Lec02: Text Segmentation and Normalization

Build Content | Assessments | Tools | Partner Content | Discover Content

Assign 01

Enabled: Statistics Tracking

Attached Files: Assignment_01.pdf (195.82 KB)

COMP8730-1-R-2022W (Natural Language Processing & Understanding)

- Homepage
- Weekly Schedule
- Announcements
- Learning Outcome
- Syllabus (Outline)
- Lectures
- Books
- Classroom (Online)
- Exams
- Research Project
- My Grades
- Discussion Board
- Office
- Email

Forum: Research Project

Forums are made up of individual discussion threads that can be organized around a particular subject. A thread is a conversation within a forum that includes the i[...] all replies to it. When you access a forum, a list of threads appears. More Help

Create Thread | Unsubscribe | Search | Discover Content | Display

Thread Actions | Collect | Delete

DATE	THREAD	AUTHOR	STATUS	TAGS	UNREAD POSTS	UNREAD REPLIES TO ME	TOT[...] POS[...]
1/21/22 1:23 PM	Reproducibility Projects	Hossein Fani	Published		0	0	1
1/18/22 12:31 PM	Free bootcamps on Python, NLP, ...	Hossein Fani	Published		0	0	1
1/18/22 7:35 AM	List of Top Rank NLP-IR Conferences	Hossein Fani	Published		0	0	1

Thread Actions | Collect | Delete

Displaying 1 to 3 of 3 items

The Turing Te

by Alan Turing in 19

Player C, the interrog
the task of trying to de
player – A or B – is a
which is a human. The
limited to using the
written questions to
determination.

VOL. LIX. NO. 236.] [October, 1950

MIND

A QUARTERLY REVIEW

OF

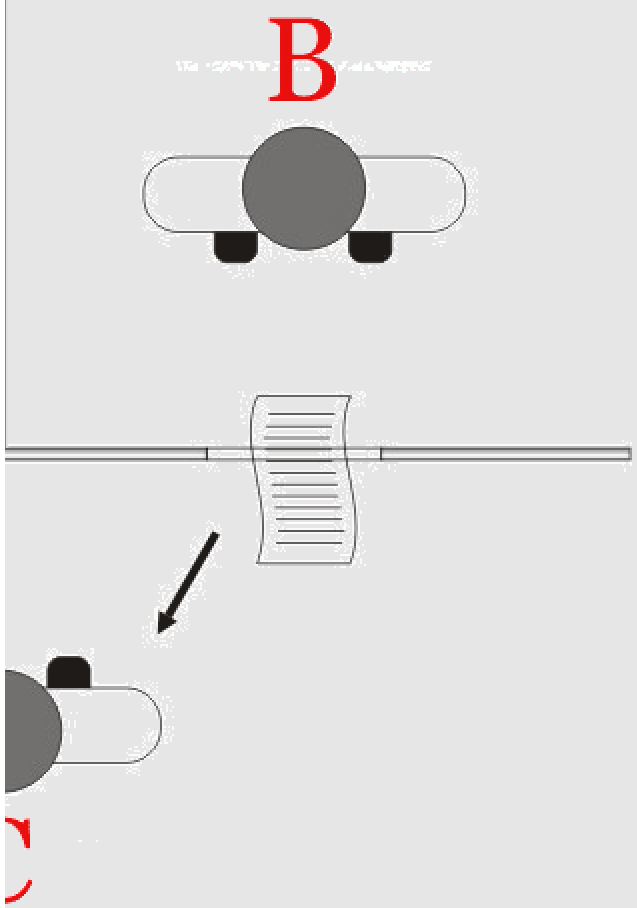
PSYCHOLOGY AND PHILOSOPHY

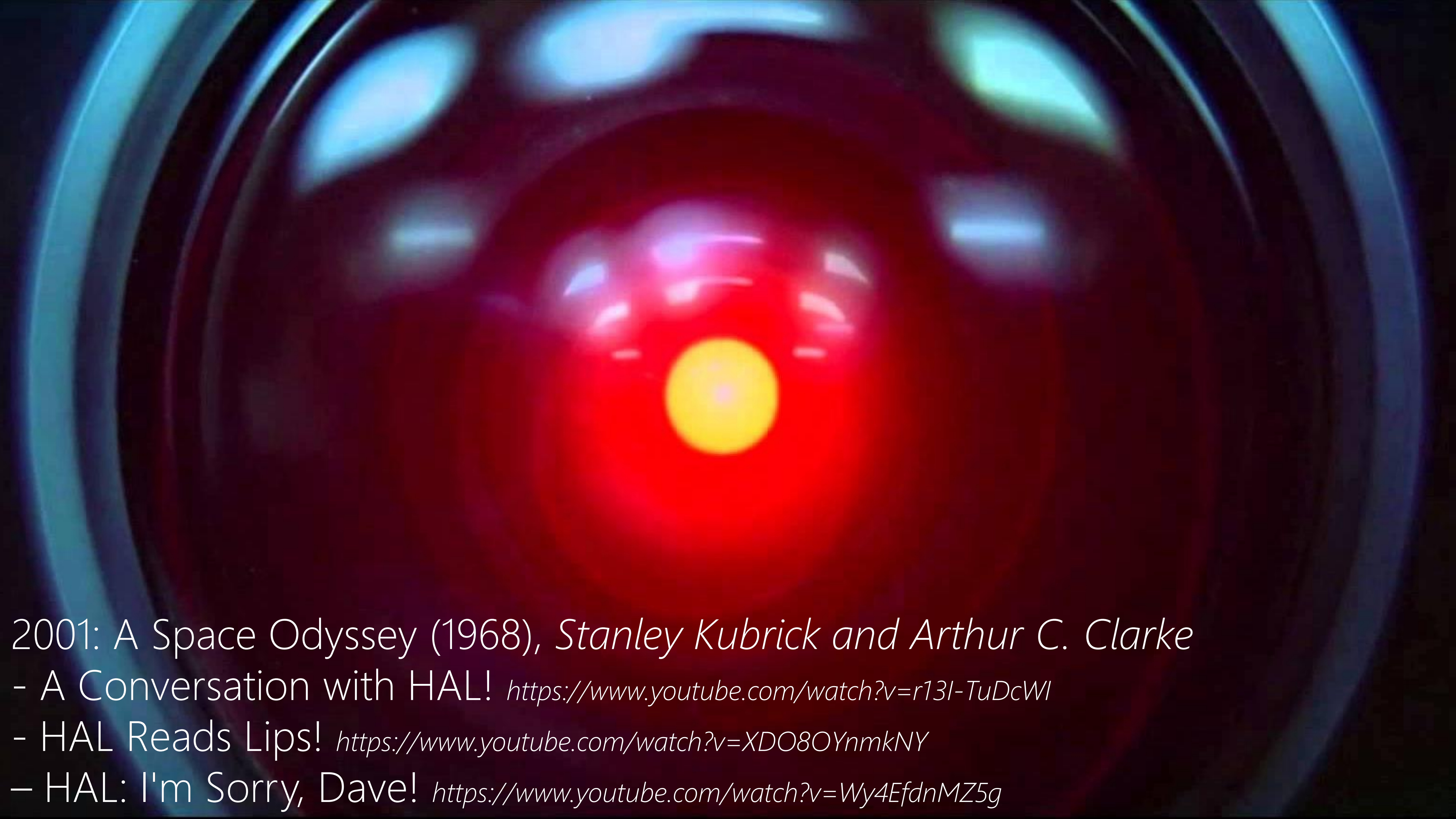
I.—COMPUTING MACHINERY AND INTELLIGENCE

BY A. M. TURING

1. *The Imitation Game.*

I PROPOSE to consider the question, 'Can machines think?' This should begin with definitions of the meaning of the terms 'machine' and 'think'. The definitions might be framed so as to reflect so far as possible the normal use of the words, but this attitude is dangerous. If the meaning of the words 'machine' and 'think' are to be found by examining how they are commonly used it is difficult to escape the conclusion that the meaning and the answer to the question, 'Can machines think?' is to be sought in a statistical survey such as a Gallup poll. But this is absurd. Instead of attempting such a definition I shall replace the question by another, which is closely related to it and is expressed in relatively unambiguous words.



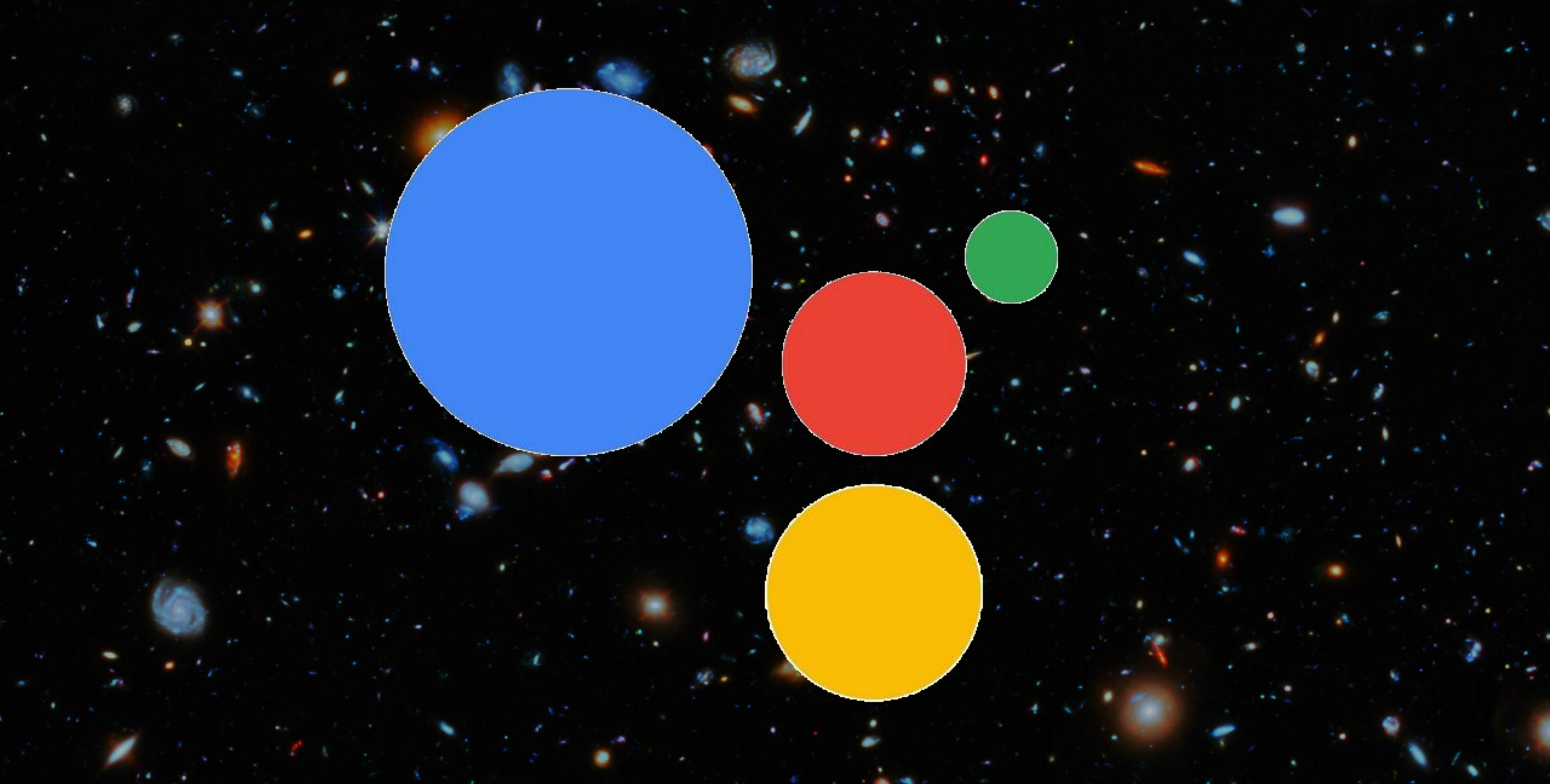


2001: A Space Odyssey (1968), *Stanley Kubrick and Arthur C. Clarke*

- A Conversation with HAL! *https://www.youtube.com/watch?v=r13I-TuDcWI*

- HAL Reads Lips! *https://www.youtube.com/watch?v=XDO8OYnmkNY*

– HAL: I'm Sorry, Dave! *https://www.youtube.com/watch?v=Wy4EfdnMZ5g*



Google Duplex: A.I. Assistant Calls Local Businesses To Make Appointments

https://www.youtube.com/watch?v=D5VN56jQMWM

Research Priorities for Artificial Intelligence

The capacity for language is one of the central features of human intelligence and is therefore a prerequisite for artificial intelligence.

