EDIT DISTANCE

CS 445 - NATURAL LANGUAGE PROCESSING

Instructor: Dilara Kekülüoğlu
Fall 2024 – Week 3

Recap

1. Regular expressions – match patterns to text
2. Precision/Recall
3. Text Normalization
    1. Tokenization
    2. Lemmatization
    3. Stemming
    4. Text Cleaning
    5. ….

Pattern	Expansion	Example
\w	[0-9a-zA-Z_]	**A**dverb
\W	[^\w]	Hi**!**
\d	[0-9]	**0**07 Bond
\D	[^0-9]	**C**risp
\s	[\r\t\n\f]	Good_news
\S	[^\s]	**G**ood news

Tophat Exercise

- Write your favorite regex =)
- Attendance

Learning Goals (Week 3)

1. **Describe** edit distance and its uses
2. **Apply** edit distance algorithm
3. **Describe** text analysis process
4. **Develop** a simple sentiment analyzer
5. **Describe** data annotation

Edit Distance

- Given two strings a and b, how **similar** are they?
  - How many edit steps between a and b?
- Why do we need this?
  - **Spellchecking** – one of the steps of text normalization
  - Bioinformatics - **DNA similarity** with A, C, G and T letters used for representation
  - **Speech Recognition**

Most of the edit distance slides are taken from Jurafsky&Martin

Edit Distance

- Various algorithms but in this lecture we will talk about **Levenshtein distance**.
- Three operations:
  - Insertion
  - Deletion
  - Substitution
- Minimum count of operations between two strings

Edit Distance

- Two strings and their alignment

```
I N T E * N T I O N
| | | | | | | | | |
* E X E C U T I O N
```

Edit Distance

```
I N T E * N T I O N
| | | | | | | | | |
* E X E C U T I O N
d s s   i s
```

d: deletion
s: substitution
i: insertion

- If each operation has a cost of 1, distance is 5.
- If substitution is 2 units of cost, distance is 8.

Alignment in Computational Biology

- Given a sequence of bases

AGGCTATCACCTGACCTCCAGGCCGATGCCC
TAGCTATCACGACCGCGGTCGATTTGCCCGAC

- An alignment:

```
-AGGCTATCACCTGACCTCCAGGCCGA--TGCCC---
TAG-CTATCAC--GACCGC--GGTCGATTTGCCCGAC
```

- Given two sequences, align each letter to a letter or gap

Speech Recognition

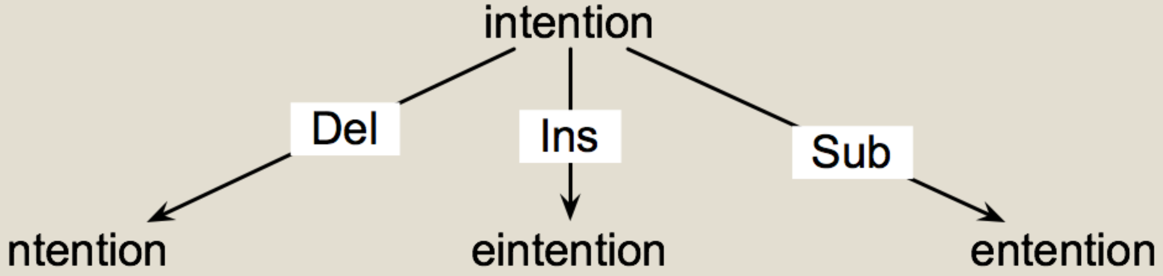
- Evaluating Machine Translation and speech recognition

```
R Spokesman confirms     senior government adviser was shot
H Spokesman said     the senior            adviser was shot dead
                S      I              D                        I
```

- Distance between the real speech and the detected speech shows the performance of the speech recognition system.

How to calculate min edit distance?

