

NATURAL LANGUAGE PROCESSING

BHAVIK GANDHI

INTRODUCTION

OHH YEAH LET'S SAY HI!

What's up in NLP?

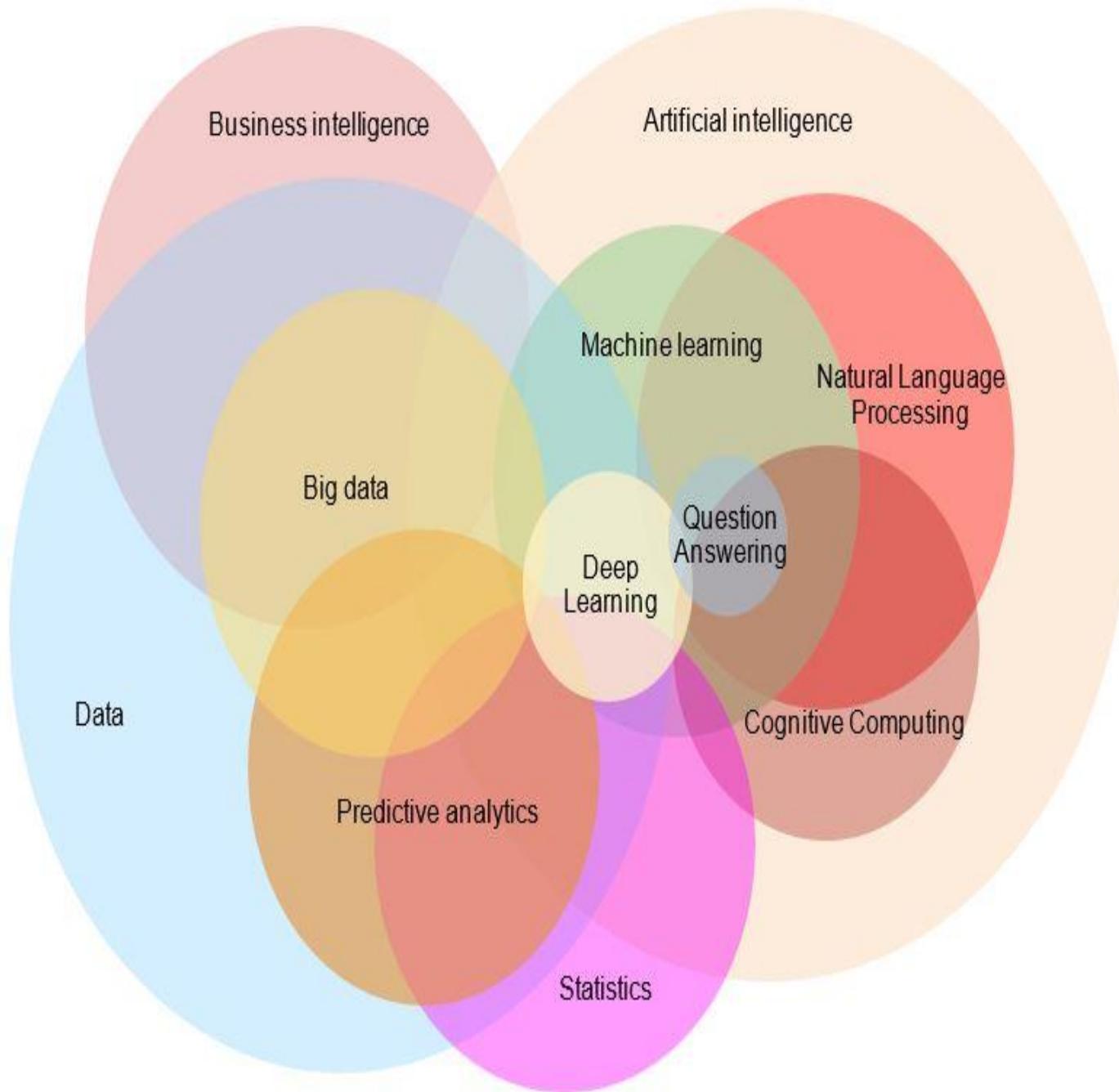
HEY, WHAT'S GOING ON?

What is NLP?

Natural language processing (NLP) is an area of computer science and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyze large amounts of natural language data



Where
does NLP
fit in?



Where
does NLP
fit in?



THINK

DINK

JOY

PIENSE



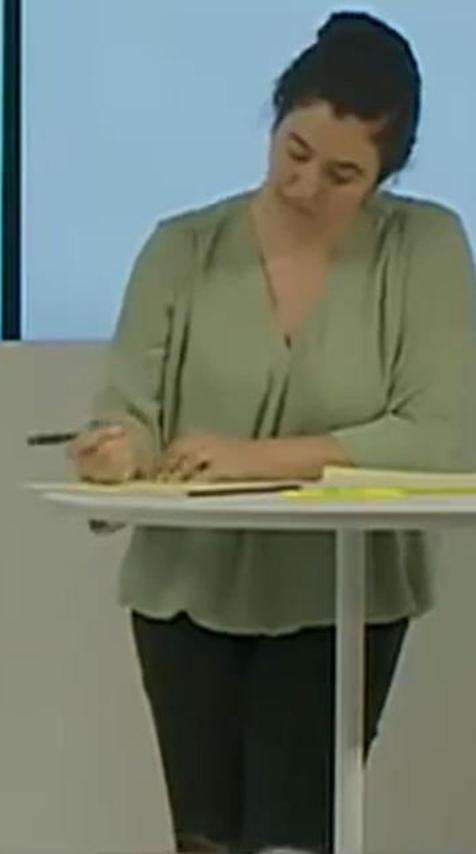
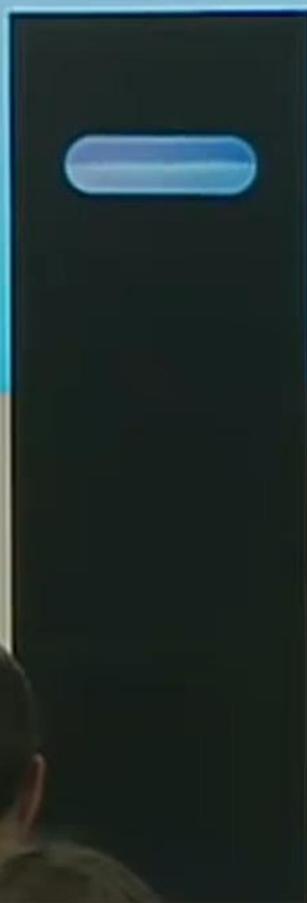


03:14:13



Project Debater

Noa Ovadia



VARNEY
& CO.

ROBOT DEBATES HUMAN

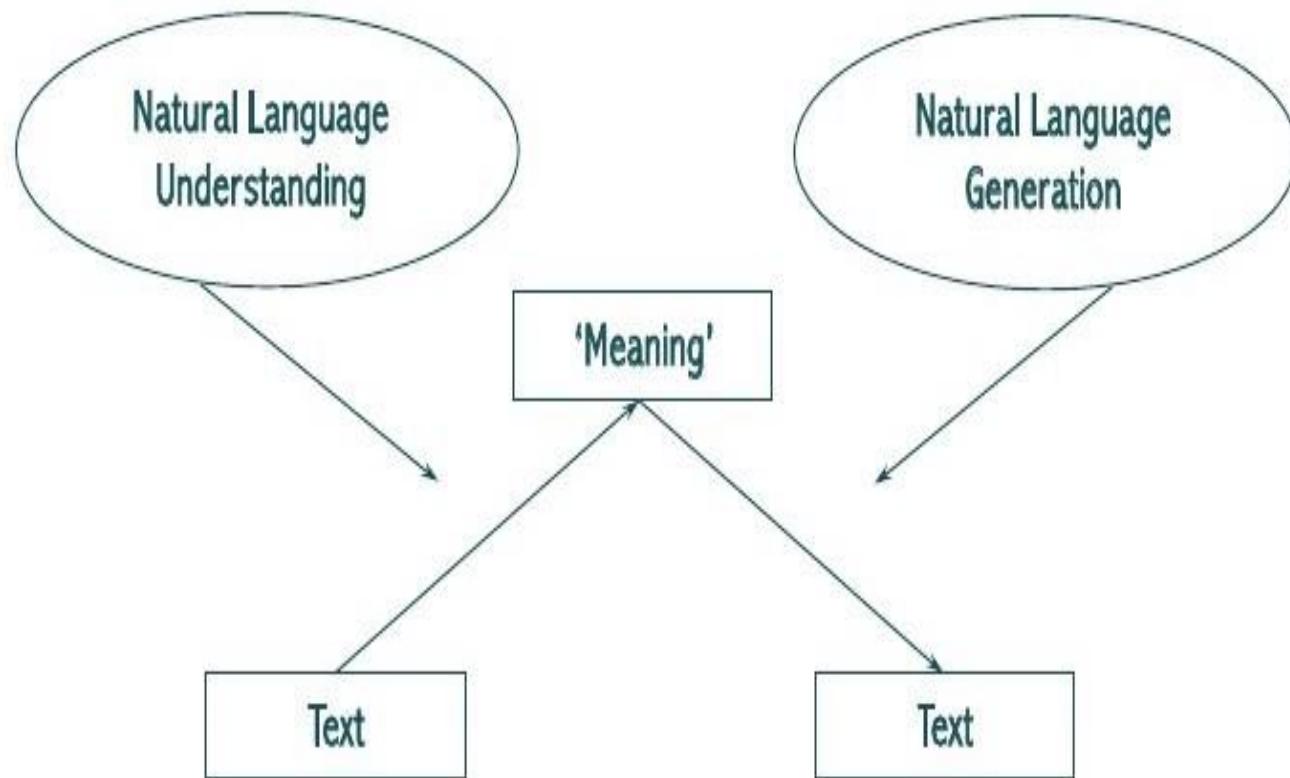
@VarneyCo



@FOXBUSINESS

Components of NLP

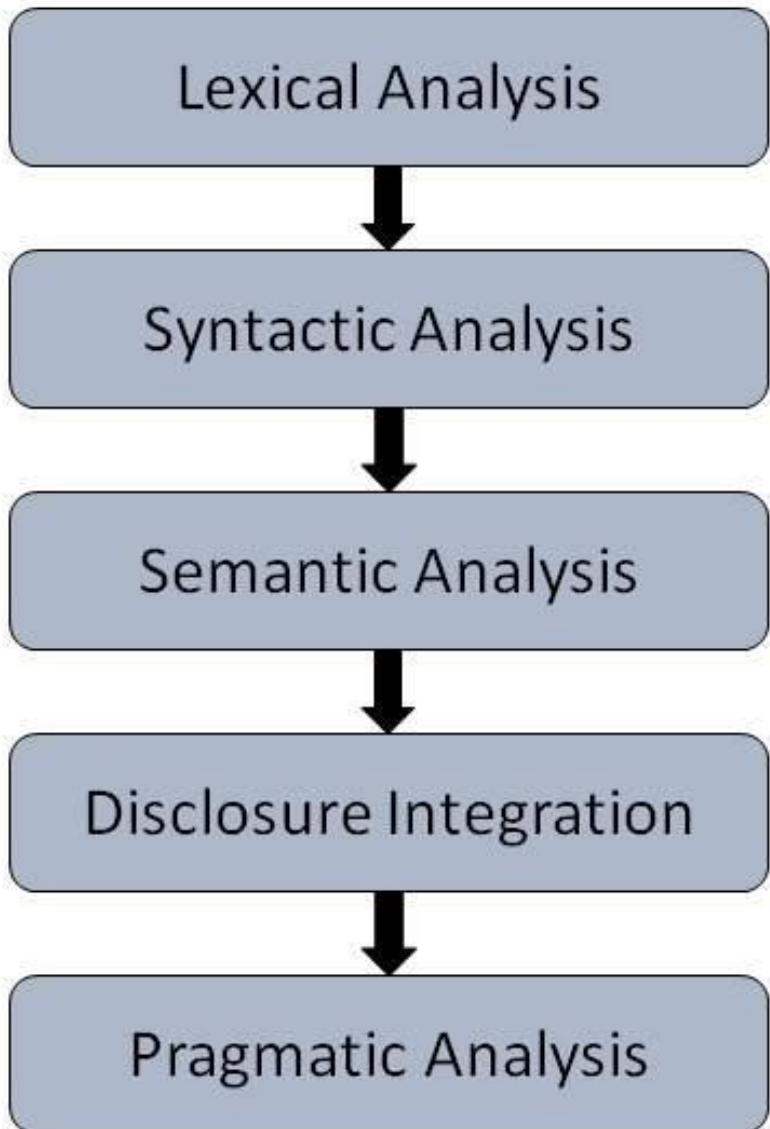
NLP = NLU + NLG



Components of NLP

Stages of NLP

Stages of NLP



Question Answering

- ▶ Won Jeopardy on February 16, 2011!

Question Answering

- ▶ Won Jeopardy on February 16, 2011!

WILLIAM WILKINSON'S
“AN ACCOUNT OF THE PRINCIPALITIES OF
WALLACHIA AND MOLDOVIA”
INSPIRED THIS AUTHOR'S
MOST FAMOUS NOVEL

Question Answering

- ▶ Won Jeopardy on February 16, 2011!

WILLIAM WILKINSON'S
“AN ACCOUNT OF THE PRINCIPALITIES OF
WALLACHIA AND MOLDOVIA”
INSPIRED THIS AUTHOR'S
MOST FAMOUS NOVEL



Bram Stoker

Information Extraction

Subject: **curriculum meeting**

Date: January 15, 2017

To: Bhavik Gandhi

Hi Bhavik, we've now scheduled the curriculum meeting.

It will be in Room 159 tomorrow from 10:00-11:30.

-Ravi

Information Extraction

Subject: **curriculum meeting**

Date: January 15, 2017

To: Bhavik Gandhi

Hi Bhavik, we've now scheduled the curriculum meeting.

It will be in Room 159 tomorrow from 10:00-11:30.

-Ravi



Create new Calendar entry

Information Extraction

Subject: **curriculum meeting**

Date: January 15, 2017

To: Bhavik Gandhi

Event: Curriculum mtg
Date: Jan-16-2017
Start: 10:00am
End: 11:30am
Where: Room 159

Hi Bhavik, we've now scheduled the curriculum meeting.

It will be in Room 159 tomorrow from 10:00-11:30.