Despite its many practical applications, language is perhaps number 300 in the priority list for AI research. It would be a great achievement if AI could attain the capabilities of an orangutan, which do not include language!
- Yann LeCun (computer vision researcher)

Russell, Stuart, Daniel Dewey, and Max Tegmark. "Research priorities for robust and beneficial artificial intelligence." *Ai Magazine* 36.4 (2015): 105-114.

Handle: Boston Dynamics' newest design. Jumps 4 feet in the air and zips around at 9 miles per hour. https://www.youtube.com/watch?v=7h8mX9ZMs7g

State of AI Report

October 12, 2021

Nathan Benaich

Ian Hogarth

DeepMind

2021-12-08

Scaling Language Models: Methods, Analysis & Insights from Training *Gopher*

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu and Geoffrey Irving

Language modelling provides a step towards intelligent communication systems by harnessing large repositories of written human knowledge to better predict and understand the world. In this paper, we present an analysis of Transformer-based language model performance across a wide range of model

1. Introduction

Natural language communication is core to intelligence, as it allows ideas to be efficiently shared between humans or artificially intelligent systems. The generality of language allows us to express many intelligence tasks as taking in natural language input and producing natural language output.

MIT Technology Review

Topics

Artificial intelligence / Machine learning

Training a single AI model can emit as much carbon as five cars in their lifetimes

Deep learning has a terrible carbon footprint.

by **Karen Hao**

June 6, 2019

Common carbon footprint benchmarks

in lbs of CO2 equivalent

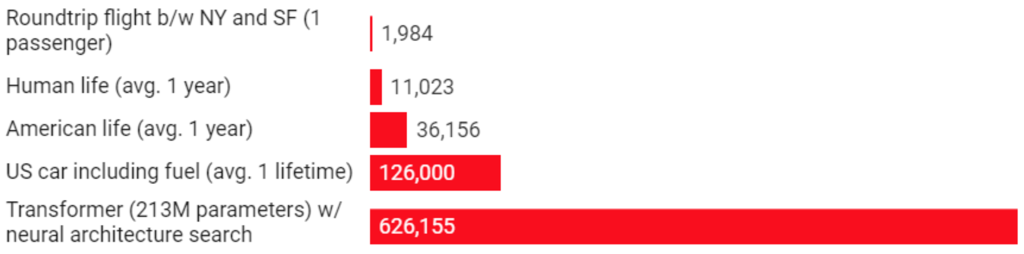


Chart: MIT Technology Review • Source: Strubell et al. • Created with Datawrapper

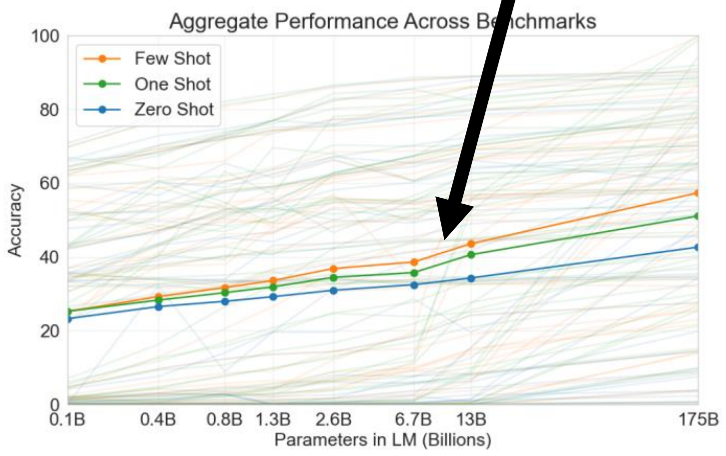
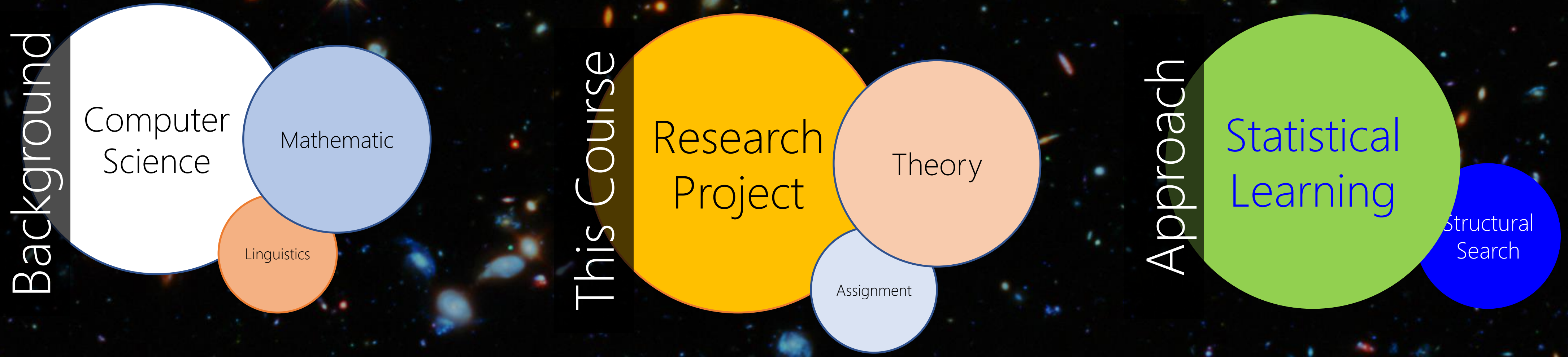


Figure 1.3: Aggregate performance for all 42 accuracy-denominated benchmarks While zero-shot performance improves steadily with model size, few-shot performance increases more rapidly, demonstrating that larger models are more proficient at in-context learning. See Figure 3.8 for a more detailed analysis on SuperGLUE, a standard NLP benchmark suite.

Communication
Understanding
Natural Language Processing

Speech, *Text*, Emotion, …



Task

NLP is AI-hard (why?)