- We need to search for a path between the two strings using the operators.
- **Initial state**: the word we are transforming
- **Operators**: insert, delete, substitute
- **Goal state**: the word we are trying to reach
- **Path cost**: the number of edits (goal is minimizing this)



Defining the edit distance problem

- For two strings X→Y
  - X of length *n*
  - Y of length *m*
- We define $D(i,j)$
  - the edit distance between X[1..*i*] and Y[1..*j*]
    - i.e., the first *i* characters of X and the first *j* characters of Y
  - The edit distance between X and Y is thus $D(n,m)$

Brute Force Approach

- Solving from the i^{th} character of the X and j^{th} character of Y – assume we know `D(i,j)`
- If they are the same - solve the problem for `D(i+1,j+1)`
- If not,
  - Apply deletion – solve the problem for `D(i+1,j)`
  - Apply insertion – solve the problem for `D(i,j+1)`
  - Apply substitution – solve the problem for `D(i+1,j+1)`
- Recursively solve the problem for every step

Brute Force Approach

- Algorithmic Complexity?
  - O(3^n)
- At worst 3 options at every step.
- Lots of sub-problems solved multiple times.
- Let’s say,
  - we make s → `solve D(i+1,j+1)`
  - we make i-d → `solve D(i+1,j+1)`
- We do not need to compute it again.

Memoization Method

- Keep a cache for already computed solutions
- If we make s →`solve D(i+1,j+1)`
- Keep the solution in a cache
- When we come across `D(i+1,j+1)` again, we know the solution.
- More space needed – O(n^2)
- Compute all paths and take the minimum

Tabulation Method

- Most optimized method
- Uses dynamic programming
- Bottom-up
    - We compute `D(i,j)` for small *i,j*
    - And compute larger `D(i,j)` based on previously computed smaller values
    - i.e., compute `D(i,j)` for all *i* (0 < *i* < n) and *j* (0 < j < m)

Dynamic Programming

- Solving problems by combining solutions to **subproblems**
  - Break the main problem into sub-problems
  - Solve the sub-problems optimally
  - Use these solutions to solve the main problem
- Will come up in interview questions
- Not really easy to design – once designed well, easy to solve

Tabulation Method

- Initialization

  `D(i,0) = i`

  `D(0,j) = j`

- Recurrence Relation:

```
For each  i = 1…M
      For each  j = 1…N
```

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 1; \text{ if } X(i) \neq Y(j) \\ 0; \text{ if } X(i) = Y(j) \end{cases} \end{cases}$$

- Termination:

  `D(N,M) is distance`

The Edit Distance Table

N	9									
O	8									
I	7									
T	6									
N	5									
E	4									
T	3									
N	2									
I	1									
#	**0**	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

The Edit Distance Table

N	9									
O	8									
I	7									
T	6									
N	5									
E	4									
T	3									
N	2									
I	1									
#	0	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 1; \text{ if } X(i) \neq Y(j) \\ 0; \text{ if } X(i) = Y(j) \end{cases} \end{cases}$$

The Edit Distance Table

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 1; \text{ if } X(i) \neq Y(j) \\ 0; \text{ if } X(i) = Y(j) \end{cases} \end{cases}$$

N	9	8								
O	8	7								
I	7	6								
T	6	5								
N	5	4								
E	4	3								
T	3	3								
N	2	2								
I	1	1								
#	0	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

The Edit Distance Table

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 1; \text{ if } X(i) \neq Y(j) \\ 0; \text{ if } X(i) = Y(j) \end{cases} \end{cases}$$

N	9	8	8	8	8	8	8	7	6	**5**
O	8	7	7	7	7	7	7	6	5	6
I	7	6	6	6	6	6	6	5	6	7
T	6	5	5	5	5	5	5	6	7	8
N	5	4	4	4	4	5	6	7	7	7
E	4	3	4	3	4	5	6	6	7	8
T	3	3	3	3	4	5	5	6	7	8
N	2	2	2	3	4	5	6	7	7	7
I	1	1	2	3	4	5	6	6	7	8
#	**0**	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

The Edit Distance Table

- We know the minimum edit distance
- How about the path?
- How do we know where to use i/d/s?
- Keep four information in each cell – backtrack from the minimum value

Cost of getting here from left neighbor (insert)	The minimum of three possible movements
Cost of getting here from lower left neighbor (copy or substitute)	Cost of getting here from lower neighbor (delete)

The Edit Distance Table (smaller example)

Distance from
Crab → Ruby

B		**4**								
		4								
A		**3**								
		3								
R		**2**								
		2								
C		**1**	2	**1**						
		1	**1**	2						
#		**0**	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
	#		R		U		B		Y	

The Edit Distance Table (smaller example)

B		4								
		4								
A		3								
		3								
R		2								
		2								
C		1	2	1						
		1	1	2						
#		0	1	1	2	2	3	3	4	4
	#		R		U		B		Y	

The Edit Distance Table (smaller example)

B		4								
		4								
A		3								
		3								
R		2								
		2								
C		1	2	1	2	2				
		1	1	2	2	3				
#		0	1	1	2	2	3	3	4	4
	#		R		U		B		Y	