-Ravi



Create new Calendar entry

Sentiment Analysis



Sentiment Analysis



Attributes:
zoom
affordability
size and weight
flash
ease of use

Sentiment Analysis



Size and weight

Attributes:
zoom
affordability
size and weight
flash
ease of use

Sentiment Analysis



Size and weight

- ▶ nice and compact to carry!
 - ▶ since the camera is small and light, I won't need to carry around those heavy, bulky professional cameras either!
 - ▶ the camera feels flimsy, is plastic and very light in weight you have to be very delicate in the handling of this camera

Attributes:
zoom
affordability
size and weight
flash
ease of use

Sentiment Analysis



Size and weight

- ✓ ▶ nice and compact to carry!
since the camera is small and light, I
won't need to carry around those heavy,
bulky professional cameras either!
- ▶ the camera feels flimsy, is plastic and
very light in weight you have to be very
delicate in the handling of this camera

Attributes:
zoom
affordability
size and weight
flash
ease of use

Sentiment Analysis



Size and weight

- ✓ ▶ nice and compact to carry!
since the camera is small and light, I
won't need to carry around those heavy,
bulky professional cameras either!
- ✓ ▶ the camera feels flimsy, is plastic and
very light in weight you have to be very
delicate in the handling of this camera

Attributes:
zoom
affordability
size and weight
flash
ease of use

Sentiment Analysis



Size and weight

- ✓ ▶ nice and compact to carry!
since the camera is small and light, I
won't need to carry around those heavy,
bulky professional cameras either!
- ✓ ▶ the camera feels flimsy, is plastic and
very light in weight you have to be very
delicate in the handling of this camera
- ✗

Attributes:
zoom
affordability
size and weight
flash
ease of use

Sentiment Analysis

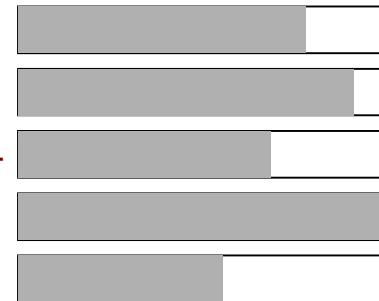


Size and weight

- ✓ ▶ nice and compact to carry!
since the camera is small and light, I
won't need to carry around those heavy,
bulky professional cameras either!
- ✓ ▶ the camera feels flimsy, is plastic and
very light in weight you have to be very
delicate in the handling of this camera
- ✗

Attributes:

zoom
affordability
size and weight
flash
ease of use



Machine Translation

Machine Translation

- ▶ Fully automatic

Machine Translation

- ▶ Fully automatic

Enter Source Text:

Translation from Stanford's *Phrasal*:

Machine Translation

- ▶ Fully automatic

Enter Source Text:

这不过是一个时间的问题.

Translation from Stanford's *Phrasal*:

Machine Translation

► Fully automatic

Enter Source Text:

这不过是一个时间的问题.

Translation from Stanford's *Phrasal*:

This is only a matter of time.

Machine Translation

- ▶ Fully automatic

Enter Source Text:

这不过是一个时间的问题.

- ▶ Helping human translators

Translation from Stanford's *Phrasal*:

This is only a matter of time.

Machine Translation

► Fully automatic

Enter Source Text:

这不过是一个时间的问题。

Translation from Stanford's *Phrasal*:

This is only a matter of time.

► Helping human translators

Enter Source Text:

عرض الرئيس اللبناني إميل لحود لـ#حملة عنيفة في مجلس التراب الذي انعقد أمس في جلسة تشريعية علنية تحررت إلى "محاكمة" لـ#رئيس الجمهورية علي موقت به من المحكمة الدولية و "الملحوظات" التي ادلّي بها #+ها حول هذا الموضوع.

Enter Translation:

lebanese |

- president
- suffered
- exposed
- president emile
- before
- presented
- offer

Done!

Mostly Solved



Mostly Solved



Mostly Solved

Spam detection

Let's go to Agra!	✓
Buy V1AGRA ...	✗

Part-of-speech (POS) tagging

ADJ ADJ NOUN VERB ADV
Colorless green ideas sleep furiously.

Mostly Solved

Spam detection

Let's go to Agra!

✓

Buy V1AGRA ...

✗

Part-of-speech (POS) tagging

ADJ ADJ NOUN VERB ADV

Colorless green ideas sleep furiously.

Named entity recognition (NER)

PERSON ORG LOC

Einstein met with UN officials in Princeton

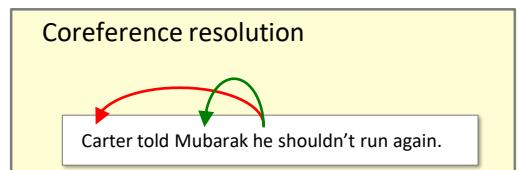
Somewhat Solved



Somewhat Solved



Somewhat Solved



Somewhat Solved

Sentiment analysis

Best roast chicken in San Francisco! 

The waiter ignored us for 20 minutes. 

Word sense disambiguation

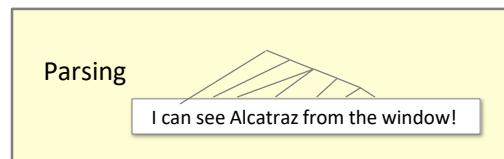
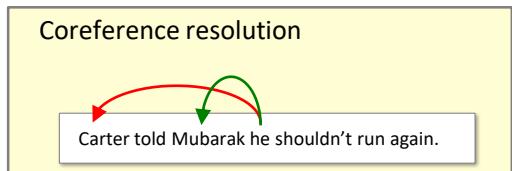
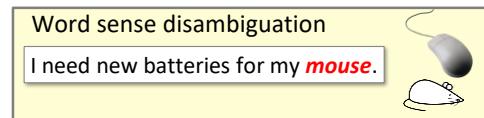
I need new batteries for my **mouse**. 

Coreference resolution

Carter told Mubarak he shouldn't run again.



Somewhat Solved



Somewhat Solved

Sentiment analysis

Best roast chicken in San Francisco!

The waiter ignored us for 20 minutes.

Word sense disambiguation

I need new batteries for my **mouse**.

Machine translation (MT)

第13届上海国际电影节开幕...

The 13th Shanghai International Film Festival...

Coreference resolution

Carter told Mubarak he shouldn't run again.

Parsing

I can see Alcatraz from the window!

Somewhat Solved

Sentiment analysis

Best roast chicken in San Francisco!

The waiter ignored us for 20 minutes.

Word sense disambiguation

I need new batteries for my **mouse**.

Machine translation (MT)

第13届上海国际电影节开幕...

The 13th Shanghai International Film Festival...

Coreference resolution

Carter told Mubarak he shouldn't run again.

Parsing

I can see Alcatraz from the window!

Information extraction (IE)

You're invited to our dinner party, Friday May 27 at 8:30

Party
May
27
[add](#)

Still very hard

Still very hard

Question answering (QA)

Q. How effective is ibuprofen in reducing fever in patients with acute febrile illness?

Still very hard

Question answering (QA)

Q. How effective is ibuprofen in reducing fever in patients with acute febrile illness?

Paraphrase

XYZ acquired ABC yesterday

ABC has been taken over by XYZ

Still very hard

Question answering (QA)

Q. How effective is ibuprofen in reducing fever in patients with acute febrile illness?

Paraphrase

XYZ acquired ABC yesterday

ABC has been taken over by XYZ

Summarization

The Dow Jones is up

Housing prices rose

The S&P500 jumped

Economy is good

Still very hard

Question answering (QA)

Q. How effective is ibuprofen in reducing fever in patients with acute febrile illness?

Paraphrase

XYZ acquired ABC yesterday

ABC has been taken over by XYZ

Summarization

The Dow Jones is up

Housing prices rose

The S&P500 jumped



Economy is good

Dialog

Where is Citizen Kane playing in SF?

Castro Theatre at 7:30. Do you want a ticket?



Ambiguity makes NLP hard: “Crash blossoms”

Violinist Linked to JAL Crash Blossoms



Ambiguity makes NLP hard: “Crash blossoms”

Violinist Linked to JAL Crash Blossoms
Teacher Strikes Idle Kids



Ambiguity makes NLP hard: “Crash blossoms”

Violinist Linked to JAL Crash Blossoms
Teacher Strikes Idle Kids
Red Tape Holds Up New Bridges
Hospitals Are Sued by 7 Foot Doctors
Juvenile Court to Try Shooting Defendant
Local High School Dropouts Cut in Half



Ambiguity is Pervasive

New York Times headline (17 May 2000)

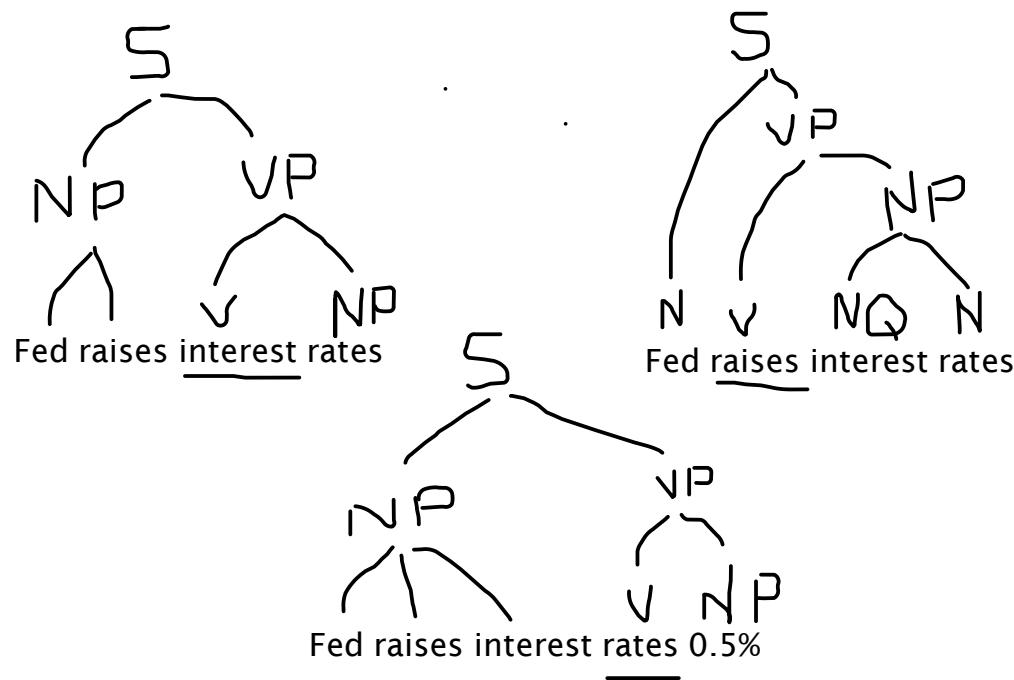
Fed raises interest rates

Fed raises interest rates

Fed raises interest rates 0.5%

Ambiguity is Pervasive

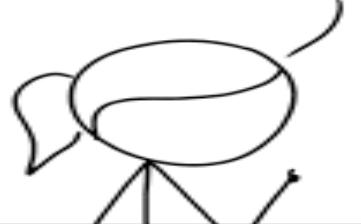
New York Times headline (17 May 2000)



...ANYWAY, I
COULD CARE LESS.



I THINK YOU MEAN YOU
~~COULDNT~~ CARE LESS.
SAYING YOU ~~COULD~~ CARE
LESS IMPLIES YOU CARE
AT LEAST SOME AMOUNT.



I DUNNO.



WE'RE THESE UNBELIEVABLY
COMPLICATED BRAINS DRIFTING
THROUGH A VOID, TRYING IN
VAIN TO CONNECT WITH ONE
ANOTHER BY BLINDLY FLINGING
WORDS OUT INTO THE DARKNESS.



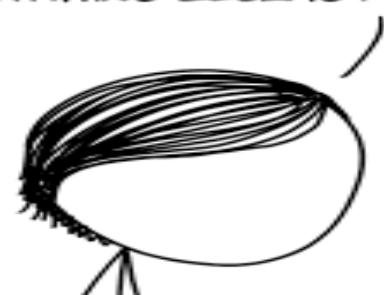
EVERY CHOICE OF PHRASING AND
SPELLING AND TONE AND TIMING
CARRIES COUNTLESS SIGNALS AND
CONTEXTS AND SUBTEXTS AND MORE,
AND EVERY LISTENER INTERPRETS
THOSE SIGNALS IN THEIR OWN WAY.
LANGUAGE ISN'T A FORMAL SYSTEM.
LANGUAGE IS GLORIOUS CHAOS.



YOU CAN NEVER KNOW FOR SURE WHAT
ANY WORDS WILL MEAN TO ANYONE.

ALL YOU CAN DO IS TRY TO GET BETTER AT
GUESSING HOW YOUR WORDS AFFECT PEOPLE,
SO YOU CAN HAVE A CHANCE OF FINDING THE
ONES THAT WILL MAKE THEM FEEL SOMETHING
LIKE WHAT YOU WANT THEM TO FEEL.

EVERYTHING ELSE IS POINTLESS.



I ASSUME YOU'RE GIVING ME TIPS ON
HOW YOU INTERPRET WORDS BECAUSE
YOU WANT ME TO FEEL LESS ALONE.
IF SO, THEN THANK YOU.
THAT MEANS A LOT.



BUT IF YOU'RE JUST RUNNING MY
SENTENCES PAST SOME MENTAL
CHECKLIST SO YOU CAN SHOW
OFF HOW WELL YOU KNOW IT,



THEN I COULD
CARE LESS.



What else makes NLP hard?

What else makes NLP hard?

non-standard English

Great job @justinbieber! Were SOO PROUD of what youve accomplished! U taught us 2 #neversaynever & you yourself should never give up either ❤

What else makes NLP hard?

non-standard English

Great job @justinbieber! Were SOO PROUD of what youve accomplished! U taught us 2 #neversaynever & you yourself should never give up either ❤

segmentation issues

the New York-New Haven Railroad
the New York-New Haven Railroad

What else makes NLP hard?

non-standard English

Great job @justinbieber! Were SOO PROUD of what youve accomplished! U taught us 2 #neversaynever & you yourself should never give up either ❤️

segmentation issues

the New York-New Haven Railroad
the New York-New Haven Railroad

idioms

dark horse
get cold feet
lose face
throw in the towel

What else makes NLP hard?

non-standard English

Great job @justinbieber! Were SOO PROUD of what youve accomplished! U taught us 2 #neversaynever & you yourself should never give up either ❤️

segmentation issues

the New York-New Haven Railroad
the New York-New Haven Railroad

idioms

dark horse
get cold feet
lose face
throw in the towel

neologisms

unfriend
Retweet
bromance

What else makes NLP hard?

non-standard English

Great job @justinbieber! Were SOO PROUD of what youve accomplished! U taught us 2 #neversaynever & you yourself should never give up either ❤️

segmentation issues

the New York-New Haven Railroad
the New York-New Haven Railroad

idioms

dark horse
get cold feet
lose face
throw in the towel

neologisms

unfriend
Retweet
bromance

world knowledge

Mary and Sue are sisters.
Mary and Sue are mothers.

What else makes NLP hard?

non-standard English

Great job @justinbieber! Were SOO PROUD of what youve accomplished! U taught us 2 #neversaynever & you yourself should never give up either ❤️

segmentation issues

the New York-New Haven Railroad
the New York-New Haven Railroad

idioms

dark horse
get cold feet
lose face
throw in the towel

neologisms

unfriend
Retweet
bromance

world knowledge

Mary and Sue are sisters.
Mary and Sue are mothers.

tricky entity names

Where is *A Bug's Life* playing ...
Let It Be was recorded ...
... a mutation on the *for* gene
...

What else makes NLP hard?

non-standard English

Great job @justinbieber! Were SOO PROUD of what youve accomplished! U taught us 2 #neversaynever & you yourself should never give up either ❤️

segmentation issues

the New York-New Haven Railroad
the New York-New Haven Railroad

idioms

dark horse
get cold feet
lose face
throw in the towel

neologisms

unfriend
Retweet
bromance

world knowledge

Mary and Sue are sisters.
Mary and Sue are mothers.

tricky entity names

Where is *A Bug's Life* playing ...
Let It Be was recorded ...
... a mutation on the *for* gene
...

But that's what makes it fun!