Pipeline of Understanding

Phonetics and Phonology, Morphology, Syntax, Semantics, Pragmatics, Discourse

Ambiguity

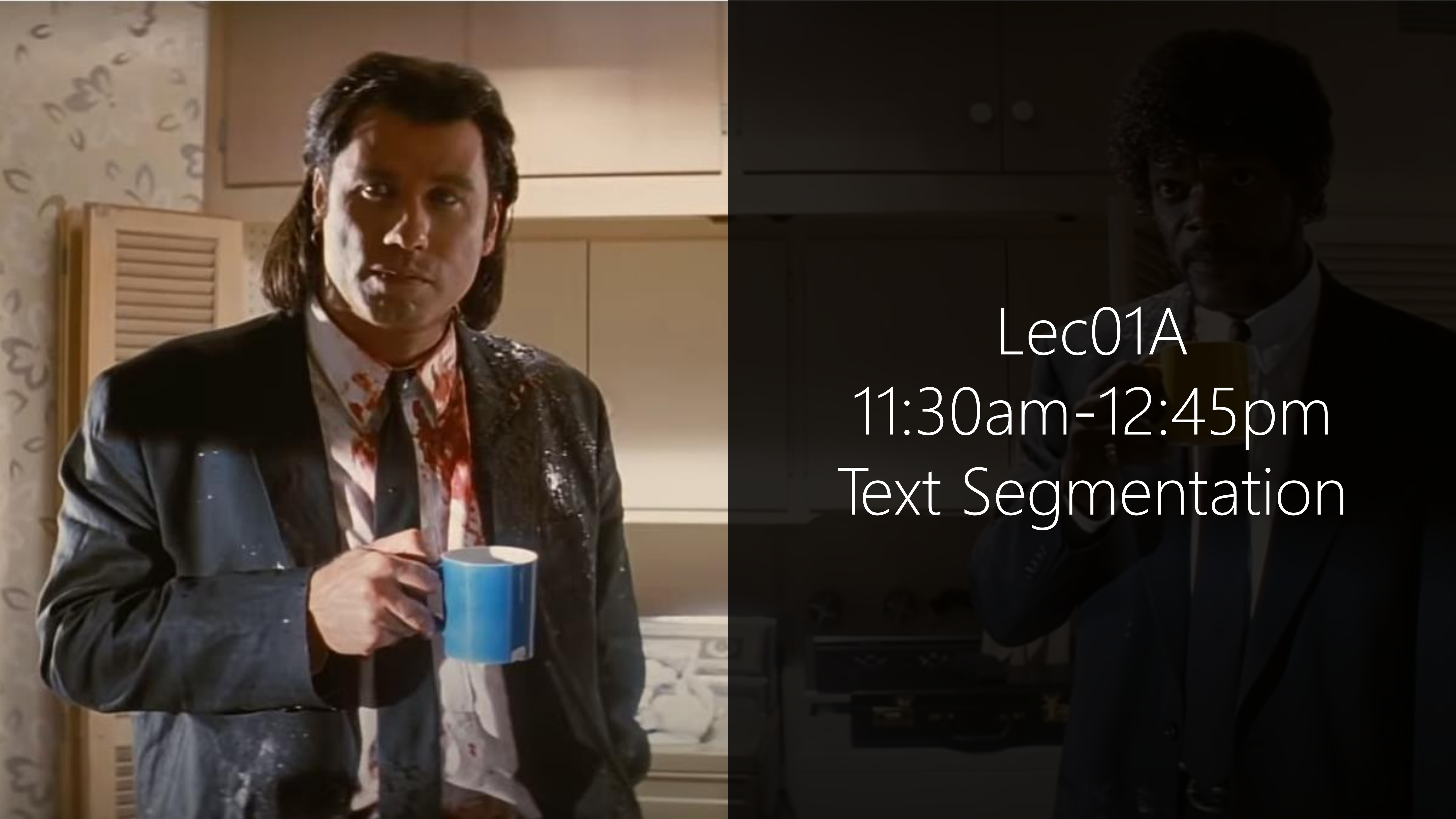
I made her duck

Language is Situated

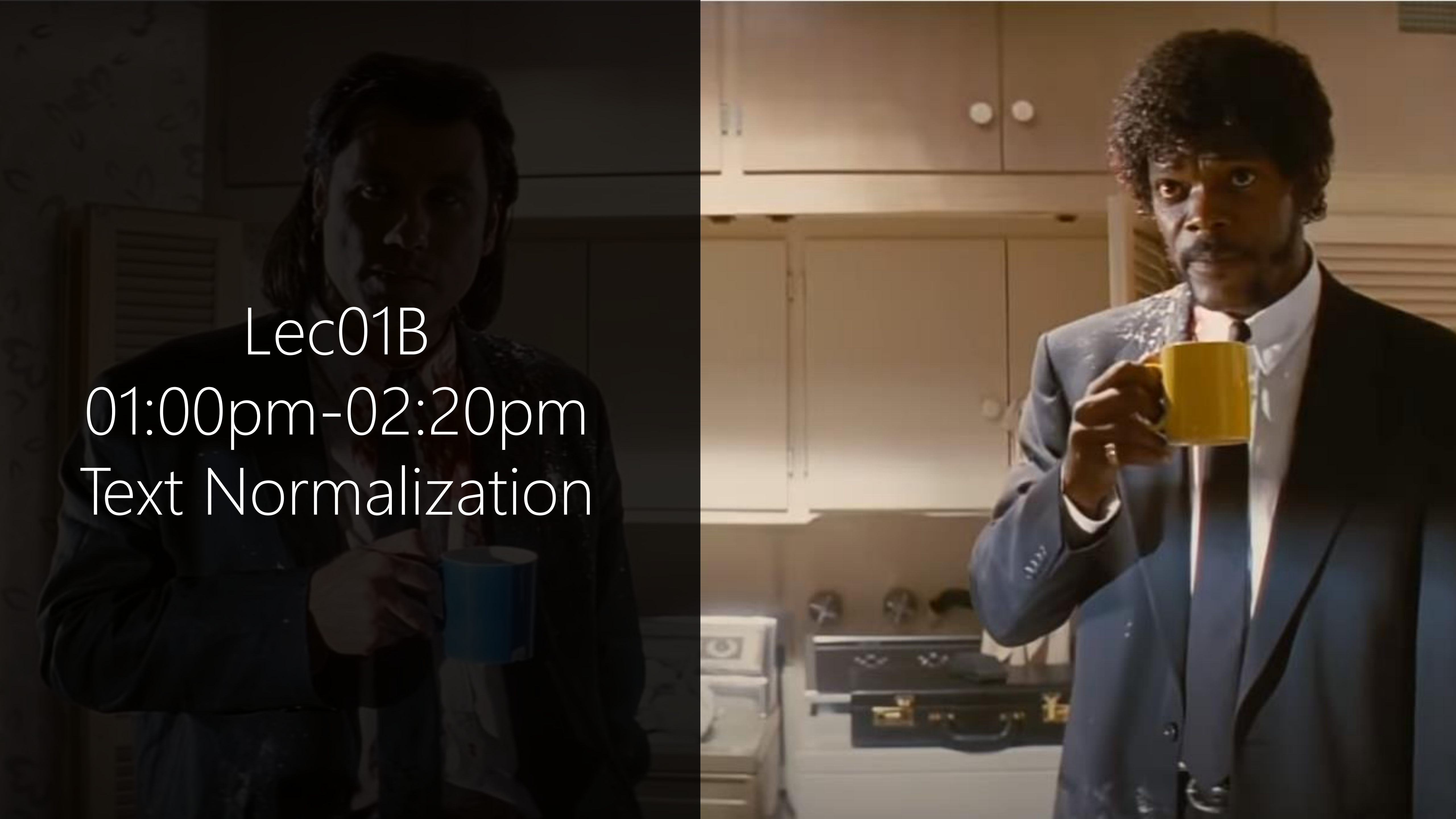
Rationale, Code Switch, Genre, Time, Place, Dialect, Author

Computational Complexity

Discrete, Compositional, Power-law distribution



Lec01A
11:30am-12:45pm
Text Segmentation



Lec01B

01:00pm-02:20pm

Text Normalization

Natural Language Processing

Phonetics and Phonology, Morphology, Syntax, Semantics, Pragmatics, Discourse

Phonetics and Phonology

knowledge about linguistic sounds
how words are pronounced in terms of sequences of sounds
how each of these sounds is realized acoustically

Phonetics and Phonology

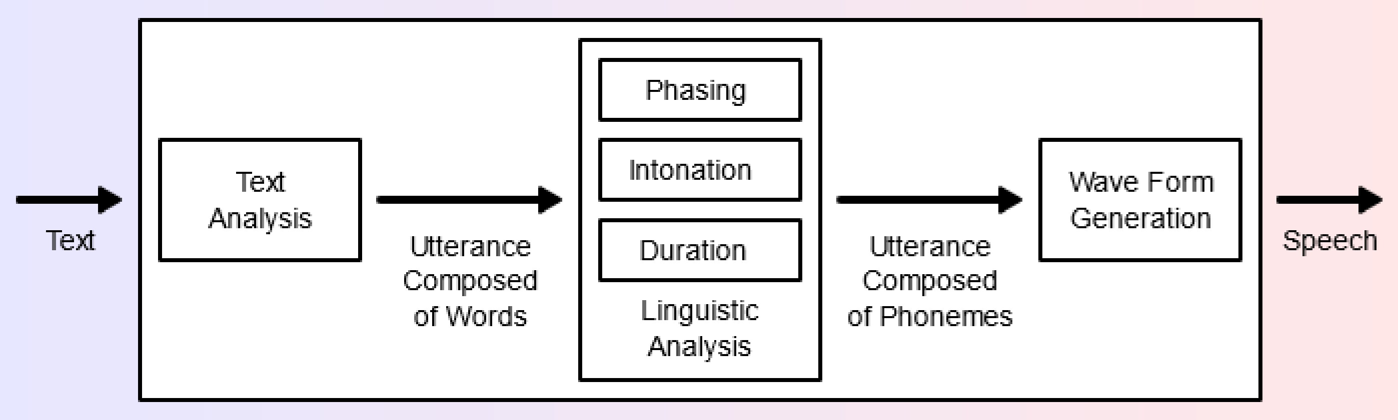
Speech Recognition (SR)

recognize words from an audio signal like in assistants: Alexa, Homepod

Speech Synthesis (Synthesizers)

generate an audio signal from a sequence of words like in Automatic Announcement
Automatic Answering Machine

Phonetics and Phonology



Text-To-Speech System (TTS)

Phone [fəʊn]→Diphones [fə], [əʊ], [ʊn]→much more natural than combining simple phones

Stephen Hawking talks about technology and ACAT (Assistive Context Aware Tool-kit)
https://www.youtube.com/watch?v=TXY8IKFeKZs
https://01.org/acat/
image: Ted S. Warren/AP

Google

Search Google or type a URL

Search by voice

Natural Language Processing

Phonetics and Phonology, **Morphology**, Syntax, Semantics, Pragmatics, Discourse

Morphology

knowledge of the meaningful components of words
producing and recognizing variations of individual words
the way words break down into component parts that carry different meanings
study of words, how they are formed, their relationships in the same language

Morphology

Lexeme

- Also known as Lemma or citation form
- Refer to a same entity or concept
- Change is called *Inflection*
- Inflection Rules:

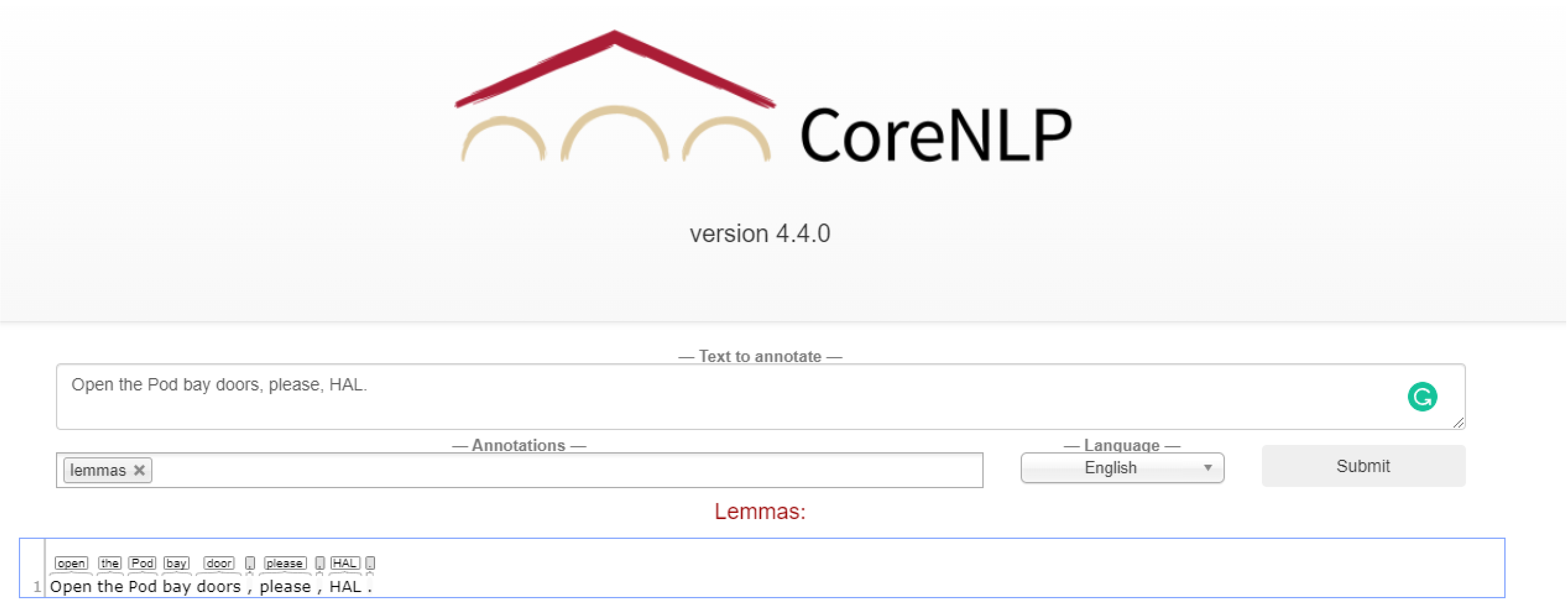
  Singular vs. Plural: index → indexes, indices
  Contractions: cannot → can't
  Tenses: do → did, done, does

Wordforms

- Form new lexemes
- Refer to different entities or concepts or …
- Wordformation Rules:

  Compounding: [Dog][catch][er], [Dish][wash][er]
  Lemma(Dishwashers) = Dishwasher
  Lemma(Dishwashers) ≠ Dish ≠ Wash

Lemmatizer


CoreNLP
version 4.4.0
— Text to annotate —
Open the Pod bay doors, please, HAL.
— Annotations —
lemmas
— Language —
English
Submit
Lemmas:
open the Pod bay door , please , HAL .
1 Open the Pod bay doors , please , HAL .

