td>1 (w_3)</td>
</tr>
<tr>
<td>d</td>
<td>1 (w_4)</td>
<td>0 (w_4)</td>
</tr>
<tr>
<td>e</td>
<td>0 (w_5)</td>
<td>1 (w_5)</td>
</tr>
</tbody>
</table>

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

\[
sim_{cos}(A, B) = \frac{A \cdot B}{\sqrt{\sum_i (w_i a_i)^2} \sqrt{\sum_i (w_i b_i)^2}}
\]

Document vs. overall 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 more informative (and should get a higher weight)?

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?

\[
sim_{cos}(A, B) = \frac{A \cdot B}{\sqrt{\sum_i (w_i a_i)^2} \sqrt{\sum_i (w_i b_i)^2}}
\]
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 dataset (i.e., the number of documents).

Inverse document frequency

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

- IDF is inversely correlated with DF.
- Higher 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>word</th>
<th>( df_w )</th>
<th>( \text{idf}_w )</th>
</tr>
</thead>
<tbody>
<tr>
<td>calpurnia</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>animal</td>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>sunday</td>
<td>1,000</td>
<td>3</td>
</tr>
<tr>
<td>fly</td>
<td>10,000</td>
<td>2</td>
</tr>
<tr>
<td>under</td>
<td>100,000</td>
<td>1</td>
</tr>
<tr>
<td>the</td>
<td>1,000,000</td>
<td>0</td>
</tr>
</tbody>
</table>

What are the IDF values assuming log base 10?

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

IDF example, suppose N=1 million

<table>
<thead>
<tr>
<th>word</th>
<th>( df_w )</th>
<th>( \text{idf}_w )</th>
</tr>
</thead>
<tbody>
<tr>
<td>calpurnia</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>animal</td>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>sunday</td>
<td>1,000</td>
<td>3</td>
</tr>
<tr>
<td>fly</td>
<td>10,000</td>
<td>2</td>
</tr>
<tr>
<td>under</td>
<td>100,000</td>
<td>1</td>
</tr>
<tr>
<td>the</td>
<td>1,000,000</td>
<td>0</td>
</tr>
</tbody>
</table>

There is one idf value/weight for each word.

\[ \text{idf}_w = \log \frac{N}{\text{df}_w} \]
TF-IDF

One of the most common weighting schemes

- **TF** = term frequency
- **IDF** = inverse document frequency

\[
a'_i = a_i \times \log \frac{N}{df_i}
\]

**Stoplists:** extreme weighting

Some words like ‘a’ and ‘the’ will occur in almost every document
- IDF will be 0 for any word that occurs in all documents
- For words that occur in almost all of the documents, they will be nearly 0

A *stoplist* is a list of words that should not be considered (in this case, similarity calculations)
- Sometimes this is the n most frequent words
- Often, it's a list of a few hundred words manually created
If most of these end up with low weights anyway, why use a stoplist?

Two main benefits
- More fine grained control: some words may not be frequent, but may not have any content value (alas, teh, gosh)
- Often does contain many frequent words, which can drastically reduce our storage and computation

Any downsides to using a stoplist?
- For some applications, some stop words may be important

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

A model of word similarity!

Word overlap problems

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 similarity

How similar are two words?

Score: \( \text{sim}(w_1, w_2) = ? \)  rank: \( w_1 \)  \( w_2 \)  \( w_3 \)

List: \( w_1 \) and \( w_2 \) are synonyms

Word similarity applications

- General text similarity
- Thesaurus generation
- Automatic evaluation
- Text-to-text
  - Paraphrasing
  - Summarization
  - Machine translation
- Information retrieval (search)

Four categories of approaches (maybe more)

- Character-based
- Turned vs. tuned
- Cognates (night, nacht, nicht, natt, nat, noch)
- Semantic web-based (e.g. WordNet)
- Dictionary-based
- Distributional similarity-based
  - Similar words occur in similar contexts
**Character-based similarity**

\[ \text{sim}(\text{turned}, \text{truned}) = ? \]

How might we do this using only the words (i.e. no outside resources?)