Recent Major Advancements



Recent Major Advancements



ULMFiT
(Jan-18)

Recent Major Advancements

ULMFiT
(Jan-18)

ELMo
(Mar-18)

Recent Major Advancements

ULMFiT
(Jan-18)

ELMo
(Mar-18)

BERT
(Oct-18)

Recent Major Advancements

ULMFiT
(Jan-18)

ELMo
(Mar-18)

BERT
(Oct-18)

Transformer
– XL
(Jan-19)

Recent Major Advancements

ULMFiT
(Jan-18)

ELMo
(Mar-18)

BERT
(Oct-18)

Transformer
– XL
(Jan-19)

MT-DNN
(Feb-19)

Recent Major Advancements

ULMFiT
(Jan-18)

ELMo
(Mar-18)

BERT
(Oct-18)

Transformer
– XL
(Jan-19)

MT-DNN
(Feb-19)

MT-DNN
Ensemble
(Apr-19)

XL-Net
(Jun-19)

Roberta
(Aug-19)

What's up in NLP?

HMM, THIS SEEMS LIKE SOME REAL DIFFICULT STUFF!!

Course Logistics

YES, HOW WILL THIS COURSE WORK?

Course Outline

Introduction to NLP

Basic Text Processing

Edit Distance

Language Models

Spell Correction

Generative Modelling

Sentiment Analysis

Max Entropy Models

Named Entity Recognition

Topic Modelling

Relation Extraction

Part of Speech Tagging

Constituency Parsing

Dependency Parsing

Information Retrieval

Word Embeddings

Question Answering

Coreference Resolution

Word Sense Disambiguation

Speech Analysis

Machine Translation

Summarization

Sequence Models

Current Sota

Course Evaluation



Class Participation & Quizzes – 15%



Assignments – 30%



Project – 20%



Mid-Term Test – 15%



Final Test – 20%

Tentative Timelines

Time	Dates	Topic	Assignments
11 AM - 1 PM	28-Jul	Intro to NLP, Stemming, Lemmatization, Edit Distance Language Models, Spell Correction, Sentiment Analysis	
2 PM - 5 PM	28-Jul		Assignment 1
10 AM - 1 PM	4-Aug	Word and Text Representations, Topic Modelling	
2 PM - 5 PM	4-Aug	Named Entity Recognition, Sequence Models	Assignment 2
10 AM - 1 PM	10-AUG	Relation Extraction, Part of Speech Tagging	
2 PM - 5 PM	10-AUG	Question Answering, Anaphora	Assignment 3
10 AM - 1 PM	24-AUG	Mid-Term Test	
2 PM - 5 PM	24-AUG	Class Project Mid-Point Presentation	Assignment 4
10 AM - 1 PM	07-SEPT	Constituency Parsing	
2 PM - 5 PM	07-SEPT	Dependency Parsing	Assignment 5
10 AM - 1 PM	21-SEPT	Chatbots, WSD, STT, TTS	
2 PM - 5 PM	21-SEPT	Machine Translation, Attention Mechanism	Assignment 6
10 AM - 1 PM	06-Oct	Final Project Presentation	
2 PM - 5 PM	06-Oct	NLG, Other Topics in NLP	
10 AM - 1 PM	13-Oct	Final Test	

Class Structure

Class Structure

- ▶ Pre-Reading Quiz (15 mins)

Class Structure

- ▶ Pre-Reading Quiz (15 mins)
- ▶ Quick Review (15 - 30 mins)

Class Structure

- ▶ Pre-Reading Quiz (15 mins)
- ▶ Quick Review (15 - 30 mins)
- ▶ Modules (90 – 120 mins)
- ▶ Code Along (15 – 30 mins)

Class Structure

- ▶ Pre-Reading Quiz (15 mins)
- ▶ Quick Review (15 - 30 mins)
- ▶ Modules (90 – 120 mins)
- ▶ Code Along (15 – 30 mins)
- ▶ Lab Sessions (2 – 2.5 hrs)
- ▶ Assignments (3 - 4 hrs)
- ▶ Pre-Reads (30 - 60 mins)
- ▶ Lecture Review (2 - 3 hrs)
- ▶ Project Work (1 - 2 hrs)

Other Guidelines

Other Guidelines

- ▶ Assignment Submissions will be due Sunday night 23:55 the week after the assignment is shared
- ▶ Assignments are required to be completed individually without any collaboration
- ▶ In assignment evaluation, points will be given for
 - ▶ Quality of code
 - ▶ Accuracy of Models / Techniques on Unseen Data
 - ▶ Explanation and Documentation of Approach

Other Guidelines

- ▶ Assignment Submissions will be due Sunday night 23:55 the week after the assignment is shared
- ▶ Assignments are required to be completed individually without any collaboration
- ▶ In assignment evaluation, points will be given for
 - ▶ Quality of code
 - ▶ Accuracy of Models / Techniques on Unseen Data
 - ▶ Explanation and Documentation of Approach
- ▶ Projects can be taken up in groups of no more than 3 people
- ▶ Projects should have a demo component
- ▶ Projects should involve NLP as the focus area
- ▶ Project should be approved by the instructor

Other Guidelines

- ▶ Assignment Submissions will be due Sunday night 23:55 the week after the assignment is shared
- ▶ Assignments are required to be completed individually without any collaboration
- ▶ In assignment evaluation, points will be given for
 - ▶ Quality of code
 - ▶ Accuracy of Models / Techniques on Unseen Data
 - ▶ Explanation and Documentation of Approach
- ▶ Projects can be taken up in groups of no more than 3 people
- ▶ Projects should have a demo component
- ▶ Projects should involve NLP as the focus area
- ▶ Project should be approved by the instructor
- ▶ Tests will be 3 hr open book coding tests, building NLP models is expected

Other Guidelines

- ▶ Assignment Submissions will be due Sunday night 23:55 the week after the assignment is shared
- ▶ Assignments are required to be completed individually without any collaboration
- ▶ In assignment evaluation, points will be given for
 - ▶ Quality of code
 - ▶ Accuracy of Models / Techniques on Unseen Data
 - ▶ Explanation and Documentation of Approach
- ▶ Projects can be taken up in groups of no more than 3 people
- ▶ Projects should have a demo component
- ▶ Projects should involve NLP as the focus area
- ▶ Project should be approved by the instructor
- ▶ Tests will be 3 hr open book coding tests, building NLP models is expected
- ▶ In-class Quizzes will consist of multiple-choice questions on prev classes and reading material

Course Logistics

OK, NOW I AM READY TO ACE THIS COURSE!!

INTRODUCTION

SO NOW WE HAVE GREETED NLP!

BASIC TEXT PROCESSING

YEAH, LET'S START WITH THE BASICS..

Regular Expressions

MY EXPRESSIONS ARE ALWAYS REGULAR!

Regular Expressions

Regular Expressions

- ▶ A formal language for specifying text strings

Regular Expressions

- ▶ A formal language for specifying text strings
- ▶ How can we search for any of these?
 - ▶ woodchuck
 - ▶ woodchucks
 - ▶ Woodchuck
 - ▶ Woodchucks



Disjunctions

Disjunctions

- ▶ Letters inside square brackets []

Pattern	Matches
[wW] oodchuck	Woodchuck, woodchuck
[1234567890]	Any digit

Disjunctions

- ▶ Letters inside square brackets []

Pattern	Matches
[wW]oodchuck	Woodchuck, woodchuck
[1234567890]	Any digit

- ▶ Ranges [A-Z]

Pattern	Matches	
[A-Z]	An upper case letter	Drenched Blossoms
[a-z]	A lower case letter	my beans were impatient
[0-9]	A single digit	Chapter 1: Down the Rabbit Hole

Negation in Disjunction

Negation in Disjunction

- ▶ Negations [^Ss]
 - ▶ Carat means negation only when first in []

Negation in Disjunction

- ▶ Negations [^Ss]
 - ▶ Carat means negation only when first in []

Pattern	Matches	
[^A-Z]	Not an upper case letter	Oyfn pripetchik
[^Ss]	Neither 'S' nor 's'	I have no exquisite reason"
[^e^]	Neither e nor ^	Look <u>here</u>
a^b	The pattern a carat b	Look up <u>a^b</u> now

More Disjunctions

More Disjunctions

- Woodchuck is another name for groundhog!

More Disjunctions

- ▶ Woodchuck is another name for groundhog!
- ▶ The pipe | for disjunction

More Disjunctions

- ▶ Woodchuck is another name for groundhog!
- ▶ The pipe | for disjunction

Pattern	Matches
groundhog woodchuck	
yours mine	yours mine
a b c	= [abc]
[gG] roundhog [Ww] oodchuck	



More Regular Expressions

More Regular Expressions

?

*

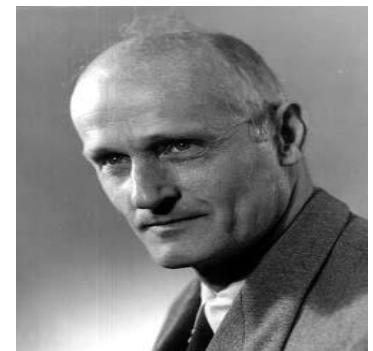
+

.

More Regular Expressions

? * + .

Pattern	Matches	Examples
colou?r	Optional previous char	color colour
oo*h!	0 or more of previous char	oh! ooh! oooh! ooooh!
o+h!	1 or more of previous char	oh! ooh! oooh! ooooh!
baa+		baa baaa baaaa baaaaa
beg.n		begin begun begun beg3n



Stephen C Kleene

Kleene *, Kleene +

Regular Expressions: Anchors ^ \$

Regular Expressions: Anchors ^ \$

Pattern

Matches

^ [A-Z]

Palo Alto

^ [^A-Za-z]

1 "Hello"

\.\$

The end._

.\$

The end? The end!

Example

- ▶ Find me all instances of the word “the” in a text.

Example

- ▶ Find me all instances of the word “the” in a text.

the

Example

- ▶ Find me all instances of the word “the” in a text.

the

Misses capitalized examples

Example

- ▶ Find me all instances of the word “the” in a text.

the

Misses capitalized examples

[tT] he

Example

- ▶ Find me all instances of the word “the” in a text.

the

Misses capitalized examples

[tT] he

Incorrectly returns other or theology

Example

- ▶ Find me all instances of the word “the” in a text.

the

Misses capitalized examples

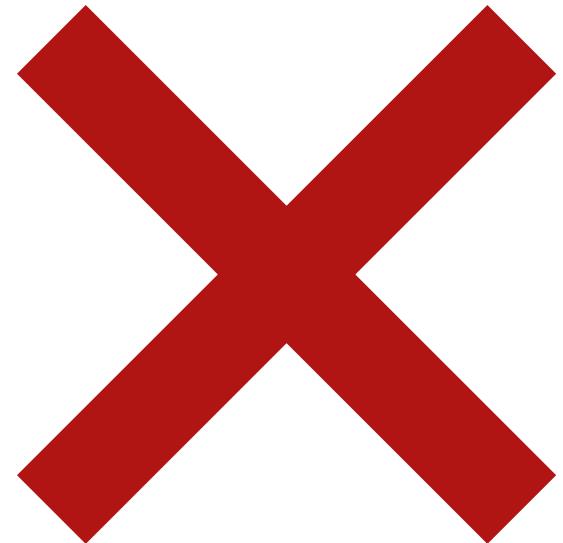
[tT] he

Incorrectly returns other or theology

[^a-zA-Z] [tT] he [^a-zA-Z]

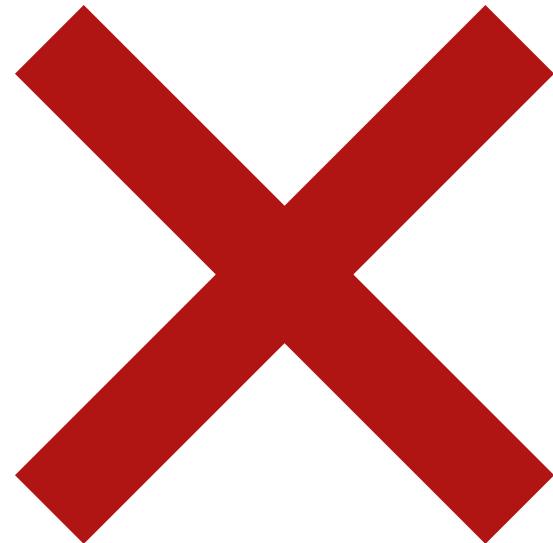
Errors

- ▶ The process we just went through was based on fixing two kinds of errors



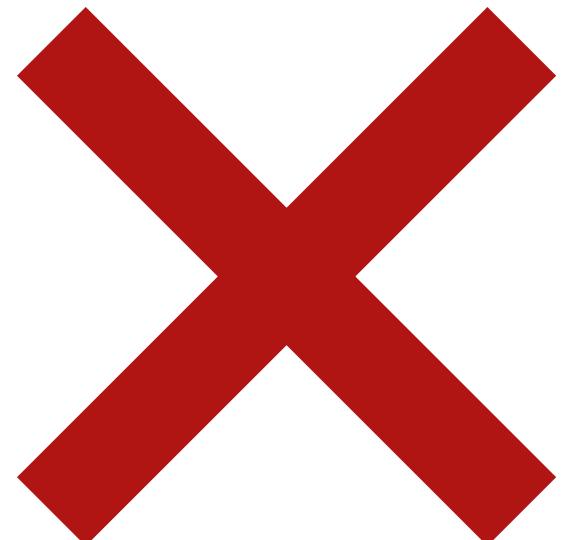
Errors

- ▶ The process we just went through was based on fixing two kinds of errors
 - ▶ Matching strings that we should not have matched (there, then, other)
 - ▶ False positives (Type I)



Errors

- ▶ The process we just went through was based on fixing two kinds of errors
 - ▶ Matching strings that we should not have matched (there, then, other)
 - ▶ False positives (Type I)
 - ▶ Not matching things that we should have matched (The)
 - ▶ False negatives (Type II)



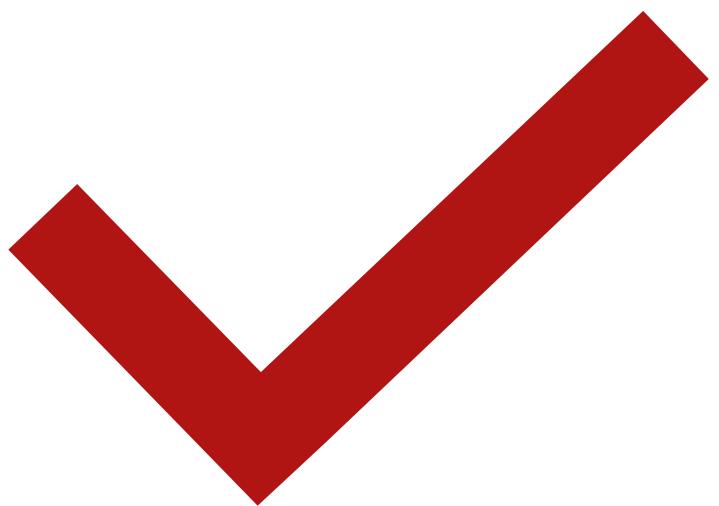
Errors cont.

- ▶ In NLP we are always dealing with these kinds of errors.

Errors cont.