Natural Language Processing

Phonetics and Phonology, Morphology, Syntax, Semantics, Pragmatics, Discourse

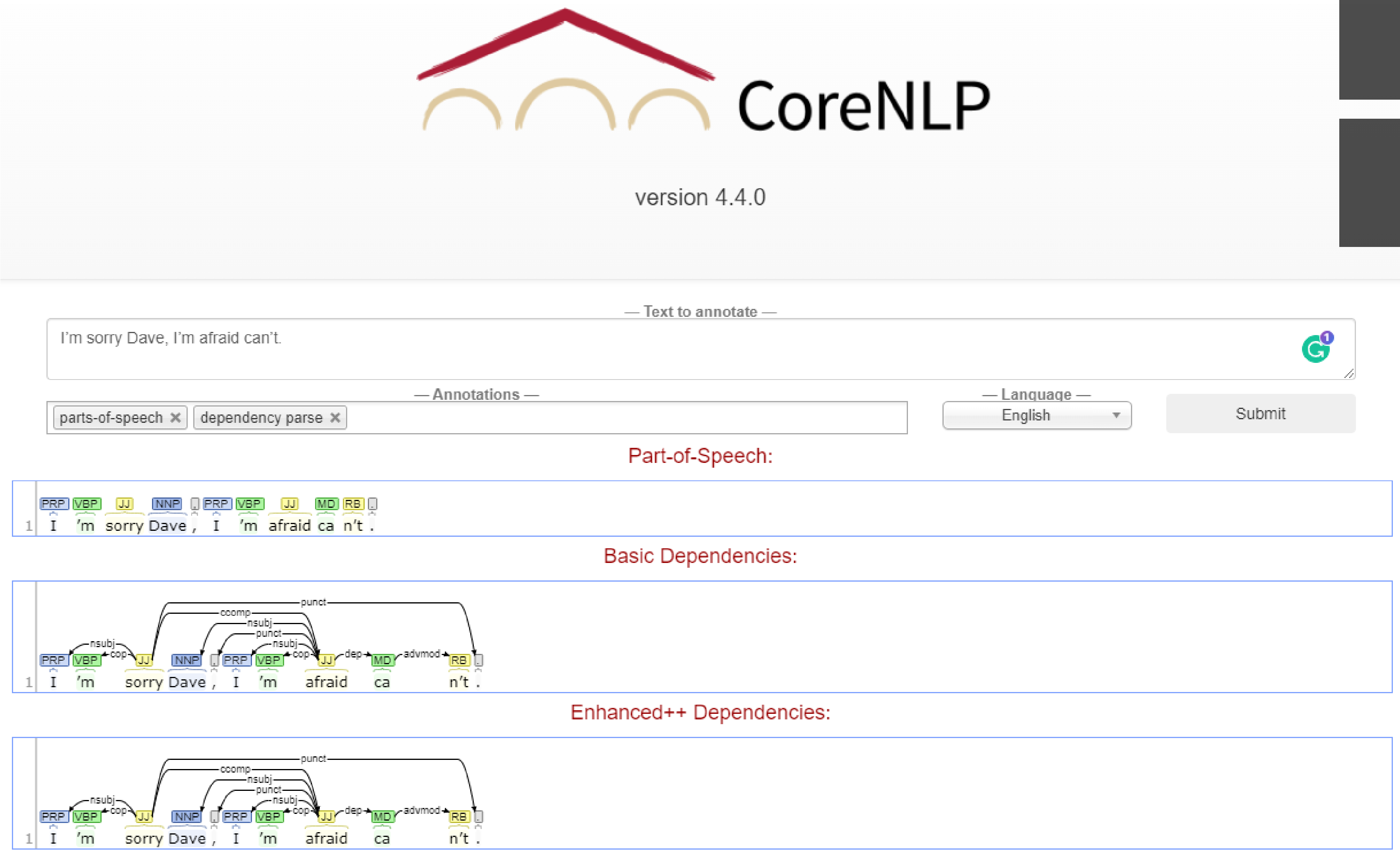
Syntax

knowledge of the structural relationships between words
knowledge needed to *stream* (order) words
moving beyond individual words

HAL: I’m I do, sorry that afraid Dave I’m can’t. ↔ I’m sorry Dave, I’m afraid can’t.

Pos Tagger

Parser



CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential *there*
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun
PRP$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	*to*
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Wh-determiner
WP	Wh-pronoun
WP$	Possessive wh-pronoun
WRB	Wh-adverb

https://cs.nyu.edu/~grishman/jet/guide/PennPOS.html

Natural Language Processing

Phonetics and Phonology, Morphology, Syntax, Semantics, Pragmatics, Discourse

Semantics

knowledge of meaning

Lexical

- The meaning of the words

  cat → a small domesticated carnivorous mammal ...

- Polysemy: one word with two or more distinct meanings

  polyseme is a word or phrase with multiple meanings

  afraid → Scared

  afraid → Politely apologetic

Compositional

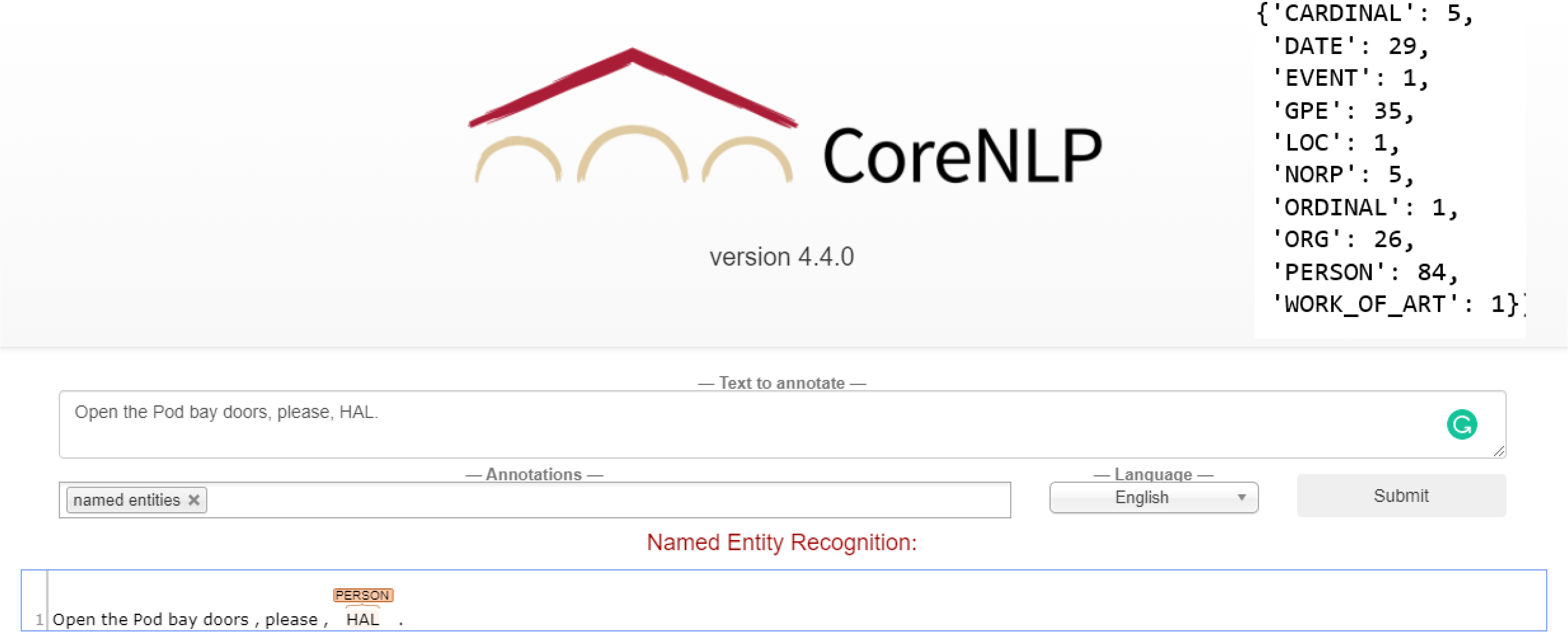
- The meaning of the sentences

  I'm afraid of being sorry

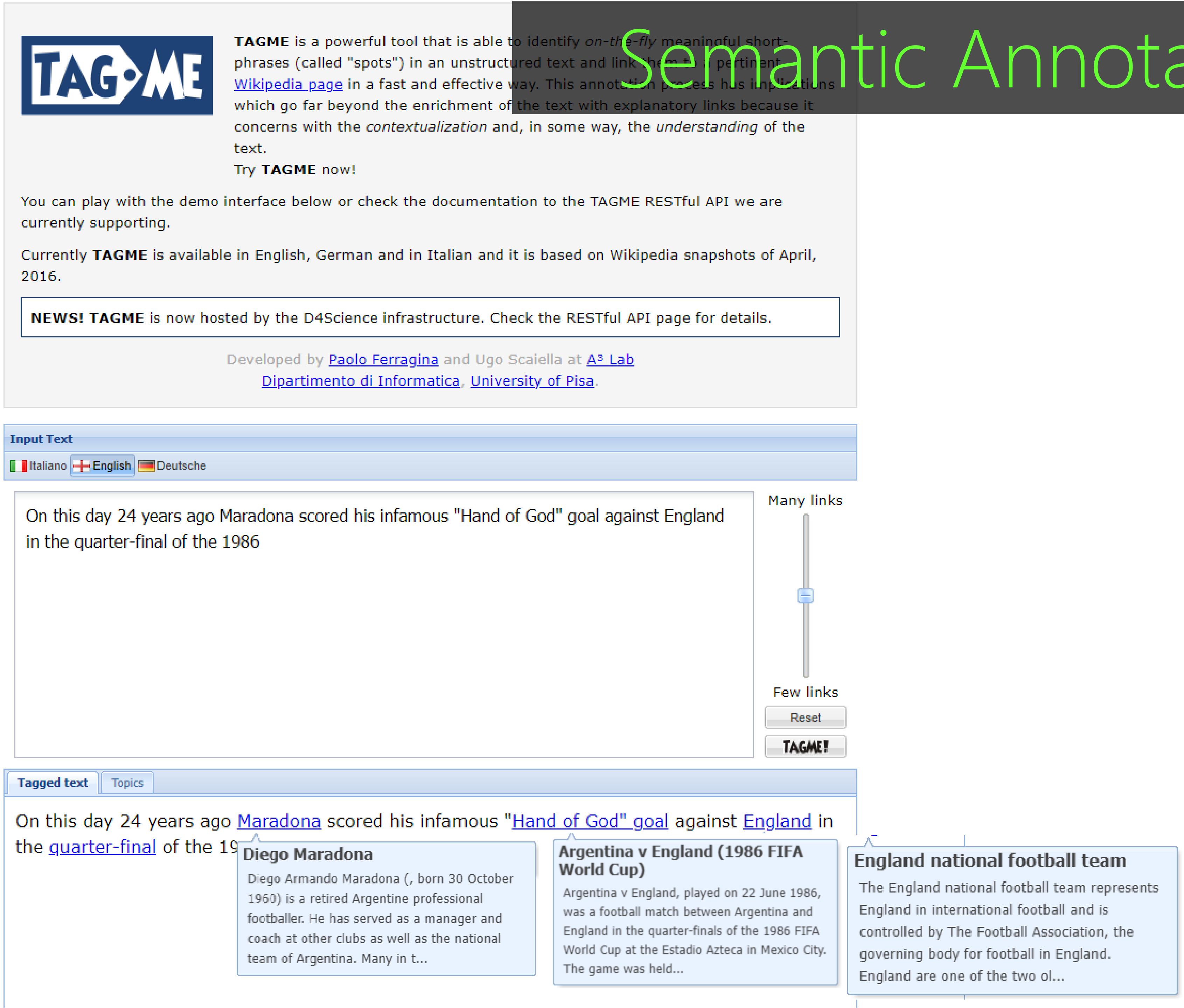
  sorry, I'm afraid I can't

Named Entity Annotators

{'CARDINAL': 5,
 'DATE': 29,
 'EVENT': 1,
 'GPE': 35,
 'LOC': 1,
 'NORP': 5,
 'ORDINAL': 1,
 'ORG': 26,
 'PERSON': 84,
 'WORK_OF_ART': 1}