The Edit Distance Table (smaller example)

B		4								
		4								
A		3								
		3								
R		2								
		2								
C		1	2	1	2	2	3	3		
		1	1	2	2	3	3	4		
#		0	1	1	2	2	3	3	4	4
	#		R		U		B		Y	

The Edit Distance Table (smaller example)

B		4								
		4								
A		3								
		3								
R		2								
		2								
C		1	2	1	2	2	3	3	4	4
		1	1	2	2	3	3	4	4	5
#		0	1	1	2	2	3	3	4	4
	#		R		U		B		Y	

The Edit Distance Table (smaller example)

Fill up the rest of the table

B		4								
		4								
A		3								
		3								
R		2								
		2								
C		1	2	1	2	2	3	3	4	4
		1	1	2	2	3	3	4	4	5
#		0	1	1	2	2	3	3	4	4
	#		R		U		B		Y	

The Edit Distance Table (smaller example)

B		4	5	3	4	3	4	2	3	3
		4	4	3	3	3	2	4	4	5
A		3	4	2	3	2	3	3	4	4
		3	3	2	2	3	3	4	4	5
R		2	3	1	2	2	3	3	4	4
		2	1	2	2	3	3	4	4	5
C		1	2	1	2	2	3	3	4	4
		1	1	2	2	3	3	4	4	5
#		0	1	1	2	2	3	3	4	4
	#		R		U		B		Y	

The Edit Distance Table

B		4	5	3	4	3	4	2	3	3
		4	4	3	3	3	2	4	4	5
A		3	4	2	3	2	3	3	4	4
		3	3	2	2	3	3	4	4	5
R		2	3	1	2	2	3	3	4	4
		2	1	2	2	3	3	4	4	5
C		1	2	1	2	2	3	3	4	4
		1	1	2	2	3	3	4	4	5
#		0	1	1	2	2	3	3	4	4
	#		R		U		B		Y	

Now we know how we arrived at 3 distance

Insert

Delete

Copy – substitute

The Edit Distance Table

B		4	5	3	4	3	4	2	3	3
		4	4	3	3	3	2	4	4	5
A		3	4	2	3	2	3	3	4	4
		3	3	2	2	3	3	4	4	5
R		2	3	1	2	2	3	3	4	4
		2	1	2	2	3	3	4	4	5
C		1	2	1	2	2	3	3	4	4
		1	1	2	2	3	3	4	4	5
#		0	1	1	2	2	3	3	4	4
	#		R		U		B		Y	

Now we know how we arrived at 3 distance

Insert
Delete
Copy – substitute

cost	operation	input	output
1	insert	*	Y
0	copy	B	B
1	substitute	A	U
0	copy	R	R
1	delete	C	*

Edit Distance

- The costs were unit – substituting s → o is same with s→d
- However, using qwerty keyboards the latter is more probable
- Using Nokia 3310, x→y is more probable than a keyboard

Photos: Pixabay, Masood Aslami

Edit Distance

- Distance between two strings in terms of the number of edits needed to reach from one to another

- Spellchecking
- Bioinformatics
- Speech analysis
- Machine translation
- ...

TEXT ANALYSIS STEPS

CS 445 - NATURAL LANGUAGE PROCESSING

Instructor: Dilara Keküllüoğlu
Fall 2024 – Week 3

Text Analysis Pipeline

- First week, we showed some examples analyzing corpora statistically
- We use NLP to analyze and classify texts for given tasks
- Steps of analysis:
  - Dataset selection and exploration (Week 1)
  - Text Pre-processing (Week 1-2)
  - Feature Extraction (Week 4-5)
  - Classification (Week 5)
  - Evaluation (Week 5)
- How to train and evaluate – using annotations and labels (Week 3)

Sentiment Analysis Example

- Predict whether an opinion expressed in a text is positive or negative
  - Movie Reviews
  - Product Reviews
  - Opinions about political figures

A lot of decisions for the development

- What is the **input data**? – sentence-level, full review, meta-data (star ratings)
- Possible **outputs** – positive, negative, or number of stars?
- **Decision** algorithm – summing sentiment scores, supervised learning with labels,
- **Evaluation** method

Decisions

- Full-text only data
- + or – output
- **Dictionary-based sentiment score calculation**
- Accuracy - # correct / # total

Sentiment Lexicons

- Lexicon: (a list of) all the words used in a particular language or subject, or a dictionary (Cambridge Dictionary)
- Sentiment Lexicons – list of positive and negative words

Positive:

absolutely	beaming	calm
adorable	beautiful	celebrated
accepted	believe	certain
acclaimed	beneficial	champ
accomplish	bliss	champion
achieve	bountiful	charming
action	bounty	cheery
active	brave	choice
admire	bravo	classic
adventure	brilliant	classical
affirm	bubbly	clean
. . .		. . .