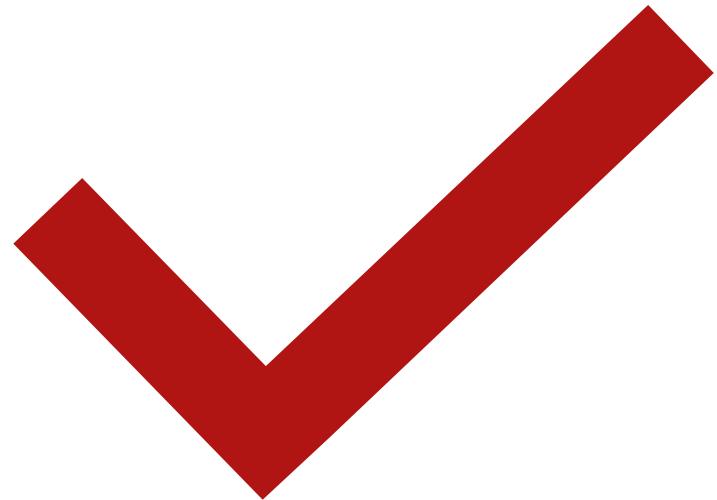
- ▶ In NLP we are always dealing with these kinds of errors.
- ▶ Reducing the error rate for an application often involves two antagonistic efforts:
 - ▶ Increasing accuracy or precision (minimizing false positives)
 - ▶ Increasing coverage or recall (minimizing false negatives).

Summary



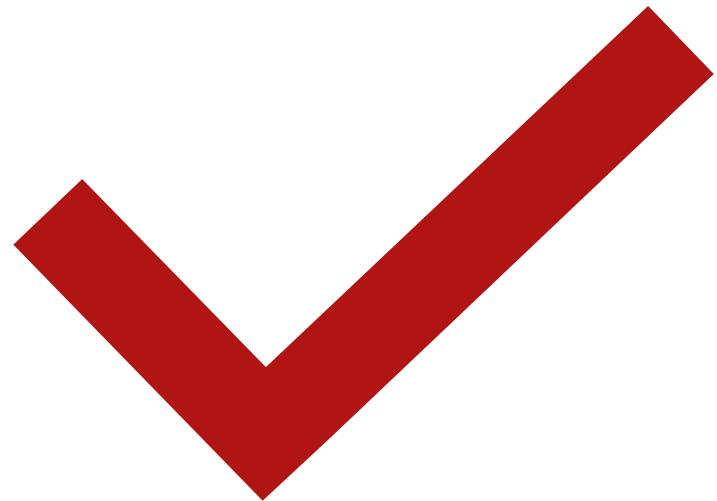
Summary

- ▶ Regular expressions play a surprisingly large role



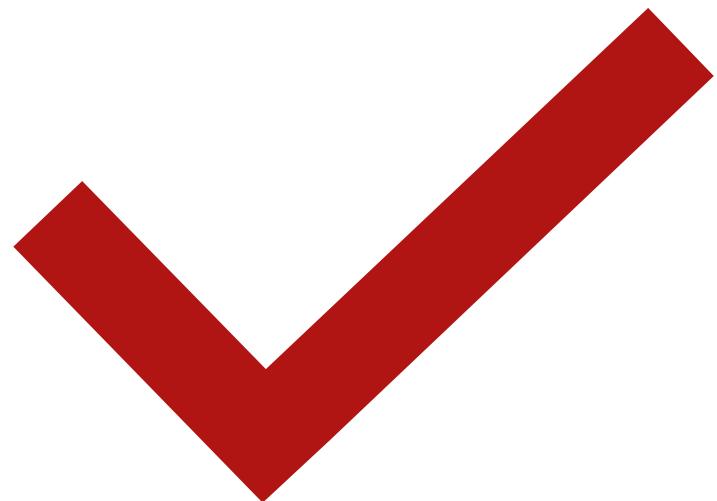
Summary

- ▶ Regular expressions play a surprisingly large role
 - ▶ Sophisticated sequences of regular expressions are often the first model for any text processing task



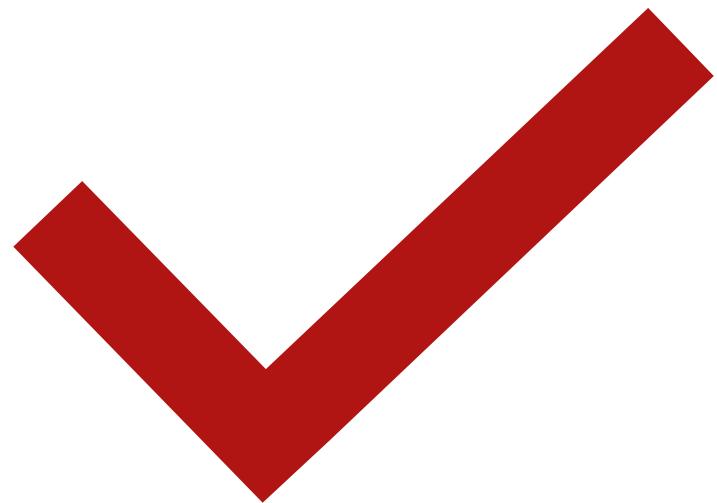
Summary

- ▶ Regular expressions play a surprisingly large role
 - ▶ Sophisticated sequences of regular expressions are often the first model for any text processing task
- ▶ For many hard tasks, we use machine learning classifiers



Summary

- ▶ Regular expressions play a surprisingly large role
 - ▶ Sophisticated sequences of regular expressions are often the first model for any text processing task
- ▶ For many hard tasks, we use machine learning classifiers
 - ▶ But regular expressions are used as features in the classifiers
 - ▶ Can be very useful in capturing generalizations



Regular Expressions

NOT ALL THAT REGULAR, EH?

Word Tokenization

WHY SHOULD WORDS HAVE TOKENS?

Text Normalization

- ▶ Every NLP task needs to do text normalization:

Text Normalization

- ▶ Every NLP task needs to do text normalization:
 1. Segmenting/tokenizing words in running text

Text Normalization

- ▶ Every NLP task needs to do text normalization:
 1. Segmenting/tokenizing words in running text
 2. Normalizing word formats

Text Normalization

- ▶ Every NLP task needs to do text normalization:
 1. Segmenting/tokenizing words in running text
 2. Normalizing word formats
 3. Segmenting sentences in running text

How many words?

- ▶ I do uh main- mainly business data processing

How many words?

- ▶ I do uh main- mainly business data processing
 - ▶ Fragments, filled pauses

How many words?

- ▶ I do uh main- mainly business data processing
 - ▶ Fragments, filled pauses
- ▶ Seuss's cat in the hat is different from other cats!

How many words?

- ▶ I do uh main- mainly business data processing
 - ▶ Fragments, filled pauses
- ▶ Seuss's cat in the hat is different from other cats!
 - ▶ **Lemma:** same stem, part of speech, rough word sense
 - ▶ cat and cats = same lemma

How many words?

- ▶ I do uh main- mainly business data processing
 - ▶ Fragments, filled pauses
- ▶ Seuss's cat in the hat is different from other cats!
 - ▶ **Lemma:** same stem, part of speech, rough word sense
 - ▶ cat and cats = same lemma
 - ▶ **Wordform:** the full inflected surface form
 - ▶ cat and cats = different wordforms

How many words?

they lay back on the San Francisco grass and looked at the stars and
their

How many words?

they lay back on the San Francisco grass and looked at the stars and their

- ▶ **Type:** an element of the vocabulary.
- ▶ **Token:** an instance of that type in running text.

How many words?

they lay back on the San Francisco grass and looked at the stars and their

- ▶ **Type:** an element of the vocabulary.
- ▶ **Token:** an instance of that type in running text.
- ▶ How many?
 - ▶ 15 tokens (or 14)

How many words?

they lay back on the San Francisco grass and looked at the stars and their

- ▶ **Type:** an element of the vocabulary.
- ▶ **Token:** an instance of that type in running text.
- ▶ How many?
 - ▶ 15 tokens (or 14)
 - ▶ 13 types (or 12) (or 11?)

How many words?

How many words?

N = number of tokens

How many words?

N = number of tokens

V = vocabulary = set of types

$|V|$ is the size of the vocabulary

How many words?

N = number of tokens

V = vocabulary = set of types

$|V|$ is the size of the vocabulary

	Tokens = N	Types = V
Switchboard phone conversations	2.4 million	20 thousand
Shakespeare	884,000	31 thousand
Google N-grams	1 trillion	13 million

How many words?

N = number of tokens

V = vocabulary = set of types

$|V|$ is the size of the vocabulary

Church and Gale (1990): $|V| > O(N^{1/2})$

	Tokens = N	Types = V
Switchboard phone conversations	2.4 million	20 thousand
Shakespeare	884,000	31 thousand
Google N-grams	1 trillion	13 million

Simple Tokenization

- ▶ Given a text file, output the word tokens and their frequencies

Simple Tokenization

- ▶ Given a text file, output the word tokens and their frequencies

1945 A

72 AARON

19 ABBESS

5 ABBOT

... ...

Simple Tokenization

- Given a text file, output the word tokens and their frequencies

Change all non-alpha to newlines

1945 A

72 AARON

19 ABBESS

5 ABBOT

... ...

Simple Tokenization

- Given a text file, output the word tokens and their frequencies

Change all non-alpha to newlines

Sort in alphabetical order

1945 A

72 AARON

19 ABBESS

5 ABBOT

... ...

Simple Tokenization

- Given a text file, output the word tokens and their frequencies

Change all non-alpha to newlines

Sort in alphabetical order

1945 A

72 AARON

19 ABESSION

5 ABBOT

... ...

Merge and count each type

25 Aaron

6 Abate

1 Abates

5 Abbess

6 Abbey

3 Abbot

....

The first step: tokenizing

THE

SONNETS

by

William

Shakespeare

From

fairest

creatures

We

...

The second step: sorting

A

A

A

A

A

A

A

A

A

...

More counting

More counting

- ▶ Merging upper and lower case

More counting

- ▶ Merging upper and lower case
- ▶ Sorting the counts

More counting

- ▶ Merging upper and lower case
- ▶ Sorting the counts

23243	the
22225	i
18618	and
16339	to
15687	of
12780	a
12163	you
10839	my
10005	in
8954	d

More counting

- ▶ Merging upper and lower case
- ▶ Sorting the counts

23243	the
22225	i
18618	and
16339	to
15687	of
12780	a
12163	you
10839	my
10005	in
8954	d

What happened here?

Issues in Tokenization

Issues in Tokenization

- ▶ Finland's capital → Finland Finlands Finland's ?

Issues in Tokenization

- ▶ Finland's capital → Finland Finlands Finland's ?
- ▶ what're, I'm, isn't → What are, I am, is not

Issues in Tokenization

- ▶ Finland's capital → Finland Finlands Finland's ?
- ▶ what're, I'm, isn't → What are, I am, is not
- ▶ Hewlett-Packard → Hewlett Packard ?

Issues in Tokenization

- ▶ Finland's capital → Finland Finlands Finland's ?
- ▶ what're, I'm, isn't → What are, I am, is not
- ▶ Hewlett-Packard → Hewlett Packard ?
- ▶ state-of-the-art → state of the art ?

Issues in Tokenization

- ▶ Finland's capital → Finland Finlands Finland's ?
- ▶ what're, I'm, isn't → What are, I am, is not
- ▶ Hewlett-Packard → Hewlett Packard ?
- ▶ state-of-the-art → state of the art ?
- ▶ Lowercase → lower-case lowercase lower case ?

Issues in Tokenization

- ▶ Finland's capital → Finland Finlands Finland's ?
- ▶ what're, I'm, isn't → What are, I am, is not
- ▶ Hewlett-Packard → Hewlett Packard ?
- ▶ state-of-the-art → state of the art ?
- ▶ Lowercase → lower-case lowercase lower case ?
- ▶ San Francisco → one token or two?

Issues in Tokenization

- ▶ Finland's capital → Finland Finlands Finland's ?
- ▶ what're, I'm, isn't → What are, I am, is not
- ▶ Hewlett-Packard → Hewlett Packard ?
- ▶ state-of-the-art → state of the art ?
- ▶ Lowercase → lower-case lowercase lower case ?
- ▶ San Francisco → one token or two?
- ▶ m.p.h., PhD. → ??

Tokenization: language issues

Tokenization: language issues

- ▶ French
 - ▶ ***L'ensemble*** → one token or two?
 - ▶ ***L*** ? ***L'*** ? ***Le*** ?
 - ▶ Want ***L'ensemble*** to match with ***un ensemble***

Tokenization: language issues

- ▶ French
 - ▶ **L'ensemble** → one token or two?
 - ▶ **L** ? **L'** ? **Le** ?
 - ▶ Want **L'ensemble** to match with **un ensemble**
- ▶ German noun compounds are not segmented
 - ▶ **Lebensversicherungsgesellschaftsangestellter**
 - ▶ ‘life insurance company employee’
 - ▶ German information retrieval needs **compound splitter**

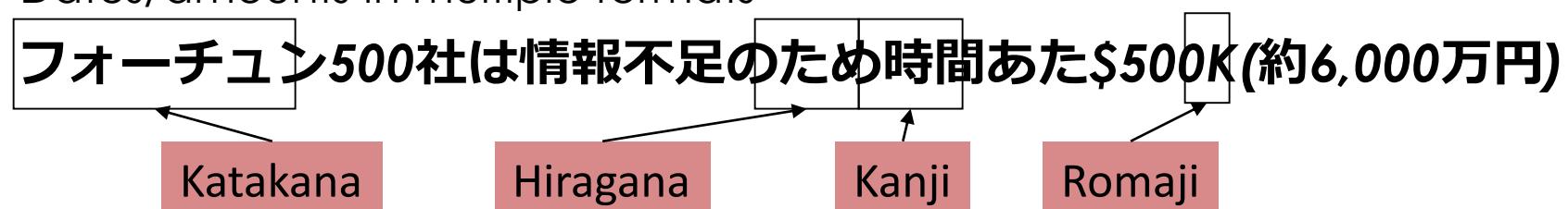
Tokenization: language issues

Tokenization: language issues

- ▶ Chinese and Japanese no spaces between words:
 - ▶ 莎拉波娃现在居住在美国东南部的佛罗里达。
 - ▶ 莎拉波娃 现在 居住 在 美国 东南部 的 佛罗里达
 - ▶ Sharapova now lives in US southeastern Florida

Tokenization: language issues

- ▶ Chinese and Japanese no spaces between words:
 - ▶ 莎拉波娃现在居住在美国东南部的佛罗里达。
 - ▶ 莎拉波娃 现在 居住 在 美国 东南部 的 佛罗里达
 - ▶ Sharapova now lives in US southeastern Florida
- ▶ Further complicated in Japanese, with multiple alphabets intermingled
 - ▶ Dates/amounts in multiple formats



End-user can express query entirely in hiragana!

Word Tokenization in Chinese

Word Tokenization in Chinese

- ▶ Also called **Word Segmentation**

Word Tokenization in Chinese

- ▶ Also called **Word Segmentation**
- ▶ Chinese words are composed of characters
 - ▶ Characters are generally 1 syllable and 1 morpheme.
 - ▶ Average word is 2.4 characters long.

Word Tokenization in Chinese

- ▶ Also called **Word Segmentation**
- ▶ Chinese words are composed of characters
 - ▶ Characters are generally 1 syllable and 1 morpheme.
 - ▶ Average word is 2.4 characters long.
- ▶ Standard baseline segmentation algorithm:
 - ▶ Maximum Matching (also called Greedy)

Maximum Matching Word Segmentation Algorithm

Maximum Matching Word Segmentation Algorithm

- ▶ Given a wordlist of Chinese, and a string.