Semantic Annotators



Natural Language Processing

Phonetics and Phonology, Morphology, Syntax, Semantics, **Pragmatics**, Discourse

Pragmatic

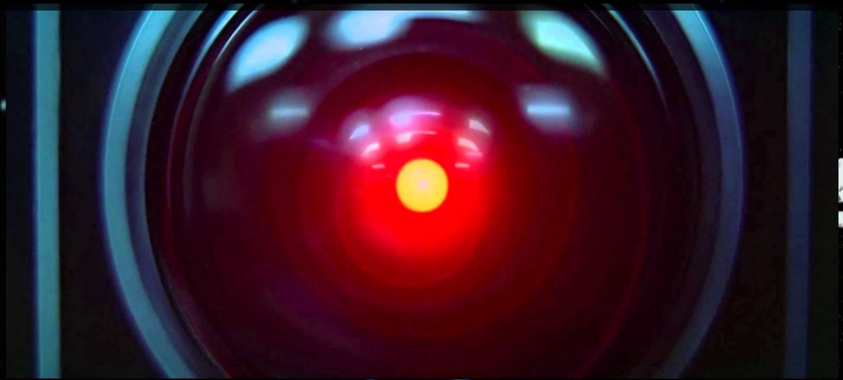
knowledge of the relationship of meaning to the *goals* & *intentions* of the speaker

Syntax: Mono relation between linguistic forms (lexeme)

Semantic: Dyadic relation between lexemes and real-world entities

Pragmatic: Triadic relation between lexemes, world, and the user

Directive: HAL, open the pod bay door.



Pragmatic

knowledge of the relationship of meaning to the *goals* & *intentions* of the speaker

Illocutionary act was introduced into linguistics by the philosopher John Langshaw Austin in his investigation of the various aspects of *speech acts*.

Locution: what was said

"*Is there any salt?*" at the dinner table, a question about the presence of salt

Illocution: what was meant

"*please give me some salt" vs. "Yes" or "No"*

Perlocution: what happened as a result, the actual effect.

to cause somebody to pass the salt.

illocutionary il·lo·cu·tion·ary | \ ˌi-lə-ˈkyü-shə-ˌner-ē

Pragmatic

knowledge of the relationship of meaning to the *goals* & *intentions* of the speaker

John Rogers Searle (/sɜːrl/; born July 31, 1932) is an American philosopher.
Searle, John R. "A Classification of Illocutionary Acts." *Language in Society*, vol. 5, no. 1, Cambridge University Press, 1976, pp. 1–23, http://www.jstor.org/stable/4166848.

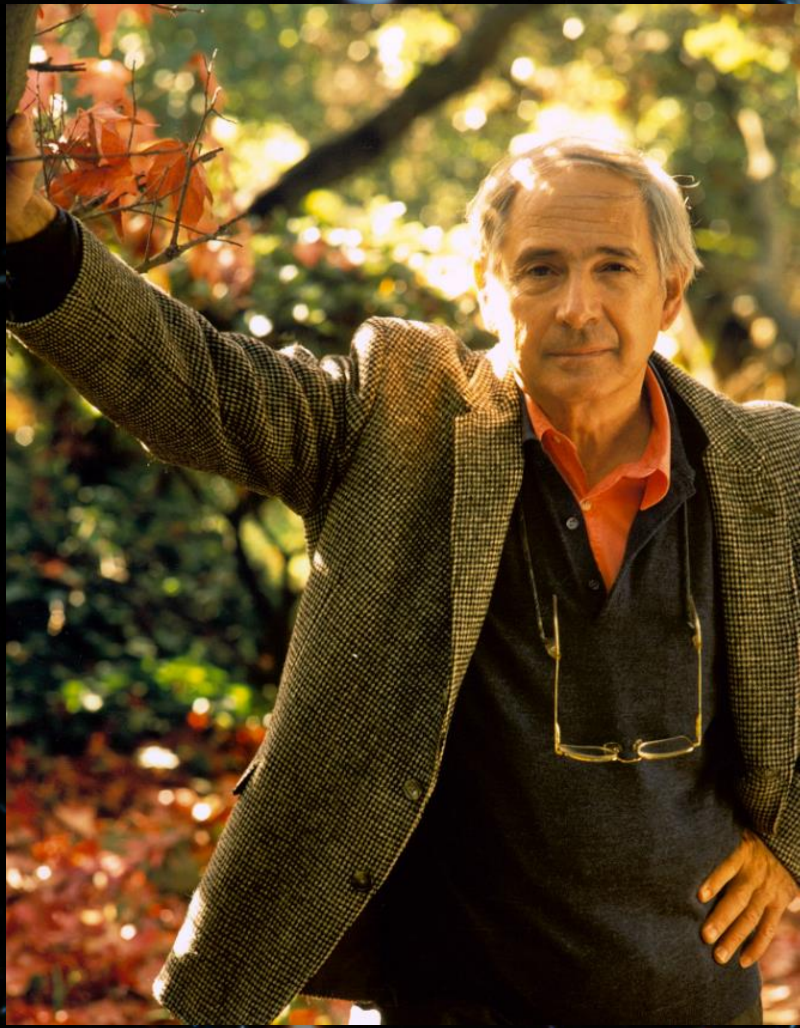
Representative (Assertive) committing the speaker to the truth of the expressed proposition, e.g., asserting, concluding

Directive attempts by the speaker to get the addressee to do something, e.g., advising, requesting

Commissive committing the speaker to a future course of action, e.g., promising, threatening, offering

Expressive expressing a psychological state, e.g., thanking, apologizing, welcoming

Declarative effecting immediate changes in an institutional state of affairs, with extra-linguistic qualities, e.g., declaring war, christening



Natural Language Processing

Phonetics and Phonology, Morphology, Syntax, Semantics, Pragmatics, Discourse

Discourse

knowledge about linguistic units larger than a single utterance

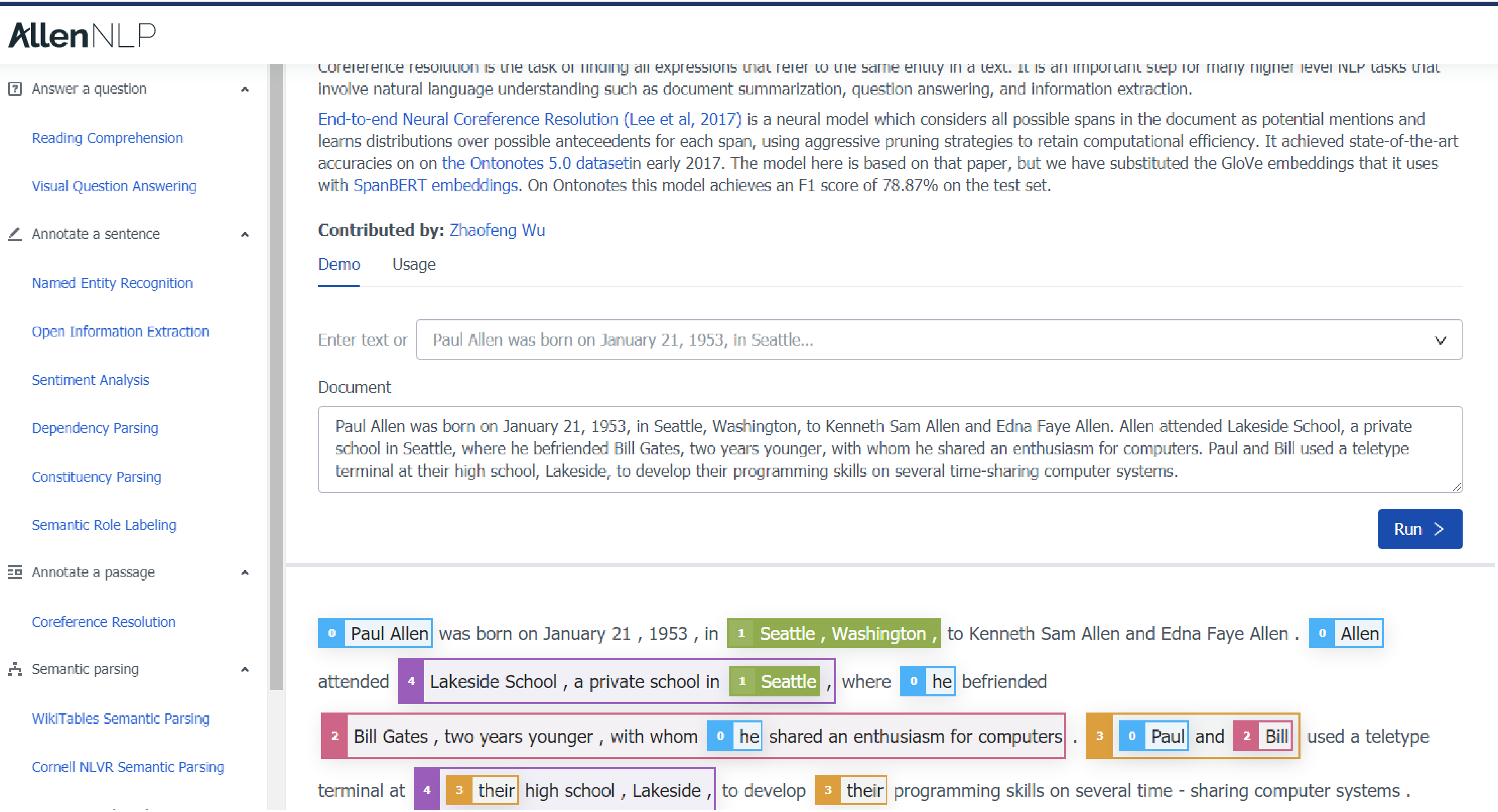
[discourse structure]

logical flow of events, states, propositions that makes for a coherent idea, argument, or story

another kind of pragmatic knowledge but within a session of communication

Co-reference

I made her duck


AllenNLP
Answer a question
Reading Comprehension
Visual Question Answering
Annotate a sentence
Named Entity Recognition
Open Information Extraction
Sentiment Analysis
Dependency Parsing
Constituency Parsing
Semantic Role Labeling
Annotate a passage
Coreference Resolution
Semantic parsing
WikiTables Semantic Parsing
Cornell NLVR Semantic Parsing
Coreference resolution is the task of finding all expressions that refer to the same entity in a text. It is an important step for many higher level NLP tasks that involve natural language understanding such as document summarization, question answering, and information extraction.
End-to-end Neural Coreference Resolution (Lee et al, 2017) is a neural model which considers all possible spans in the document as potential mentions and learns distributions over possible anteceedents for each span, using aggressive pruning strategies to retain computational efficiency. It achieved state-of-the-art accuracies on on the Ontonotes 5.0 datasetin early 2017. The model here is based on that paper, but we have substituted the GloVe embeddings that it uses with SpanBERT embeddings. On Ontonotes this model achieves an F1 score of 78.87% on the test set.
Contributed by: Zhaofeng Wu
Demo
Usage
Enter text or
Paul Allen was born on January 21, 1953, in Seattle...
Document
Paul Allen was born on January 21, 1953, in Seattle, Washington, to Kenneth Sam Allen and Edna Faye Allen. Allen attended Lakeside School, a private school in Seattle, where he befriended Bill Gates, two years younger, with whom he shared an enthusiasm for computers. Paul and Bill used a teletype terminal at their high school, Lakeside, to develop their programming skills on several time-sharing computer systems.
Run
0 Paul Allen was born on January 21 , 1953 , in 1 Seattle , Washington , to Kenneth Sam Allen and Edna Faye Allen . 0 Allen
attended 4 Lakeside School , a private school in 1 Seattle , where 0 he befriended
2 Bill Gates , two years younger , with whom 0 he shared an enthusiasm for computers . 3 0 Paul and 2 Bill used a teletype
terminal at 4 3 their high school , Lakeside , to develop 3 their programming skills on several time - sharing computer systems .