---

**Edit distance (Levenshtein distance)**

The edit distance between \( w_1 \) and \( w_2 \) is the minimum number of operations to transform \( w_1 \) into \( w_2 \)

Operations:
- insertion
- deletion
- substitution

\[ \text{EDIT}(\text{turned}, \text{truned}) = ? \]
\[ \text{EDIT}(\text{computer}, \text{commuter}) = ? \]
\[ \text{EDIT}(\text{banana}, \text{apple}) = ? \]
\[ \text{EDIT}(\text{wombat}, \text{worcester}) = ? \]

---

**Edit distance**

\[ \text{EDIT}(\text{turned}, \text{truned}) = 2 \]
- delete u
- insert u

\[ \text{EDIT}(\text{computer}, \text{commuter}) = 1 \]
- replace p with m

\[ \text{EDIT}(\text{banana}, \text{apple}) = 5 \]
- delete b
- replace n with p
- replace a with p
- replace a with t
- replace a with e

\[ \text{EDIT}(\text{wombat}, \text{worcester}) = 6 \]

---

**Better edit distance**

Are all operations equally likely?
- No

Improvement, give different weights to different operations
- replacing a for e is more likely than z for y

Ideas for weightings?
- Learn from actual data (known typos, known similar words)
- Intuitions: phonetics
- Intuitions: keyboard configuration
Vector character-based word similarity

\[ \text{sim}(\text{turned, truned}) = ? \]

Any way to leverage our vector-based similarity approaches from last time?

Vector character-based word similarity

\[ \text{sim}(\text{restful, fluster}) = ? \]

Character level loses a lot of information

Character bigrams or even trigrams

Use character bigrams or even trigrams
Word similarity

Four general categories
- Character-based
  - turned vs. tuned
  - cognates (night, nacht, nicht, natt, nat, noc, noch)
- Semantic web-based (e.g. WordNet)
- Dictionary-based
- Distributional similarity-based
  - similar words occur in similar contexts

WordNet

Lexical database for English
- 155,867 words
- 206,941 word senses
- 117,689 synsets (synonym sets)
- ~400K relations between senses
- Parts of speech: nouns, verbs, adjectives, adverbs

Psycholinguistics
- WN attempts to model human lexical memory
- Design based on psychological testing

Created by researchers at Princeton
- http://wordnet.princeton.edu/
- Lots of programmatic interfaces

WordNet relations
- synonym
- antonym
- hypernyms
- hyponyms
- holonym
- meronym
- troponym
- entailment
- (and a few others)
WordNet relations

- **troponym** – for verbs, a more specific way of doing an action
  - run is a troponym of move
  - dice is a troponym of cut

- **entailment** – for verbs, one activity leads to the next
  - sleep is entailed by snore

(and a few others)

**WordNet:**

**dog**

**Noun**

- [6] dog, domestic dog, Canis familiaris: a member of the genus Canis (probably descended from the common wolf) that has been domesticated by man since prehistoric times; occurs in many breeds: "the dog licked all night";
- [6] dog (a full-uncovered unambitious girl or woman) "she got a reputation as a flapper"; "she's a real dog";
- [5] domestic dog (informal term for a man) "you lucky dog";
- [5] in dog country (anywhere), dog, domestic dog (someone who is morally reprehensible) "you sleepy dog";
- [5] race, rambler, hobbyist, hunting dog, cattledog, terrier: fancied as having qualities suitable to some particular purpose (can be trained to perform a duty)
- [5] pure breed, pure dog, pure dog (a breed that fits into a branch of a breed to move a whole forward or prevent it from becoming backward);
- [5] working dog, hunting dog, dog-kennel (dwelling for dogs in a place) "the audience were too far to smell";

**Verb**

- [5] chase, chase after, trail, raid, pounce, chase, dog, go after, hunt (go after with the intent to catch): "The policeman chased the mugger down the alley"; "the dog chased the rabbit";
To utilize WordNet, we often want to think about some graph-based measure.

Rank the following based on similarity:

\[
\begin{align*}
\text{SIM}(\text{wolf}, \text{dog}) \\
\text{SIM}(\text{wolf}, \text{amphibian}) \\
\text{SIM}(\text{terrier}, \text{wolf}) \\
\text{SIM}(\text{dachshund}, \text{terrier})
\end{align*}
\]

What information/heuristics did you use to rank these?

- path length is important (but not the only thing)
- words that share the same ancestor are related
- words lower down in the hierarchy are finer grained and therefore closer
WordNet similarity measures

Path length doesn't work very well

Some ideas:
- Path length scaled by the depth (Leacock and Chodorow, 1998)

With a little cheating:
- Measure the “information content” of a word using a corpus: how specific is a word?
  - Words higher up tend to have less information content
  - More frequent words (and ancestors of more frequent words) tend to have less information content

Utilizing information content:
- Information content of the lowest common parent (Resnik, 1995)
- Information content of the words minus information content of the lowest common parent (Jiang and Conrath, 1997)
- Information content of the lowest common parent divided by the information content of the words (Lin, 1998)