Maximum Matching Word Segmentation Algorithm

- ▶ Given a wordlist of Chinese, and a string.
- 1) Start a pointer at the beginning of the string

Maximum Matching Word Segmentation Algorithm

- ▶ Given a wordlist of Chinese, and a string.
- 1) Start a pointer at the beginning of the string
 - 2) Find the longest word in dictionary that matches the string starting at pointer

Maximum Matching Word Segmentation Algorithm

- ▶ Given a wordlist of Chinese, and a string.
- 1) Start a pointer at the beginning of the string
 - 2) Find the longest word in dictionary that matches the string starting at pointer
 - 3) Move the pointer over the word in string

Maximum Matching Word Segmentation Algorithm

- ▶ Given a wordlist of Chinese, and a string.
- 1) Start a pointer at the beginning of the string
- 2) Find the longest word in dictionary that matches the string starting at pointer
- 3) Move the pointer over the word in string
- 4) Go to 2

Max-match segmentation illustration

▶ Thecatinthehat

Max-match segmentation illustration

- ▶ The cat in the hat
- ▶ The table down there

Max-match segmentation illustration

- ▶ The cat in the hat
- ▶ The table down there
- ▶ Doesn't generally work in English!

Max-match segmentation illustration

- ▶ The cat in the hat
- ▶ The table down there
- ▶ Doesn't generally work in English!
- ▶ But works astonishingly well in Chinese
 - ▶ 莎拉波娃现在居住在美国东南部的佛罗里达。
 - ▶ 莎拉波娃 现在 居住 在 美国 东南部 的 佛罗里达

Max-match segmentation illustration

- ▶ The cat in the hat
- ▶ The table down there
- ▶ Doesn't generally work in English!
- ▶ But works astonishingly well in Chinese
 - ▶ 莎拉波娃现在居住在美国东南部的佛罗里达。
 - ▶ 莎拉波娃 现在 居住 在 美国 东南部 的 佛罗里达
- ▶ Modern probabilistic segmentation algorithms even better

Max-match segmentation illustration

the cat in the hat

- ▶ The cat in the hat
- ▶ The table down there
- ▶ Doesn't generally work in English!
- ▶ But works astonishingly well in Chinese
 - ▶ 莎拉波娃现在居住在美国东南部的佛罗里达。
 - ▶ 莎拉波娃 现在 居住 在 美国 东南部 的 佛罗里达
- ▶ Modern probabilistic segmentation algorithms even better

Max-match segmentation illustration

- ▶ the cat in the hat
- ▶ the table down there
- ▶ Doesn't generally work in English!
- ▶ But works astonishingly well in Chinese
 - ▶ 莎拉波娃现在居住在美国东南部的佛罗里达。
 - ▶ 莎拉波娃 现在 居住 在 美国 东南部 的 佛罗里达
- ▶ Modern probabilistic segmentation algorithms even better

Max-match segmentation illustration

- ▶ Thecatinthehat the cat in the hat
 - ▶ Thetabledownthere the table down there
 - ▶ Doesn't generally work in English!
 - ▶ But works astonishingly well in Chinese
 - ▶ 莎拉波娃现在居住在美国东南部的佛罗里达。
 - ▶ 莎拉波娃 现在 居住 在 美国 东南部 的 佛罗里达
 - ▶ Modern probabilistic segmentation algorithms even better

Word Tokenization

THAT SEEMS SIMPLE!

Word Normalization

AREN'T WORDS ALREADY NORMAL?

Normalization

- ▶ Need to “normalize” terms



Normalization

- ▶ Need to “normalize” terms
 - ▶ Information Retrieval: indexed text & query terms must have same form.
 - ▶ We want to match ***U.S.A.*** and ***USA***



Normalization

- ▶ Need to “normalize” terms
 - ▶ Information Retrieval: indexed text & query terms must have same form.
 - ▶ We want to match **U.S.A.** and **USA**
 - ▶ We implicitly define equivalence classes of terms
 - ▶ e.g., deleting periods in a term



Normalization

- ▶ Need to “normalize” terms
 - ▶ Information Retrieval: indexed text & query terms must have same form.
 - ▶ We want to match **U.S.A.** and **USA**
- ▶ We implicitly define equivalence classes of terms
 - ▶ e.g., deleting periods in a term
- ▶ Alternative: asymmetric expansion:



Normalization

- ▶ Need to “normalize” terms
 - ▶ Information Retrieval: indexed text & query terms must have same form.
 - ▶ We want to match **U.S.A.** and **USA**
- ▶ We implicitly define equivalence classes of terms
 - ▶ e.g., deleting periods in a term
- ▶ Alternative: asymmetric expansion:
 - ▶ Enter: **window** Search: **window, windows**
 - ▶ Enter: **windows** Search: **Windows, windows, window**
 - ▶ Enter: **Windows** Search: **Windows**



Normalization

- ▶ Need to “normalize” terms
 - ▶ Information Retrieval: indexed text & query terms must have same form.
 - ▶ We want to match **U.S.A.** and **USA**
- ▶ We implicitly define equivalence classes of terms
 - ▶ e.g., deleting periods in a term
- ▶ Alternative: asymmetric expansion:
 - ▶ Enter: **window** Search: **window, windows**
 - ▶ Enter: **windows** Search: **Windows, windows, window**
 - ▶ Enter: **Windows** Search: **Windows**
- ▶ Potentially more powerful, but less efficient



Case folding

- ▶ Applications like IR: reduce all letters to lower case
 - ▶ Since users tend to use lower case



Case folding

- ▶ Applications like IR: reduce all letters to lower case
 - ▶ Since users tend to use lower case
 - ▶ Possible exception: upper case in mid-sentence?
 - ▶ e.g., **General Motors**
 - ▶ **Fed** vs. **fed**
 - ▶ **SAIL** vs. **sail**



Case folding

- ▶ Applications like IR: reduce all letters to lower case
 - ▶ Since users tend to use lower case
 - ▶ Possible exception: upper case in mid-sentence?
 - ▶ e.g., **General Motors**
 - ▶ **Fed** vs. **fed**
 - ▶ **SAIL** vs. **sail**
- ▶ For sentiment analysis, MT, Information extraction
 - ▶ Case is helpful (**US** versus **us** is important)



Lemmatization

- ▶ Reduce inflections or variant forms to base form

Lemmatization

- ▶ Reduce inflections or variant forms to base form
 - ▶ am, are, is → be
 - ▶ car, cars, car's, cars' → car

Lemmatization

- ▶ Reduce inflections or variant forms to base form
 - ▶ am, are, is → be
 - ▶ car, cars, car's, cars' → car
- ▶ *the boy's cars are different colors* → *the boy car be different color*

Lemmatization

- ▶ Reduce inflections or variant forms to base form
 - ▶ am, are, is → be
 - ▶ car, cars, car's, cars' → car
- ▶ *the boy's cars are different colors* → *the boy car be different color*
- ▶ Lemmatization: have to find correct dictionary headword form

Lemmatization

- ▶ Reduce inflections or variant forms to base form
 - ▶ am, are, is → be
 - ▶ car, cars, car's, cars' → car
- ▶ *the boy's cars are different colors* → *the boy car be different color*
- ▶ Lemmatization: have to find correct dictionary headword form
- ▶ Machine translation
 - ▶ Spanish quiero ('I want'), quieres ('you want') same lemma as querer 'want'

Morphology

- ▶ **Morphemes:**
 - ▶ The small meaningful units that make up words

Morphology

► **Morphemes:**

- The small meaningful units that make up words
- **Stems:** The core meaning-bearing units

Morphology

► **Morphemes:**

- ▶ The small meaningful units that make up words
- ▶ **Stems:** The core meaning-bearing units
- ▶ **Affixes:** Bits and pieces that adhere to stems
 - ▶ Often with grammatical functions

Stemming

- ▶ Reduce terms to their stems in information retrieval

Stemming

- ▶ Reduce terms to their stems in information retrieval
- ▶ Stemming is crude chopping of affixes
 - ▶ language dependent
 - ▶ e.g., **automate(s)**, **automatic**, **automation** all reduced to **automat**.

Stemming

- ▶ Reduce terms to their stems in information retrieval
- ▶ Stemming is crude chopping of affixes
 - ▶ language dependent
 - ▶ e.g., **automate(s)**, **automatic**, **automation** all reduced to **automat**.

*for example compressed
and compression are both
accepted as equivalent to
compress.*

Stemming

- ▶ Reduce terms to their stems in information retrieval
- ▶ Stemming is crude chopping of affixes
 - ▶ language dependent
 - ▶ e.g., **automate(s)**, **automatic**, **automation** all reduced to **automat**.

*for example compressed
and compression are both
accepted as equivalent to
compress.*



*for exampl compress and
compress ar both accept
as equival to compress*

Porter's algorithm

The most common English stemmer

Porter's algorithm

The most common English stemmer

Step 1a

sses	→ ss	caresses	→ caress
ies	→ i	ponies	→ poni
ss	→ ss	caress	→ caress
s	→ Ø	cats	→ cat

Porter's algorithm

The most common English stemmer

Step 1a

sses	→ ss	caresses	→ caress
ies	→ i	ponies	→ poni
ss	→ ss	caress	→ caress
s	→ Ø	cats	→ cat

Step 1b

(*v*)ing	→ Ø	walking	→ walk
		sing	→ sing
(*v*)ed	→ Ø	plastered	→ plaster
...			

Porter's algorithm

The most common English stemmer

Step 1a

sses	→ ss	caresses	→ caress
ies	→ i	ponies	→ poni
ss	→ ss	caress	→ caress
s	→ Ø	cats	→ cat

Step 1b

(*v*)ing	→ Ø	walking	→ walk
		sing	→ sing
(*v*)ed	→ Ø	plastered	→ plaster
...			

Step 2 (for long stems)

ational	→ ate	relational	→ relate
izer	→ ize	digitizer	→ digitize
ator	→ ate	operator	→ operate
...			

Porter's algorithm

The most common English stemmer

Step 1a

sses	→ ss	caresses	→ caress
ies	→ i	ponies	→ poni
ss	→ ss	caress	→ caress
s	→ Ø	cats	→ cat

Step 1b

(*v*)ing	→ Ø	walking	→ walk
		sing	→ sing
(*v*)ed	→ Ø	plastered	→ plaster

...

Step 2 (for long stems)

ational	→ ate	relational	→ relate
izer	→ ize	digitizer	→ digitize
ator	→ ate	operator	→ operate
...			

Step 3 (for longer stems)

al	→ Ø	revival	→ reviv
able	→ Ø	adjustable	→ adjust
ate	→ Ø	activate	→ activ
...			

Viewing morphology in a corpus

Why only strip –ing if there is a vowel?

Viewing morphology in a corpus

Why only strip –ing if there is a vowel?

(^{*}v^{*}) ing → Ø walking → walk
sing → sing

Viewing morphology in a corpus

Why only strip –ing if there is a vowel?

Viewing morphology in a corpus

Why only strip –ing if there is a vowel?

(^{*}v^{*}) ing → Ø walking → walk
sing → sing

Viewing morphology in a corpus

Why only strip –ing if there is a vowel?

(*v*) ing → Ø walking → walk
sing → sing

1312 King
548 being
541 nothing
388 king
375 bring
358 thing
307 ring
152 something
145 coming
130 morning

Viewing morphology in a corpus

Why only strip –ing if there is a vowel?

(*v*) ing → Ø walking → walk
sing → sing

1312 King	548 being
548 being	541 nothing
541 nothing	152 something
388 king	145 coming
375 bring	130 morning
358 thing	122 having
307 ring	120 living
152 something	117 loving
145 coming	116 Being
130 morning	102 going

Word Normalization

OKAY THAT SEEMS IMPORTANT!

Sentence Segmentation

JUST LOOK FOR THE PERIOD, EASY, RIGHT?

Sentence Segmentation

Sentence Segmentation

- ▶ !, ? are relatively unambiguous

Sentence Segmentation

- ▶ !, ? are relatively unambiguous
- ▶ Period “.” is quite ambiguous

Sentence Segmentation

- ▶ !, ? are relatively unambiguous
- ▶ Period “.” is quite ambiguous
 - ▶ Sentence boundary

Sentence Segmentation

- ▶ !, ? are relatively unambiguous
- ▶ Period “.” is quite ambiguous
 - ▶ Sentence boundary
 - ▶ Abbreviations like Inc. or Dr.

Sentence Segmentation

- ▶ !, ? are relatively unambiguous
- ▶ Period “.” is quite ambiguous
 - ▶ Sentence boundary
 - ▶ Abbreviations like Inc. or Dr.
 - ▶ Numbers like .02% or 4.3

Sentence Segmentation

- ▶ !, ? are relatively unambiguous
- ▶ Period “.” is quite ambiguous
 - ▶ Sentence boundary
 - ▶ Abbreviations like Inc. or Dr.
 - ▶ Numbers like .02% or 4.3
- ▶ Build a binary classifier

Sentence Segmentation

- ▶ !, ? are relatively unambiguous
- ▶ Period “.” is quite ambiguous
 - ▶ Sentence boundary
 - ▶ Abbreviations like Inc. or Dr.
 - ▶ Numbers like .02% or 4.3
- ▶ Build a binary classifier
 - ▶ Looks at a “.”

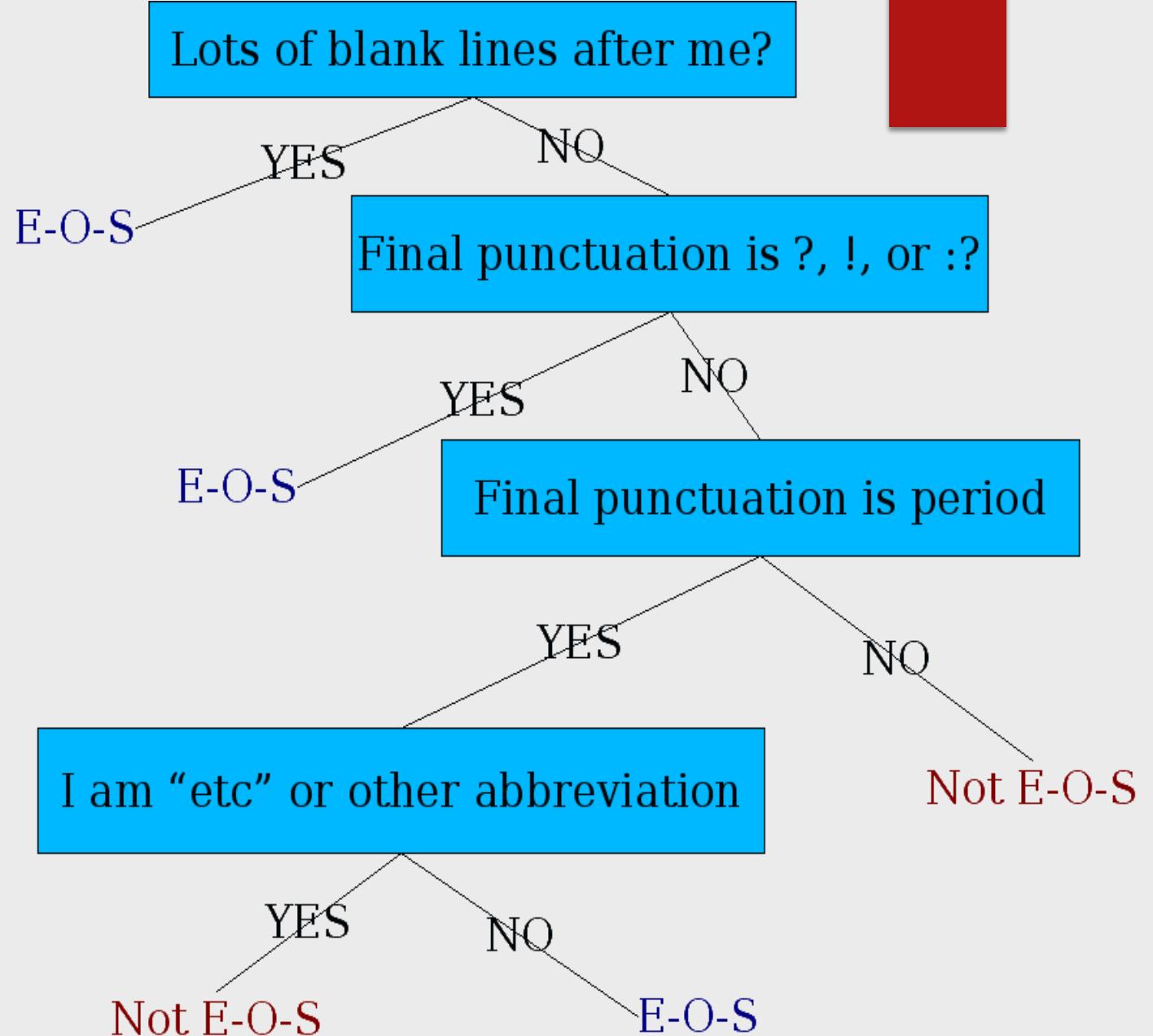
Sentence Segmentation

- ▶ !, ? are relatively unambiguous
- ▶ Period “.” is quite ambiguous
 - ▶ Sentence boundary
 - ▶ Abbreviations like Inc. or Dr.
 - ▶ Numbers like .02% or 4.3
- ▶ Build a binary classifier
 - ▶ Looks at a “.”
 - ▶ Decides EndOfSentence/NotEndOfSentence