Discourse

knowledge about linguistic units larger than a single utterance

[discourse structure]

logical flow of events, states, propositions that makes for a coherent idea, argument, or story

another kind of pragmatic knowledge but within a session of communication

Discourse Relation

I refused to pay the cobbler the full $95 because he did poor work. (Contingency)

He knows a tasty meal when he eats one. (Temporal)

IBM's stock price rose, but the overall market fell. (Comparison)

I never gamble too far; in particular, I quit after one try. (Expansion)

Penn Discourse Treebank PDTB

The Penn Discourse Treebank Project is an NSF funded project, supported by NSF grants:
IIS-14-22186 (2014-2017)
IIS-14-21067 (2014-2017)
CNS-10-59353 (2011-2013)
IIS-07-05671 (2007-2012)
CNS-02-24417 (2002-2006)

The PDTB 3.0 corpus was released on March 15, 2019 through the Linguistic Data Consortium. N.B. The corpus was updated on February 4, 2020, to include the final versions of two files of *to clause* annotation that were discovered to not have been loaded earlier, as well as several tokens were inadvertently omitted on the assumption that they were duplicates, when they weren't. Specific changes/additions are recorded in the file pdtb3-revision-jan-2020.txt.

For an introduction to PDTB 3.0 and the PDTB 3.0 Annotation Manual, click here. Please visit the tools page for technical support.

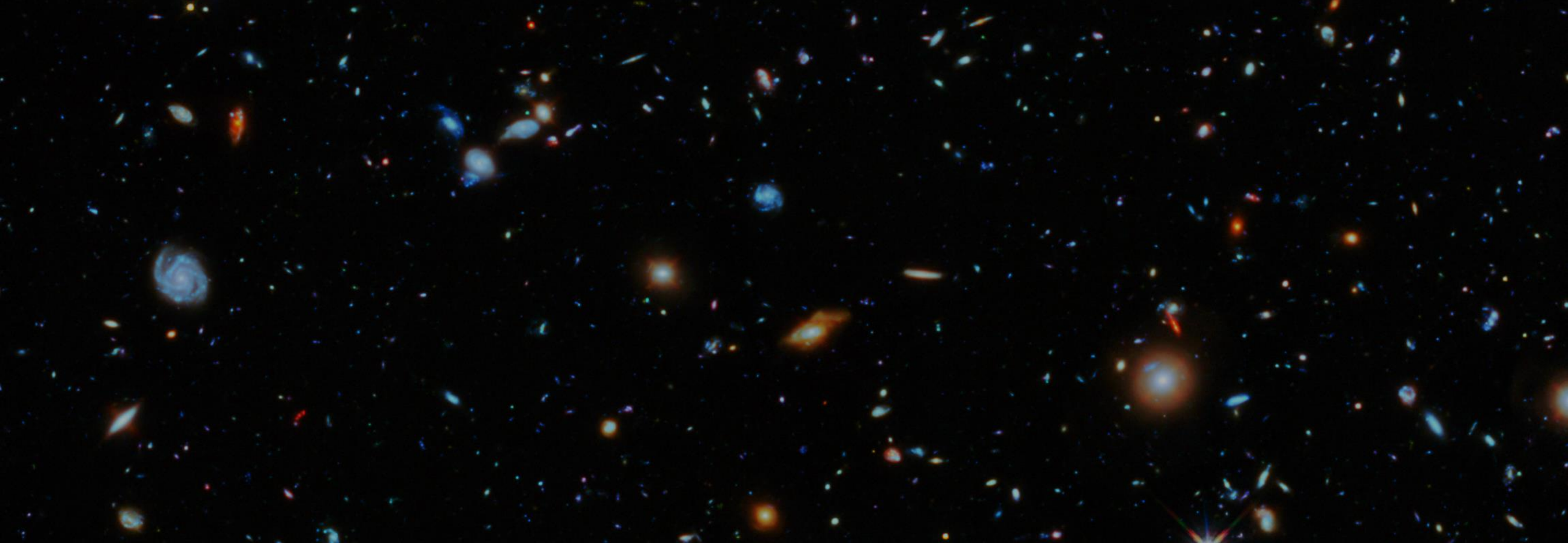
The PDTB 2.0 corpus is still available from this LDC page.

The Penn Discourse Treebank (PDTB) is a large scale corpus annotated with information related to discourse structure and discourse semantics. While there are many aspects of discourse that are crucial to a complete understanding of natural language, the PDTB focuses on encoding *discourse relations*. The annotation methodology follows a lexically-grounded approach. The PDTB has strived to maintain a theory-neutral approach with respect to the nature of high-level

Discourse

due to background knowledge, linguistic elements are *latent* or *implicit!*

great movie by Stanley Kubrick



Let's build an NLP model

Phonetics and Phonology, Morphology, Syntax, Semantics, Pragmatics, Discourse

Lec01B
01:00pm-02:20pm
Text Normalization

Text Segmentation

dividing text into linguistic units, such as words, sentences, or topics

What should be considered as word?

- Disfluencies in *utterances*
  - Fragments: broken-off repeated words: miss- misspelled, you- yourself
  - Fillers: non-lexical: huh, uh, erm, um, well, so, like, hmm
- Punctuations , . : ; ? ! Special Chars $, /, ...
  - part-of-speech tagging
  - parsing
  - speech synthesis
- Morphemes:
  - smallest meaning-bearing unit of a language
  - ‘unlikeliest’ : morphemes [un-], [likely], [-est]

Word Segmentation → Tokenization

- Word Boundary: Whitespace (natural word delimiter)
  - Split()

- Exceptions
  - New York, rock 'n' roll
  - Contractions: I'm
  - Languages: Japanese | Chinese | Thai don't have spaces between words
  - Emoticons: :)
  - Hashtags: #nlproc.

As Chen et al. (2017) point out, this could be treated as 3 words ('Chinese Treebank' segmentation):

(2.5) 姚明 进入 总决赛
YaoMing reaches finals

or as 5 words ('Peking University' segmentation):

(2.6) 姚 明 进入 总 决赛
Yao Ming reaches overall finals

Finally, it is possible in Chinese simply to ignore words altogether and use characters as the basic elements, treating the sentence as a series of 7 characters:

(2.7) 姚 明 进 入 总 决 赛
Yao Ming enter enter overall decision game

Adversarial Multi-Criteria Learning for Chinese Word Segmentation

Xinchi Chen, Zhan Shi, Xipeng Qiu,* Xuanjing Huang
Shanghai Key Laboratory of Intelligent Information Processing, Fudan University
School of Computer Science, Fudan University
825 Zhangheng Road, Shanghai, China
{xinchichen13,zshi16,xpqiu,xjhuang}@fudan.edu.cn

Abstract

Different linguistic perspectives causes many diverse segmentation criteria for Chinese word segmentation (CWS). Most existing methods focus on improve the performance for each single criterion. However, it is interesting to exploit these different criteria and mining their common underlying knowledge. In this paper, we propose adversarial multi-criteria learning

Corpora	Yao	Ming	reaches	the final	
CTB	姚明		进入	总决赛	
PKU	姚	明	进入	总	决赛

Table 1: Illustration of the different segmentation criteria.

Recently, some efforts have been made to exploit heterogeneous annotation data for Chinese word segmentation or part-of-speech tagging (Jiang et al., 2009; Sun and Wan, 2012; Qiu et al.,

Chen, X., Shi, Z., Qiu, X., and Huang, X. (2017). Adversarial multi-criteria learning for Chinese word segmentation. In ACL 2017, 1193–1203.

As Chen et al. (2017) point out, this could be treated as 3 words ('Chinese Treebank' segmentation):

(2.5) 姚明 进入 总决赛
YaoMing reaches finals

characters are at a reasonable semantic level for most applications

take characters as words!

Not for Japanese and Thai!

(2.7) 姚 明 进 入 总 决 赛
Yao Ming enter enter overall decision game

Adversarial Multi-Criteria Learning for Chinese Word Segmentation

Xinchi Chen, Zhan Shi, Xipeng Qiu, Xuanjing Huang
Shanghai Key Laboratory of Intelligent Information Processing, Fudan University
School of Computer Science, Fudan University
825 Zhangheng Road, Shanghai, China
{xinchichen13,zshi16,xpqiu,xjhuang}@fudan.edu.cn

Abstract

Different linguistic perspectives causes many diverse segmentation criteria for Chinese word segmentation (CWS). Most existing methods focus on improve the performance for each single criterion. However, it is interesting to exploit these different criteria and mining their common underlying knowledge. In this paper, we propose adversarial multi-criteria learning

Corpora	Yao	Ming	reaches	the final	
CTB	姚明		进入	总决赛	
PKU	姚	明	进入	总	决赛

Table 1: Illustration of the different segmentation criteria.

Recently, some efforts have been made to exploit heterogeneous annotation data for Chinese word segmentation or part-of-speech tagging (Jiang et al., 2009; Sun and Wan, 2012; Qiu et al.,

Chen, X., Shi, Z., Qiu, X., and Huang, X. (2017). Adversarial multi-criteria learning for Chinese word segmentation. In ACL 2017, 1193–1203.

Word Boundaries → Space

- Regular Expressions (RE): Finite State Automata
  - Alphabetical: [a-zA-Z]*
  - Alpha-numerical: [a-zA-Z0-9]*
  - Punctuations: Ph.D., AT&T, cap'n
  - Special Chars

    Currency $45.55

    Dates (01/02/06)

    URLs http://www.stanford.edu

    Twitter hashtags #nlproc

    Email hfani@uwindsor.ca

Learn to Tokenize

Learn to Tokenize

Word→ Subword → Word

- Byte-Pair Encoding (BPE)

Sennrich, et al., (2016). Neural machine translation of rare words with subword units. In ACL 2016.

- Wordpiece

Wu et al. (2016) Google's neural machine translation system: Bridging the gap between human and machine translation." arXiv.

- MaxMatch in BERT

Devlin et al. (2019). BERT: Pretraining of deep bidirectional transformers for language understanding. In NAACL HLT.

- SentencePiece

Kudo, T. and Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In EMNLP.