Negative:

abysmal	bad	callous
adverse	banal	can't
alarming	barbed	clumsy
angry	belligerent	coarse
annoy	bemoan	cold
anxious	beneath	collapse
apathy	boring	confused
appalling	broken	contradictory
atrocious		contrary
awful		corrosive
		corrupt
		. . .

From `http://www.enchantedlearning.com/wordlist/`

Sentiment Prediction

- Simple: # positive - # negative
- Some words might be stronger in sentiment than others
  - Extremely – Fairly
  - Great – Good
  - Disgusting – Unpleasant
- Weighted scores might be needed

Simple counting disadvantages

- **Sarcasm**! – not easy to determine whether positive words used positively
- **Negation** – not great?
- The text might not reflect the opinion of the author but others
- Movie reviews – explaining the character but **not the opinion**
- Instead of counting tokens, **labels** could be used for training the data

Annotation

- Gold standard – labels given to data
  - To train and evaluate the performance of classifiers, ML systems
- Movie_reviews – positive, negative
- Sarcasm detection – sarcastic, not sarcastic
- News labels – health, sports, entertainment, …

Play Annotator - Sarcasm

sarcasm: the use of remarks that clearly mean the opposite of what they say, made in order to hurt someone's feelings or to criticize something in a humorous way. – Cambridge Dictionary

Given following sentences, annotate each either sarcastic or not-sarcastic.

- Really, Sherlock? No! You are clever.
- I'm glad we're having a rehearsal dinner. I rarely practice my meals before I eat.
- You look really nice today!
- I am glad I had my coffee for today.
- I like chocolates.

Annotators

- Gold $\neq$ Perfect
- People can label incorrectly
- Some labels might be subjective or ambiguous – sarcasm, hate speech, etc.
- People bring their biases

Annotation Guidelines

- Follow a guideline to have **reliable annotations**
- Guideline created **iteratively** documenting steps
- Common understanding by the annotators and people who use these annotated dataset
- Penn Treebank POS Guidelines – 3 pages on adjectives vs verbs

It is human to err

- Even with perfect guidelines – there could be incorrect labels
- Hitting wrong button
- Not reading fully
- Getting distracted by other things
- Cases that were not covered in guidelines

How can we measure the quality of annotations?

Inter-Annotator Agreement

- Have multiple people label the same data
- Compare the labels – percentage of agreement
- Ideally, 100% of the labels would be agreed by all
- Not commonly the case
- Different tasks have different thresholds for acceptable agreement rates

Inter-Annotator Agreement

- If the agreement is low, reinspection on the data and iteration on the guideline is required.

- The human agreement rate is used as a benchmark for the algorithm performance on the task

Where to find annotators?

- Small datasets – annotated by a few people
- What about large datasets?
- Not feasible to annotate by the same people
- Crowdsourcing!

Crowdsourcing

- Using the wisdom of the crowd
- Amazon Mechanical Turk, Prolific Academic, etc. to recruit people
- Easier to find people to annotate but also might need more training to ensure quality

Crowdsourcing

- 5+ annotators for each data
- Test data with known labels – reject people who consistently fail them
- Might be really expensive to achieve quality labels

Next Steps on Analysis

- Once we have our data and labels
  - Extract **features** from the data
  - **Train** systems with given labels
  - **Test** systems on the unseen data
- When we do not have labels, **unsupervised learning** could be used – more on this later
- Language models and feature extraction – next weeks
- Classification – week 5

Learning Goals (Week 3)- revisited

1. **Describe** edit distance and its uses
2. **Apply** edit distance algorithm
3. **Describe** text analysis process
4. **Develop** a simple sentiment analyzer
5. **Describe** data annotation

Following lectures

- Language Models

Further resources

- Jurasky & Martin, Chapter 2
- Peter Norvig's Spellchecker - https://norvig.com/spell-correct.html
- https://www.yourdictionary.com/articles/examples-sarcasm-meaning-types