Sentence Segmentation

- ▶ !, ? are relatively unambiguous
- ▶ Period “.” is quite ambiguous
 - ▶ Sentence boundary
 - ▶ Abbreviations like Inc. or Dr.
 - ▶ Numbers like .02% or 4.3
- ▶ Build a binary classifier
 - ▶ Looks at a “.”
 - ▶ Decides EndOfSentence/NotEndOfSentence
 - ▶ Classifiers: hand-written rules, regular expressions, or machine-learning

Decision Tree Approach



More sophisticated decision tree features

More sophisticated decision tree features

- ▶ Case of word with “.”: Upper, Lower, Cap, Number

More sophisticated decision tree features

- ▶ Case of word with “.”: Upper, Lower, Cap, Number
- ▶ Case of word after “.”: Upper, Lower, Cap, Number

More sophisticated decision tree features

- ▶ Case of word with “.”: Upper, Lower, Cap, Number
- ▶ Case of word after “.”: Upper, Lower, Cap, Number

- ▶ Numeric features
 - ▶ Length of word with “.”
 - ▶ Probability(word with “.” occurs at end-of-s)
 - ▶ Probability(word after “.” occurs at beginning-of-s)

Implementing Decision Trees

- ▶ A decision tree is just an if-then-else statement

Implementing Decision Trees

- ▶ A decision tree is just an if-then-else statement
- ▶ The interesting research is choosing the features

Implementing Decision Trees

- ▶ A decision tree is just an if-then-else statement
- ▶ The interesting research is choosing the features
- ▶ Setting up the structure is often too hard to do by hand

Implementing Decision Trees

- ▶ A decision tree is just an if-then-else statement
- ▶ The interesting research is choosing the features
- ▶ Setting up the structure is often too hard to do by hand
 - ▶ Hand-building only possible for very simple features, domains

Implementing Decision Trees

- ▶ A decision tree is just an if-then-else statement
- ▶ The interesting research is choosing the features
- ▶ Setting up the structure is often too hard to do by hand
 - ▶ Hand-building only possible for very simple features, domains
 - ▶ For numeric features, it's too hard to pick each threshold

Implementing Decision Trees

- ▶ A decision tree is just an if-then-else statement
- ▶ The interesting research is choosing the features
- ▶ Setting up the structure is often too hard to do by hand
 - ▶ Hand-building only possible for very simple features, domains
 - ▶ For numeric features, it's too hard to pick each threshold
 - ▶ Instead, structure usually learned by machine learning from a training corpus

Decision Trees and other classifiers

- ▶ We can think of the questions in a decision tree
- ▶ As features that could be exploited by any kind of classifier

Decision Trees and other classifiers

- ▶ We can think of the questions in a decision tree
- ▶ As features that could be exploited by any kind of classifier
 - ▶ Logistic regression
 - ▶ SVM
 - ▶ Neural Nets
 - ▶ etc.

Sentence Segmentation

NEVER THOUGHT EVEN SEPARATING SENTENCES WOULD
BE SO COMPLEX..

BASIC TEXT PROCESSING

NOT SO BASIC, WAS IT?

MINIMUM EDIT DISTANCE

WHAT DISTANCE?

Defining Minimum Edit Distance

WHY DO WE HAVE TO DEFINE EVERYTHING?

How similar are two strings?

How similar are two strings?

- ▶ Spell correction
 - ▶ The user typed
“graffe”

How similar are two strings?

- ▶ Spell correction
 - ▶ The user typed
“graffe”
 - ▶ Which is closest?
 - ▶ graf
 - ▶ graft
 - ▶ grail
 - ▶ giraffe

How similar are two strings?

- ▶ Spell correction
 - ▶ The user typed “graffe”
Which is closest?
 - ▶ graf
 - ▶ graft
 - ▶ grail
 - ▶ giraffe
- Computational Biology
 - Align two sequences of nucleotides

AGGCTATCACCTGACCTCCAGGCCGATGCC
TAGCTATCACGACCGCGGTGATTGCCGAC

How similar are two strings?

- ▶ Spell correction
 - ▶ The user typed “graffe”
Which is closest?
 - ▶ graf
 - ▶ graft
 - ▶ grail
 - ▶ giraffe
- Computational Biology
 - Align two sequences of nucleotides

AGGCTATCACCTGACCTCCAGGCCGATGCC
TAGCTATCACGACC CGCGGTGATTGCCCGAC

 - Resulting alignment:

— **AGGCTATCACCTGACCTCCAGGCCGATGCC** ---
TAG-CTATCAC--GACC GC--GGTCGA TT**TGCCCGAC**

How similar are two strings?

- ▶ Spell correction
 - ▶ The user typed “graffe”
Which is closest?
 - ▶ graf
 - ▶ graft
 - ▶ grail
 - ▶ giraffe
- Computational Biology
 - Align two sequences of nucleotides

AGGCTATCACCTGACCTCCAGGCCGATGCC
TAGCTATCACGACC CGCGGTGATTGCCCGAC

 - Resulting alignment:

— **AGGCTATCACCTGACCTCCAGGCCGATGCC** — TGCCC —
TAG — CTATCAC — GACC GC — GGTCGA TT **TGCCCGAC**
- Also for Machine Translation, Information Extraction, Speech Recognition

Edit Distance



The minimum edit distance between two strings



Is the minimum number of editing operations

Insertion
Deletion
Substitution



Needed to transform one into the other

Minimum 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

Minimum 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				

Minimum 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

- ▶ If each operation has cost of 1

Minimum 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				

- ▶ If each operation has cost of 1
- ▶ Distance between these is 5

Minimum 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				

- ▶ If each operation has cost of 1
 - ▶ Distance between these is 5
- ▶ If substitutions cost 2 (Levenshtein)

Minimum 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				

- ▶ If each operation has cost of 1
 - ▶ Distance between these is 5
- ▶ If substitutions cost 2 (Levenshtein)
 - ▶ Distance between them is 8

Alignment in Computational Biology

- Given a sequence of bases

AGGCTATCACCTGACCTCCAGGCCGATGCC
TAGCTATCACGACCGCGGGTCGATTGCCCGAC

Alignment in Computational Biology

- Given a sequence of bases

AGGCTATCACCTGACCTCCAGGCCGATGCC
TAGCTATCACGACCGCGGGTCGATTGCCCGAC

- An alignment:

Alignment in Computational Biology

- Given a sequence of bases

AGGCTATCACCTGACCTCCAGGCCGATGCC

TAGCTATCACGACC CGCGGT CGATTGCCCGAC

- An alignment:

-AGGCTATCACCTGACCTCCAGGCCGATGCC---

TAG-CTATCAC--GACC GC--GGT CGA TTGCCCGAC

Alignment in Computational Biology

- Given a sequence of bases

AGGCTATCACCTGACCTCCAGGCCGATGCC

TAGCTATCACGACC CGGGT CGATTGCCCGAC

- An alignment:

-AGGCTATCACCTGACCTCCAGGCCGATGCC---

TAG-CTATCAC--GACC GC--GGT CGATTGCCCGAC

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

Other uses of Edit Distance in NLP

- ▶ 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

Other uses of Edit Distance in NLP

- ▶ 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

- ▶ Named Entity Extraction and Entity Coreference

- ▶ IBM Inc. announced today

- ▶ IBM profits

- ▶ Stanford President John Hennessy announced yesterday

- ▶ for Stanford University President John Hennessy

How to find the Min Edit Distance?

- ▶ Searching for a path (sequence of edits) from the start string to the final string:

How to find the Min Edit Distance?

- ▶ Searching for a path (sequence of edits) from the start string to the final string:
 - ▶ **Initial state:** the word we're transforming

How to find the Min Edit Distance?

- ▶ Searching for a path (sequence of edits) from the start string to the final string:
 - ▶ **Initial state:** the word we're transforming
 - ▶ **Operators:** insert, delete, substitute

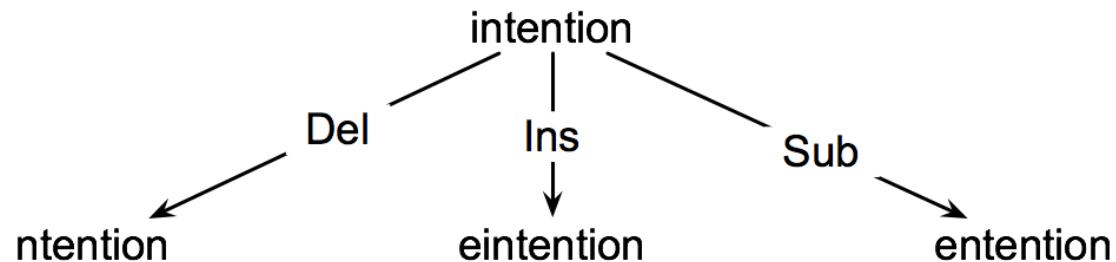
How to find the Min Edit Distance?

- ▶ Searching for a path (sequence of edits) from the start string to the final string:
 - ▶ **Initial state:** the word we're transforming
 - ▶ **Operators:** insert, delete, substitute
 - ▶ **Goal state:** the word we're trying to get to

How to find the Min Edit Distance?

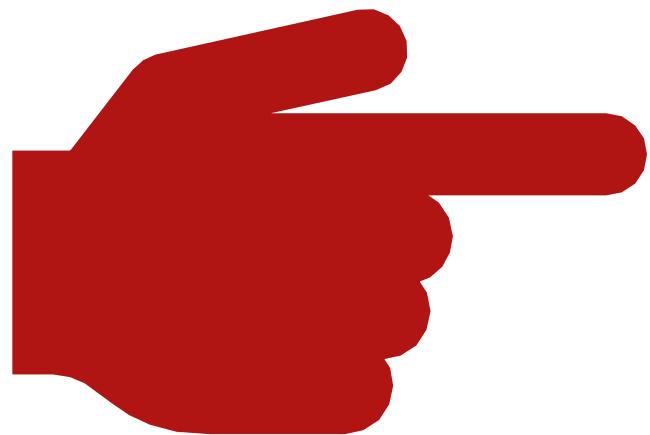
- ▶ Searching for a path (sequence of edits) from the start string to the final string:
 - ▶ **Initial state:** the word we're transforming
 - ▶ **Operators:** insert, delete, substitute
 - ▶ **Goal state:** the word we're trying to get to
 - ▶ **Path cost:** what we want to minimize: the number of edits

How to find the Min Edit Distance?



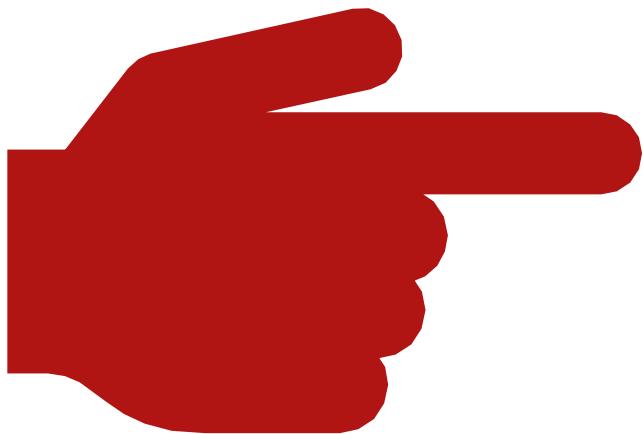
- ▶ Searching for a path (sequence of edits) from the start string to the final string:
- ▶ **Initial state:** the word we're transforming
- ▶ **Operators:** insert, delete, substitute
- ▶ **Goal state:** the word we're trying to get to
- ▶ **Path cost:** what we want to minimize: the number of edits

Minimum Edit as Search



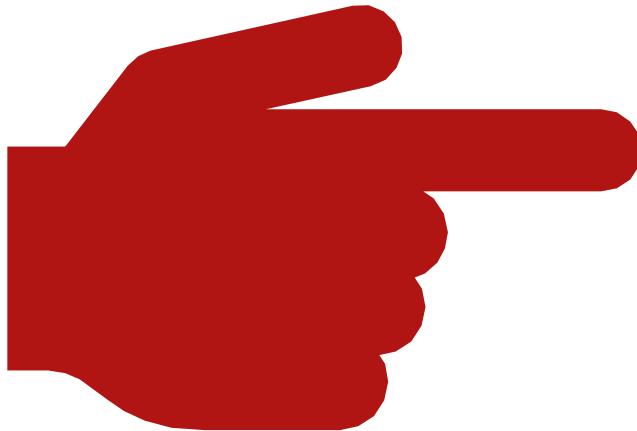
Minimum Edit as Search

- ▶ But the space of all edit sequences is huge!