Learn to Tokenize

Byte-Pair Encoding (BPE) : Data Compression

Gage, P. (1994). A new algorithm for data compression. The C Users Journal, 12(2), 23–38.

aaabdaaabac → *pair 'aa' occurs most often* → *replace it with a char (byte) that is not used 'Z'*

→ ZabdZabac; Z=aa

Learn to Tokenize

Byte-Pair Encoding (BPE) : Data Compression

Gage, P. (1994). A new algorithm for data compression. The C Users Journal, 12(2), 23–38.

aaabdaaabac → *pair 'aa' occurs most often* → *replace it with a char (byte) that is not used 'Z'*

→ ZabdZabac; Z=aa

→ ZabdZabac → *pair 'ab' occurs most often* → *replace it with 'Y'*

→ ZYdZYac; Y=ab, Z=aa

Learn to Tokenize

Byte-Pair Encoding (BPE) : Data Compression

Gage, P. (1994). A new algorithm for data compression. The C Users Journal, 12(2), 23–38.

aaabdaaabac → *pair 'aa' occurs most often* → *replace it with a char (byte) that is not used 'Z'*

→ ZabdZabac; Z=aa

→ ZabdZabac → *pair 'ab' occurs most often* → *replace it with 'Y'*

→ ZYdZYac; Y=ab, Z=aa

→ ZYdZYac → *byte pair 'ZY' with 'X'*

→ XdXac; X=ZY, Y=ab, Z=aa

Learn to Tokenize

Byte-Pair Encoding (BPE) : Data Compression

Gage, P. (1994). A new algorithm for data compression. The C Users Journal, 12(2), 23–38.

aaabdaaabac → *pair 'aa' occurs most often* → *replace it with a char (byte) that is not used 'Z'*

→ ZabdZabac; Z=aa

→ ZabdZabac → *pair 'ab' occurs most often* → *replace it with 'Y'*

→ ZYdZYac; Y=ab, Z=aa

→ ZYdZYac → *byte pair 'ZY' with 'X'*

→ XdXac; X=ZY, Y=ab, Z=aa

no pairs of bytes that occur more than 1 → no more compression

Learn to Tokenize

Byte-Pair Encoding (BPE) : *Train*

Dictionary$^{(0)}$	Vocabulary$^{(0)}$	get_stats()	most frequent pair
'l o w </w>' 'l o w e r </w>' 'n e w e s t </w>' 'w i d e s t </w>' 'e s t e r </w>'	l o w e r t n s d </w>	(l, l) → 0 (l, o) → 2 (l, w) → 0 .... [all pairs of 'l' with others] (o, l) → 0 (o, o) → 0 .... [all pairs of 'o' with others] (e, s) → 3 .... [all pairs of 'e' with others] (o, </w>) → 0	(e, s)

Learn to Tokenize

Byte-Pair Encoding (BPE) : *Train*

Dictionary(1)	Vocabulary(1)	get_stats()	most frequent pair
'l o w </w>' 'l o w e r </w>' 'n e w es t </w>' 'w i d es t </w>' 'es t e r </w>'	l o w e r t n s d </w> es	(l, l) → 0 (l, o) → 2 (l, w) → 0 .... [all pairs of 'l' with others] (o, l) → 0 (o, o) → 0 .... [all pairs of 'o' with others] (e, s) → 3 → 0 .... [all pairs of 'e' with others] (o, </w>) → 0 .... (es, l) → 0 .... [all pairs of 'es' with others] (es, t) → 3	(e, s) → (es, t)

Learn to Tokenize

Byte-Pair Encoding (BPE) : *Train*

Dictionary(2)	Vocabulary(2)	get_stats()	most frequent pair
'l o w </w>'	l	(l, l) → 0	(es, t) → ?
'l o w e r </w>'	o	(l, o) → 2	
'n e w est </w>'	w	....	
'w i d est </w>'	e	(es, l) → 0	
'est e r </w>'	r	.... [all pairs of 'es' with others]	
	t	(es, t) → 0	
	n	(est, l) → 0	
	s	....[all pairs of 'est' with others]	
	d	(est, </w>) → 2	
	</w>		
	es		
	est		

Learn to Tokenize

Byte-Pair Encoding (BPE) : *Train*

Dictionary$^{(n)}$	Vocabulary$^{(n)}$	get_stats()	most frequent pair
'low</w>' 'lower</w>' 'newest</w>' 'widest</w>' 'ester</w>'	l,o,w,e,r,t,n,s,d,</w>, es est lo low, … newer wider	…	…

Learn to Tokenize

Byte-Pair Encoding (BPE) : *Test*

Vocabulary$^{(n)}$
l
o
w
e
r
t
n
s
d
</w>
es
est
lo
low
…
newer
wider
….

Input
… l o w es t p a r t o f t h e n e w e r h o u s e ….
… l o w est p a r t o f t h e n e w e r h o u s e ….
… lo w est p a r t o f t h e n e w e r h o u s e ….
… low est p a r t o f t h e n e w e r h o u s e ….
… low est p a r t o f t h e ne w e r h o u s e ….
… low est p a r t o f t h e new e r h o u s e ….
… low est p a r t o f t h e new er h o u s e ….
… low est p a r t o f t h e newer h o u s e ….

Learn to Tokenize

Byte-Pair Encoding (BPE) : *Test*

Vocabulary$^{(n)}$
l
o
w
e
r
t
n
s
d
</w>
es
est
lo
low
…
….

Input

… l o w es t p a r t o f t h e n e w e r h o u s e ….
… l o w est p a r t o f t h e n e w e r h o u s e ….
… lo w est p a r t o f t h e n e w e r h o u s e ….
… low est p a r t o f t h e n e w e r h o u s e ….
… low est p a r t o f t h e ne w e r h o u s e ….
… low est p a r t o f t h e new e r h o u s e ….
… low est p a r t o f t h e new er h o u s e ….
… low est p a r t o f t h e newer h o u s e ….

What if `lowest` is not built during training?

Learn to Tokenize

Byte-Pair Encoding (BPE)

1. Help to generalize to produce *unseen* words
2. Rare words into subword units is sufficient for translation

100 rare tokens in German training data and they are translatable from English via smaller units!

https://github.com/rsennrich/subword-nmt

function BYTE-PAIR ENCODING(strings C, number of merges k) **returns** vocab V

$V \leftarrow$ all unique characters in C # initial set of tokens is characters
for $i = 1$ **to** k **do** # merge tokens til k times
$t_L, t_R \leftarrow$ Most frequent pair of adjacent tokens in C
$t_{NEW} \leftarrow t_L + t_R$ # make new token by concatenating
$V \leftarrow V + t_{NEW}$ # update the vocabulary
Replace each occurrence of t_L, t_R in C with t_{NEW} # and update the corpus
return V

Figure 2.13 The token learner part of the BPE algorithm for taking a corpus broken up into individual characters or bytes, and learning a vocabulary by iteratively merging tokens. Figure adapted from Bostrom and Durrett (2020).

Learn to Tokenize

Wordpiece

Same as BPE but:

- Merging the pairs that *minimizes language model likelihood* of the training data.
- </w> appears at the beginning of words

Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi
yonghui,schuster,zhifengc,qvl,mnorouzi@google.com

Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, Jeffrey Dean

Learn to Tokenize

MaxMatch in BERT

Devlin et al. (2019). BERT: Pretraining of deep bidirectional transformers for language understanding. In NAACL HLT.

```
function MAXMATCH(string, dictionary) returns list of tokens T

    if string is empty
        return empty list
    for i ← length(sentence) downto 1
        firstword = first i chars of sentence
        remainder = rest of sentence
        if InDictionary(firstword, dictionary)
            return list(firstword, MaxMatch(remainder,dictionary) )
```

Learn to Tokenize

MaxMatch in BERT

Devlin et al. (2019). BERT: Pretraining of deep bidirectional transformers for language understanding. In NAACL HLT.

unaffable
- [u][naffable]
- [un][affable]
- [una][ffable]
- [unaf][fable]
- [unaff][able]
- [unaffa][ble]
- [unaffab][le]
- [unaffabl][e]
- [unaffable]

function MAXMATCH(string, dictionary) **returns** list of tokens T

    if string is empty
        return empty list
    for $i \leftarrow$ *length*(sentence) **downto** 1
        firstword = first i chars of *sentence*
        remainder = rest of *sentence*
        if InDictionary(*firstword*, *dictionary*)
            return list(firstword, MaxMatch(remainder,dictionary))

Learn to Tokenize

MaxMatch in BERT

Devlin et al. (2019). BERT: Pretraining of deep bidirectional transformers for language understanding. In NAACL HLT.

unaffable

- [u][naffable] return
- [un][affable]
- [una][ffable]
- [unaf][fable]
- [unaff][able]
- [unaffa][ble]
- [unaffab][le]
- [unaffabl][e]
- [unaffable]

function MAXMATCH(string, dictionary) **returns** list of tokens T

    if string is empty
        return empty list
    for $i \leftarrow$ *length*(sentence) **downto** 1
        firstword = first *i* chars of *sentence*
        remainder = rest of *sentence*
        if InDictionary(*firstword*, *dictionary*)
            return list(firstword, MaxMatch(remainder,dictionary))

Learn to Tokenize

MaxMatch in BERT

Devlin et al. (2019). BERT: Pretraining of deep bidirectional transformers for language understanding. In NAACL HLT.

unaffable

[u][naffable] return

[un][affable]

{'un'} U [affable]

[a][ffable]

[af][fable]

[aff][able]

...

[una][ffable]

[unaf][fable]

[unaff][able]

[unaffa][ble]

[unaffab][le]

[unaffabl][e]

```
function MAXMATCH(string, dictionary) returns list of tokens T

    if string is empty
        return empty list
    for i ← length(sentence) downto 1
        firstword = first i chars of sentence
        remainder = rest of sentence
        if InDictionary(firstword, dictionary)
            return list(firstword, MaxMatch(remainder,dictionary) )
```

Learn to Tokenize

MaxMatch in BERT

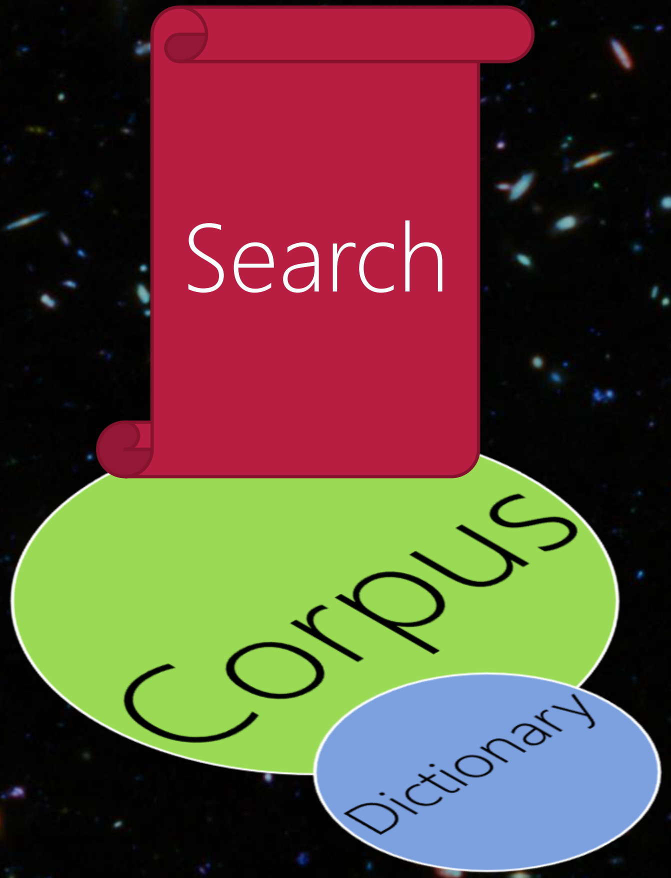
Devlin et al. (2019). BERT: Pretraining of deep bidirectional transformers for language understanding. In NAACL HLT.