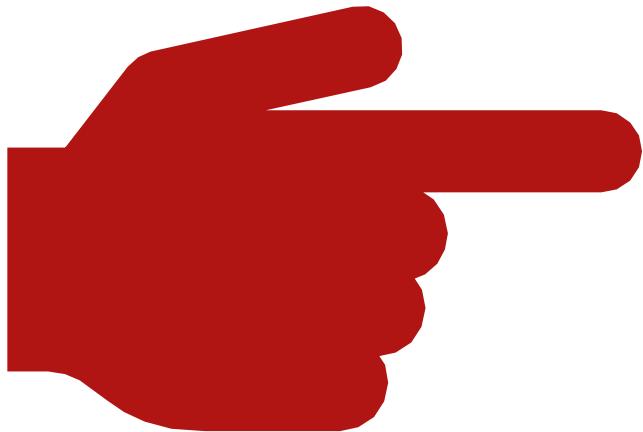


Minimum Edit as Search

- ▶ But the space of all edit sequences is huge!
 - ▶ We can't afford to navigate naïvely

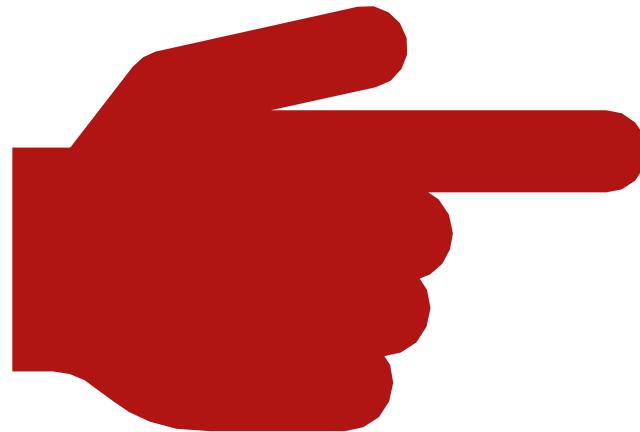


Minimum Edit as Search



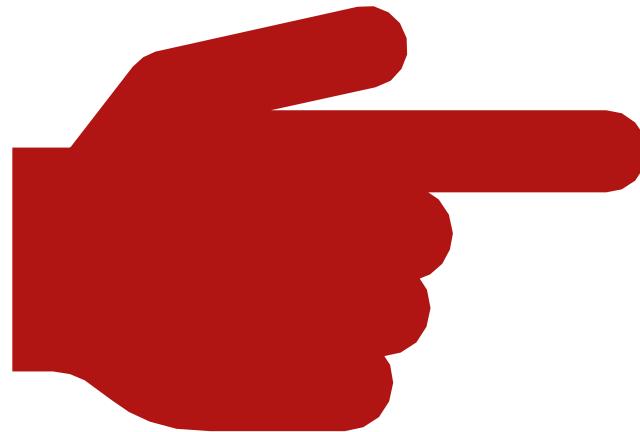
- ▶ But the space of all edit sequences is huge!
 - ▶ We can't afford to navigate naïvely
 - ▶ Lots of distinct paths wind up at the same state.

Minimum Edit as Search



- ▶ But the space of all edit sequences is huge!
 - ▶ We can't afford to navigate naïvely
 - ▶ Lots of distinct paths wind up at the same state.
 - ▶ We don't have to keep track of all of them

Minimum Edit as Search



- ▶ But the space of all edit sequences is huge!
 - ▶ We can't afford to navigate naïvely
 - ▶ Lots of distinct paths wind up at the same state.
 - ▶ We don't have to keep track of all of them
 - ▶ Just the shortest path to each of those revisited states.

Defining Min Edit Distance

- ▶ For two strings
 - ▶ X of length n
 - ▶ Y of length m

Defining Min Edit Distance

- ▶ For two strings
 - ▶ X of length n
 - ▶ Y of length m
- ▶ We define $D(i,j)$

Defining Min Edit Distance

- ▶ For two strings
 - ▶ 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 $D(n,m)$

Defining Minimum Edit Distance

DIDN'T WE SPEND TOO MUCH TIME DEFINING?

Computing Minimum Edit Distance

YES, I HAVE MY CALCULATOR OUT, LET'S COMPUTE!

Dynamic Programming for Minimum Edit Distance

Dynamic Programming for Minimum Edit Distance

- ▶ **Dynamic programming:** A tabular computation of $D(n,m)$

Dynamic Programming for Minimum Edit Distance

- ▶ **Dynamic programming:** A tabular computation of $D(n,m)$
- ▶ Solving problems by combining solutions to subproblems.

Dynamic Programming for Minimum Edit Distance

- ▶ **Dynamic programming:** A tabular computation of $D(n,m)$
- ▶ Solving problems by combining solutions to subproblems.
- ▶ Bottom-up

Dynamic Programming for Minimum Edit Distance

- ▶ **Dynamic programming:** A tabular computation of $D(n,m)$
- ▶ Solving problems by combining solutions to subproblems.
- ▶ Bottom-up
 - ▶ We compute $D(i,j)$ for small i,j

Dynamic Programming for Minimum Edit Distance

- ▶ **Dynamic programming:** A tabular computation of $D(n,m)$
- ▶ Solving problems by combining solutions to subproblems.
- ▶ Bottom-up
 - ▶ We compute $D(i,j)$ for small i,j
 - ▶ And compute larger $D(i,j)$ based on previously computed smaller values

Dynamic Programming for Minimum Edit Distance

- ▶ **Dynamic programming:** A tabular computation of $D(n,m)$
- ▶ Solving problems by combining solutions to subproblems.
- ▶ 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 ($0 < j < m$)

Defining Min Edit Distance (Levenshtein)

Defining Min Edit Distance (Levenshtein)

- ▶ Initialization

$$D(i, 0) = i$$

$$D(0, j) = j$$

Defining Min Edit Distance (Levenshtein)

▶ Initialization

$$D(i, 0) = i$$

$$D(0, j) = j$$

▶ Recurrence Relation:

For each $i = 1 \dots M$

For each $j = 1 \dots N$

$$D(i-1, j) + 1$$

$$D(i, j) = \min D(i, j-1) + 1$$

$$D(i-1, j-1) + 2; \text{ if } X(i) \neq Y(j)$$

$$0; \text{ if } X(i) = Y(j)$$

Defining Min Edit Distance (Levenshtein)

▶ Initialization

$$D(i, 0) = i$$

$$D(0, j) = j$$

▶ Recurrence Relation:

For each $i = 1 \dots M$

For each $j = 1 \dots N$

$$D(i, j) = \min$$

$$D(i, j) = \min \left\{ \begin{array}{l} D(i-1, j) + 1 \\ D(i, j-1) + 1 \\ D(i-1, j-1) + \begin{cases} 2; & \text{if } X(i) \neq Y(j) \\ 0; & \text{if } X(i) = Y(j) \end{cases} \end{array} \right.$$

▶ 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	$D(i,j) = \min \left\{ \begin{array}{l} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{array} \right\}$								
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

$$D(i,j) = \min \left\{ \begin{array}{l} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{array} \right\}$$



Edit Distance

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

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

The Edit Distance Table

N	9	8	9	10	11	12	11	10	9	8
O	8	7	8	9	10	11	10	9	8	9
I	7	6	7	8	9	10	9	8	9	10
T	6	5	6	7	8	9	8	9	10	11
N	5	4	5	6	7	8	9	10	11	10
E	4	3	4	5	6	7	8	9	10	9
T	3	4	5	6	7	8	7	8	9	8
N	2	3	4	5	6	7	8	7	8	7
I	1	2	3	4	5	6	7	6	7	8
#	0	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

Computing Minimum Edit Distance

THAT'S A GOOD DEAL OF CODING!

Backtrace for Computing Alignments

BACKTRACING?

Computing alignments

Computing alignments

- ▶ Edit distance isn't sufficient

Computing alignments

- ▶ Edit distance isn't sufficient
 - ▶ We often need to **align** each character of the two strings to each other

Computing alignments

- ▶ Edit distance isn't sufficient
 - ▶ We often need to **align** each character of the two strings to each other
- ▶ We do this by keeping a “backtrace”

Computing alignments

- ▶ Edit distance isn't sufficient
 - ▶ We often need to **align** each character of the two strings to each other
- ▶ We do this by keeping a “backtrace”
- ▶ Every time we enter a cell, remember where we came from

Computing alignments

- ▶ Edit distance isn't sufficient
 - ▶ We often need to **align** each character of the two strings to each other
- ▶ We do this by keeping a “backtrace”
- ▶ Every time we enter a cell, remember where we came from
- ▶ When we reach the end,

Computing alignments

- ▶ Edit distance isn't sufficient
 - ▶ We often need to **align** each character of the two strings to each other
- ▶ We do this by keeping a “backtrace”
- ▶ Every time we enter a cell, remember where we came from
- ▶ When we reach the end,
 - ▶ Trace back the path from the upper right corner to read off the alignment

Edit Distance

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

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

MinEdit with Backtrace

n	9	↓8	/←↓9	/←↓10	/←↓11	/←↓12	↓11	↓10	↓9	/8	
o	8	↓7	/←↓8	/←↓9	/←↓10	/←↓11	↓10	↓9	/8	←9	
i	7	↓6	/←↓7	/←↓8	/←↓9	/←↓10	↓9	/8	←9	←10	
t	6	↓5	/←↓6	/←↓7	/←↓8	/←↓9	/8	←9	←10	↓11	
n	5	↓4	/←↓5	/←↓6	/←↓7	/←↓8	/←↓9	/←↓10	/←↓11	/10	
e	4	/3	←4	/←5	←6	←7	←8	/←↓9	/←↓10	↓9	
t	3	/←↓4	/←↓5	/←↓6	/←↓7	/←↓8	/7	←8	/←↓9	↓8	
n	2	/←↓3	/←↓4	/←↓5	/←↓6	/←↓7	/←↓8	↓7	/←↓8	/7	
i	1	/←↓2	/←↓3	/←↓4	/←↓5	/←↓6	/←↓7	/6	←7	←8	
#	0	1	2	3	4	5	6	7	8	9	
#		e	x	e	c	u	t	i	o	n	

Adding Backtrace to Minimum Edit Distance

► Base conditions:

$$D(i, 0) = i$$

$$D(0, j) = j$$

Termination:

$D(N, M)$ is distance

► Recurrence Relation:

For each $i = 1 \dots M$

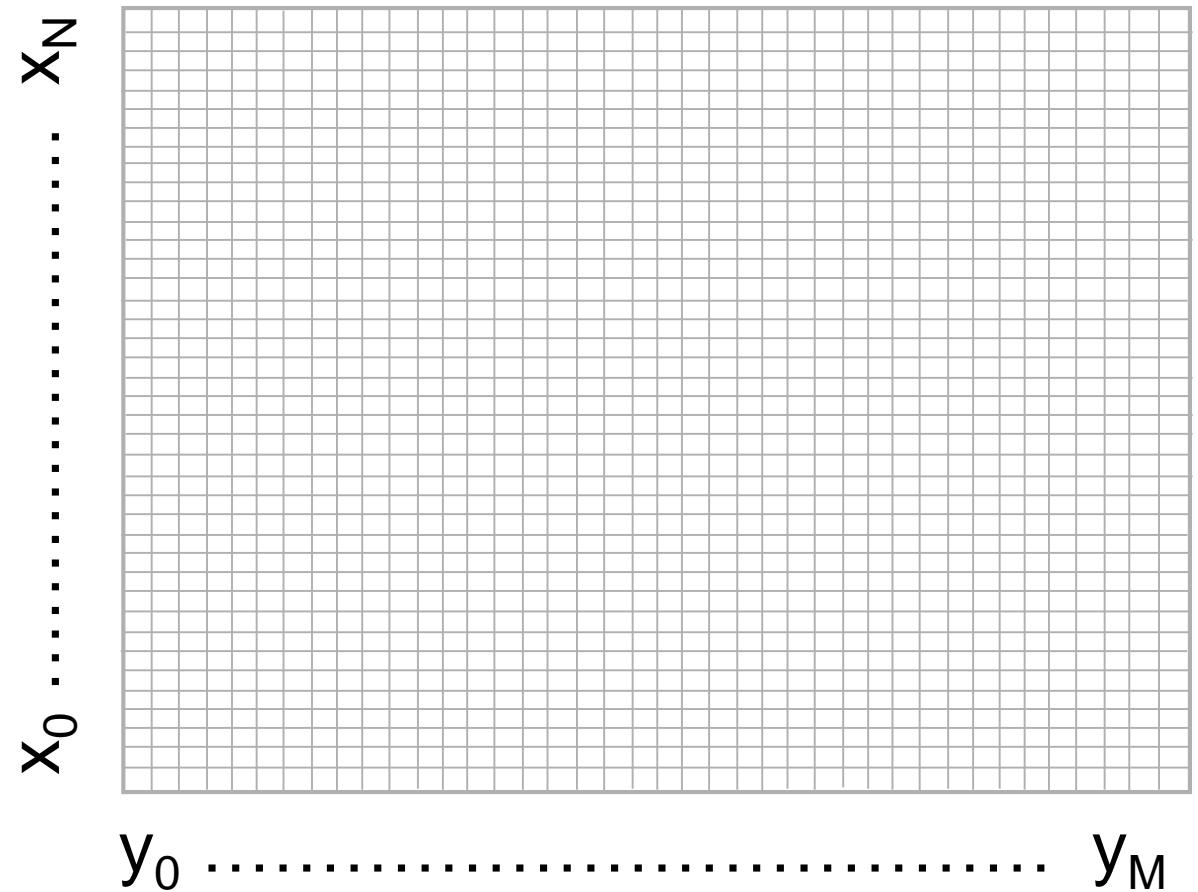
For each $j = 1 \dots N$

$$D(i, j) = \min \left\{ \begin{array}{l} D(i-1, j) + 1 \\ D(i, j-1) + 1 \\ D(i-1, j-1) + \end{array} \right. \begin{array}{l} \text{deletion} \\ \text{insertion} \\ \left\{ \begin{array}{l} 2; \text{ if } X(i) \neq Y(j) \\ 0; \text{ if } X(i) = Y(j) \end{array} \right. \end{array}$$

$\text{ptr}(i, j) = \left\{ \begin{array}{l} \text{LEFT} \\ \text{DOWN} \\ \text{DIAG} \end{array} \right. \begin{array}{l} \text{insertion} \\ \text{deletion} \\ \text{substitution} \end{array}$

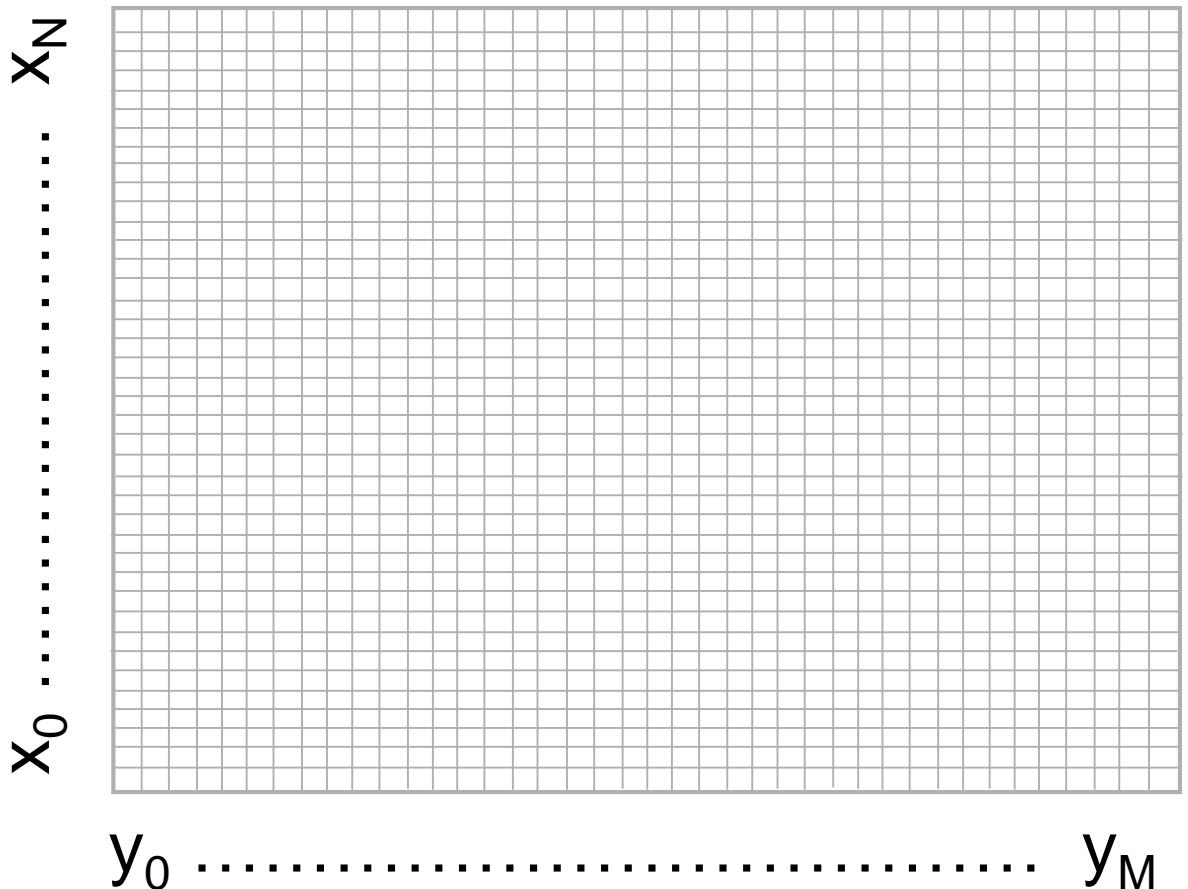
substitution

The Distance Matrix



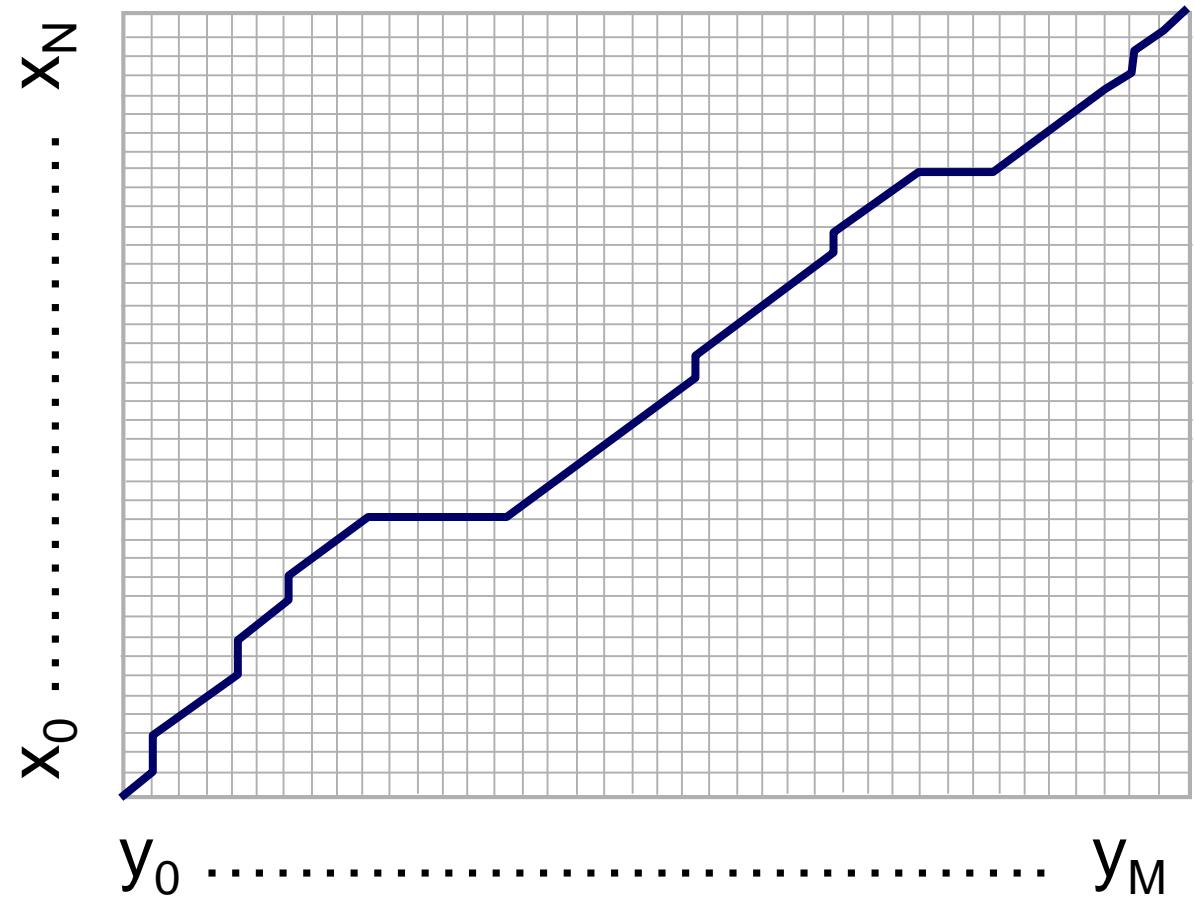
The Distance Matrix

- ▶ Every non-decreasing path from $(0,0)$ to (M, N)
- ▶ corresponds to an alignment of the two sequences



The Distance Matrix

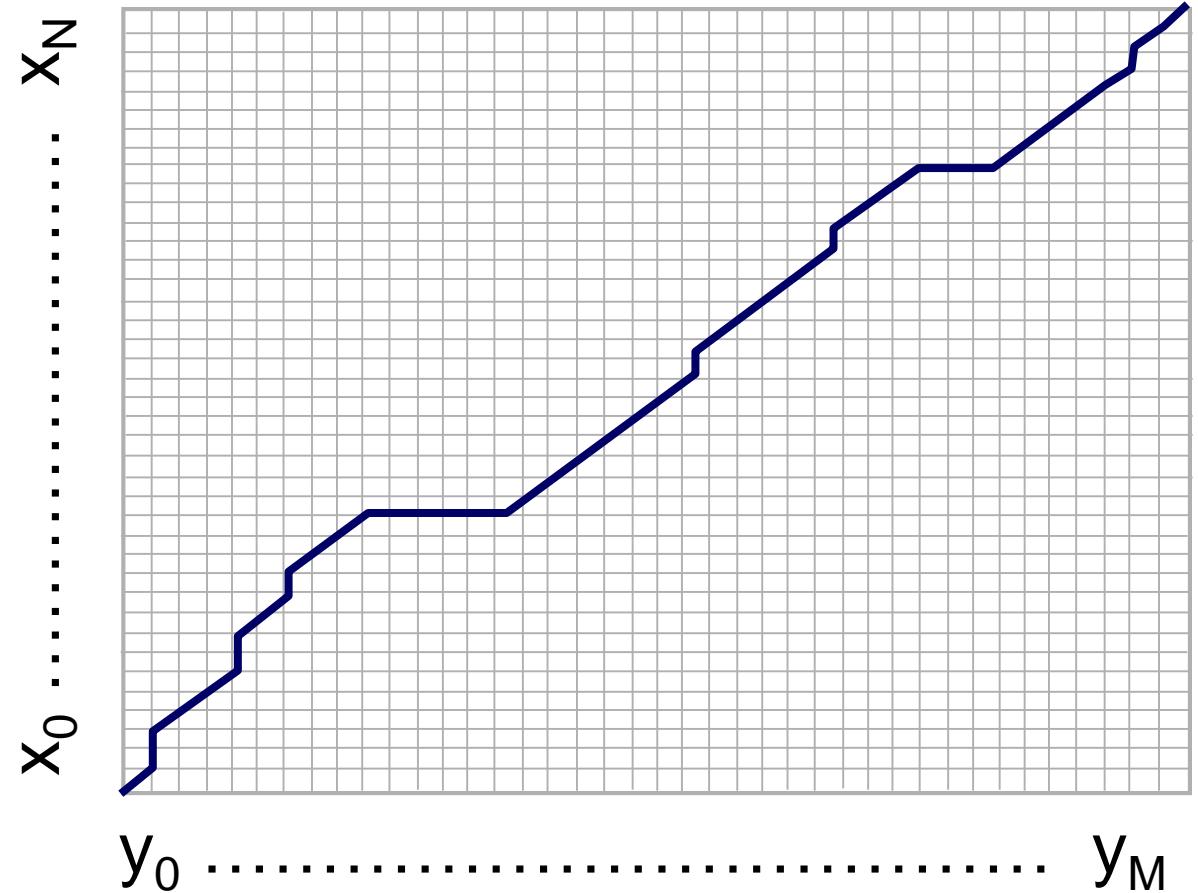
- ▶ Every non-decreasing path from $(0,0)$ to (M, N)
- ▶ corresponds to an alignment of the two sequences



The Distance Matrix

- ▶ Every non-decreasing path from $(0,0)$ to (M, N)
- ▶ corresponds to an alignment of the two sequences

An optimal alignment is composed of optimal subalignments



Result of Backtrace

- ▶ Two strings and their **alignment**:

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

Performance

- ▶ Time:

Performance

- ▶ Time:
 $O(nm)$

Performance

- ▶ Time:
 $O(nm)$
- ▶ Space:
 $O(nm)$

Performance

- ▶ Time:
 $O(nm)$
- ▶ Space:
 $O(nm)$
- ▶ Backtrace
 $O(n+m)$

Backtrace for Computing Alignments

ACE-ING BACKTRACING! WOW I CAN RHYME!

Weighted Minimum Edit Distance

WHY WOULD WE WEIGHT WHEN IT'S ALREADY MINIMUM?

Weighted Edit Distance

Weighted Edit Distance

- ▶ Why would we add weights to the computation?
 - ▶ Spell Correction: some letters are more likely to be mistyped than others
 - ▶ Biology: certain kinds of deletions or insertions are more likely than others

Confusion
matrix for
spelling
errors

sub[X, Y] = Substitution of X (incorrect) for Y (correct)

X	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
	Y (correct)																									
a	0	0	7	1	342	0	0	2	118	0	1	0	0	3	76	0	0	1	35	9	9	0	1	0	5	0
b	0	0	9	9	2	2	3	1	0	0	0	5	11	5	0	10	0	0	2	1	0	0	8	0	0	0
c	6	5	0	16	0	9	5	0	0	0	1	0	7	9	1	10	2	5	39	40	1	3	7	1	1	0
d	1	10	13	0	12	0	5	5	0	0	2	3	7	3	0	1	0	43	30	22	0	0	4	0	2	0
e	388	0	3	11	0	2	2	0	89	0	0	3	0	5	93	0	0	14	12	6	15	0	1	0	18	0
f	0	15	0	3	1	0	5	2	0	0	0	3	4	1	0	0	0	6	4	12	0	0	2	0	0	0
g	4	1	11	11	9	2	0	0	0	1	1	3	0	0	2	1	3	5	13	21	0	0	1	0	3	0
h	1	8	0	3	0	0	0	0	0	0	2	0	12	14	2	3	0	3	1	11	0	0	2	0	0	0
i	103	0	0	0	146	0	1	0	0	0	0	6	0	0	49	0	0	0	2	1	47	0	2	1	15	0
j	0	1	1	9	0	0	1	0	0	0	0	2	1	0	0	0	0	0	5	0	0	0	0	0	0	0
k	1	2	8	4	1	1	2	5	0	0	0	5	0	2	0	0	0	6	0	0	0	4	0	0	0	3
l	2	10	1	4	0	4	5	6	13	0	1	0	0	14	2	5	0	11	10	2	0	0	0	0	0	0
m	1	3	7	8	0	2	0	6	0	0	4	4	0	180	0	6	0	0	9	15	13	3	2	2	3	0
n	2	7	6	5	3	0	1	19	1	0	4	35	78	0	0	7	0	28	5	7	0	0	1	2	0	2
o	91	1	1	3	116	0	0	0	25	0	2	0	0	0	0	14	0	2	4	14	39	0	0	0	18	0
p	0	11	1	2	0	6	5	0	2	9	0	2	7	6	15	0	0	1	3	6	0	4	1	0	0	0
q	0	0	1	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
r	0	14	0	30	12	2	2	8	2	0	5	8	4	20	1	14	0	0	12	22	4	0	0	1	0	0
s	11	8	27	33	35	4	0	1	0	1	0	27	0	6	1	7	0	14	0	15	0	0	5	3	20	1
t	3	4	9	42	7	5	19	5	0	1	0	14	9	5	5	6	0	11	37	0	0	2	19	0	7	6
u	20	0	0	0	44	0	0	0	64	0	0	0	0	2	43	0	0	4	0	0	0	0	2	0	8	0
v	0	0	7	0	0	3	0	0	0	0	0	1	0	0	1	0	0	0	8	3	0	0	0	0	0	0
w	2	2	1	0	1	0	0	2	0	0	1	0	0	0	0	7	0	6	3	3	1	0	0	0	0	0
x	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0
y	0	0	2	0	15	0	1	7	15	0	0	0	2	0	6	1	0	7	36	8	5	0	0	1	0	0
z	0	0	0	7	0	0	0	0	0	0	0	7	5	0	0	0	0	2	21	3	0	0	0	0	3	0

Confusion matrix for spelling errors



Weighted Min Edit Distance

Weighted Min Edit Distance

- ▶ Initialization:

$$D(0, 0) = 0$$

$$D(i, 0) = D(i-1, 0) + \text{del}[x(i)]; \quad 1 < i \leq N$$

$$D(0, j) = D(0, j-1) + \text{ins}[y(j)]; \quad 1 < j \leq M$$

Weighted Min Edit Distance

► Initialization:

$$D(0, 0) = 0$$

$$D(i, 0) = D(i-1, 0) + \text{del}[x(i)]; \quad 1 < i \leq N$$

$$D(0, j) = D(0, j-1) + \text{ins}[y(j)]; \quad 1 < j \leq M$$

► Recurrence Relation:

$$D(i-1, j) + \text{del}[x(i)]$$

$$D(i, j) = \min D(i, j-1) + \text{ins}[y(j)]$$

$$D(i-1, j-1) + \text{sub}[x(i), y(j)]$$

Weighted Min Edit Distance

► Initialization:

$$D(0, 0) = 0$$

$$D(i, 0) = D(i-1, 0) + \text{del}[x(i)]; \quad 1 < i \leq N$$

$$D(0, j) = D(0, j-1) + \text{ins}[y(j)]; \quad 1 < j \leq M$$

► Recurrence Relation:

$$D(i, j) = \min \begin{cases} D(i-1, j) + \text{del}[x(i)] \\ D(i, j-1) + \text{ins}[y(j)] \\ D(i-1, j-1) + \text{sub}[x(i), y(j)] \end{cases}$$

► Termination:

$D(N, M)$ is distance

Where did the name, dynamic programming, come from?

...The 1950s were not good years for mathematical research. [the] Secretary of Defense ...had a pathological fear and hatred of the word, research...

I decided therefore to use the word, “**programming**”.

I wanted to get across the idea that this was dynamic, this was multistage... I thought, let's ... take a word that has an absolutely precise meaning, namely **dynamic**... it's impossible to use the word, **dynamic**, in a pejorative sense. Try thinking of some combination that will possibly give it a pejorative meaning. It's impossible.

Thus, I thought dynamic programming was a good name. It was something not even a Congressman could object to.”

Richard Bellman, “Eye of the Hurricane: an autobiography” 1984.

Weighted Minimum Edit Distance

WE END UP WEIGHTING EVERYTHING, I GUESS.

Minimum Edit Distance

AWESOME, I AM READY TO START!



END OF SESSION 1

PHEW, FINALLY!!