unaffable	→	[un, ##aff, ##able]
intention	→	[intent, ##ion]
unwanted running	→	[un, ##want, ##ed, runn, ##ing]

→ marking as internal subwords that do not start words

Learn to Tokenize

Unsupervised or Weakly Supervised (Dictionary)



Learn to Tokenize

Supervised

Corpus + Label

Sentence	Word Boundary by Oracle
i_am_sorry.	[0,0], [2,3], [5,9]
open_the_door	[0,3], [5,7], [9, 12]
hello_world!	[0,4], [6, 10]
great_news,_mary!	[0, 4], [6, 9], [12, 15]

Learn to Tokenize

Supervised (Boolean Classifier) : *Train*

Corpus
+
Label

Sentence	Word Boundary by Oracle
I_am_sorry.	[0,0], [2,3], [5,9]
Open_the_door	[0,3], [5,7], [9, 12]
Hello_world!	[0,4], [6, 10]
Great_news,_Mary!	[0, 4], [6, 9], [12, 15]

Sentence	Word Boundary by Oracle
I_	0 SEP
_a	0 SEP
am	1 APN
m_	0 SEP
_s	1 APN
so	1 APN
or	1 APN
…	…

$$f(x_i, x_{i+1}) \rightarrow \{0, 1\}$$

Learn to Tokenize

Supervised (Boolean Classifier) : *Test*

natural_language

Sentence	$f(x_i, x_{i+1})$	Truth	Error
na	1 APN	1 APN	0
at	1 APN	1 APN	0
tu	1 APN	1 APN	0
ur	1 APN	1 APN	0
ra	0 SEP	1 APN	1
al	1 APN	1 APN	0
l_	1 APN	0 SEP	1
_l	1 APN	0 SEP	1
la	0 SEP	1 APN	1
....	....	....	

[natur][al_l][anguage]

Learn to Tokenize

Supervised (Boolean Classifier)

State	Recognized words	Partial word	Incoming chars	Next Action
state0	[ ]	ϕ	[我去过火车站那边]	SEP
state1	[ ]	我	[去过火车站那边]	SEP
state2	[我]	去	[过火车站那边]	SEP
state3	[我,去]	过	[火车站那边]	SEP
state4	[我,去,过]	火	[车站那边]	APP
state5	[我,去,过]	火车	[站那边]	APP
state6	[我,去,过]	火车站	[那边]	SEP
state7	[我,去,过,火车站]	那	[边]	APP
state8	[我,去,过,火车站]	那边	[ ]	FIN
state9	[我,去,过,火车站,那边]	ϕ	[ ]	- -

Table 1: A transition based word segmentation example.

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 839–849
Vancouver, Canada, July 30 - August 4, 2017. ©2017 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17-1078

Neural Word Segmentation with Rich Pretraining

Jie Yang* and **Yue Zhang*** and **Fei Dong**
Singapore University of Technology and Design
{jie_yang, fei_dong}@mymail.sutd.edu.sg
yue_zhang@sutd.edu.sg

Text Segmentation

dividing text into linguistic units, such as words, sentences, or topics

Sentence Segmentation

- Rule-based: Regular Expression
  - Stanford's CoreNLP
  - Combined by word segmentation (Tokenizer)

Sentence Segmentation

- Boundary markers
  - Exclamation (!)
  - Question (?)
  - Period (.)
    - Abbreviation: Mr. or Inc.

      ABS Inc. hold a conference.
    - Both

      The conference is in ABS Inc.

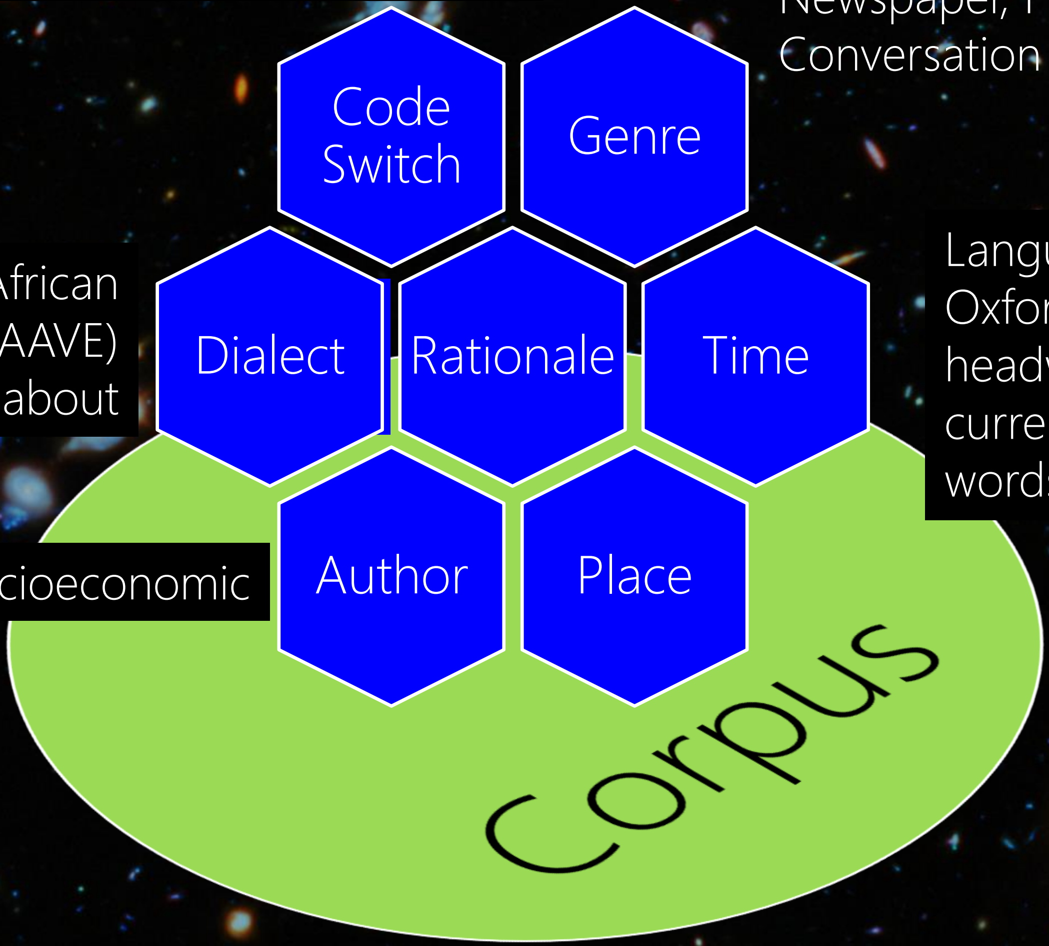
Sentence Segmentation

- Learn to Segment

  Learn to label (.) as sentence marker or abbreviation marker or both *heads up!*

dost tha or ra- hega … dont wory … but dherya rakhe
[he was and will remain a friend … don't worry … but have faith]

Newspaper, Fiction, non-Fiction, Academic, Social, Conversation → Disfluencies



Standard American English (SAE) vs. African American Vernacular English (AAVE)
iont: I don't, talmebout: talking about

Language changes over time
Oxford Dictionary has 273,000 headwords; 171,476 of them being in current use, 47,156 being obsolete words

Age, Gender, Race, Socioeconomic

Corpus (*plural* Corpora)

Brown University

English, newspaper, fiction, non-fiction, academic, etc., 1963–64

#Documents = Size = 500, #Tokens = 1 M, #Vocab = Unique Tokens = Types = 38 K

Switchboard

American English, Telephone Conversations between strangers, Early 1990s

#Conversations = Size = 2430, #Tokens = 2.4 M, #Vocab = Unique Tokens = Types = 20 K

Google N-grams

English, Google Books

#Tokens = 1 G, #Vocab = Unique Tokens = Types = 13 M

Datasheets (Data Statements) for Datasets

Emily M. Bender, Batya Friedman, ACL (2018)
Gebru, Timnit, et al. (2018)

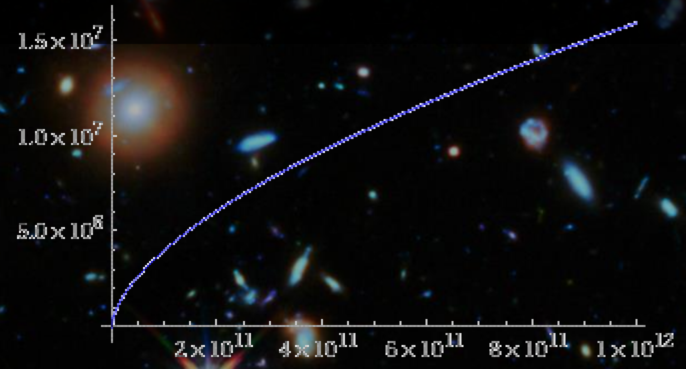
direct stakeholders. For example, Speer (2017) found that a sentiment analysis system rated reviews of Mexican restaurants as more negative than other types of food with similar star ratings, because of associations between the word *Mexican* and words with negative sentiment in the larger corpus on which the word embeddings were trained. (See also Kiritchenko and Mohammad,

Herdan's Law or Heaps' Law

Herdan, G. (1960). Type-token mathematics. The Hague, Mouton.
Heaps, H. S. (1978). Information retrieval. Computational and theoretical aspects. Academic Press.

$$|V| = kN^{\beta}; \; 0 < \beta < 1$$

k and β are positive constants.
The value of β depends on the corpus size and the genre, but at least for the large corpora β ranges from 0.67 to 0.75.



Text Normalization

putting tokens in a *standard format*

choosing a single normal form for tokens with multiple forms

USA $\rightarrow$ *US*

ain't $\rightarrow$ *am not*

Text Normalization

- Case Folding

  mapping everything to lower (upper) case

- Lemmatization

  converts the word to its meaningful base form (lemma).
  The same word may have multiple different Lemmas

- Stemming

  removes or stems the last few characters of a word
  often leading to incorrect meanings and spelling

- Autocorrection

Text Normalization

Case Folding

Positive Impact: ‘USA’ vs. ‘usa’

Information Retrieval, Speech Recognition

Negative Impact: ‘US’ the country vs. ‘us’ the pronoun

Sentiment Analysis, Text Classification, Information Extraction, Machine Translation

Text Normalization

Lemmatization

Polysemy

the association of one word with two or more distinct meanings

a polyseme is a word or phrase with multiple meanings

saw (noun) vs. *saw* (verb) $\longrightarrow$ see

Text Normalization

Lemmatization : Stemming

crude, chopping off word-final stemming affixes

aim is not to produce a linguistic root, but *to improve performance*

Porter, M. F. (1980). An algorithm for suffix stripping. Program, 14(3), 130–137.

Errors of Commission		Errors of Omission	
organization	organ	European	Europe
doing	doe	analysis	analyzes
numerical	numerous	noise	noisy
policy	police	sparse	sparsity

A commission error is an error made due to using an item in the wrong context.

Text Normalization

Spelling Correction

via

Minimum Edit Distance : Word Similarity (Distance)

Text Normalization

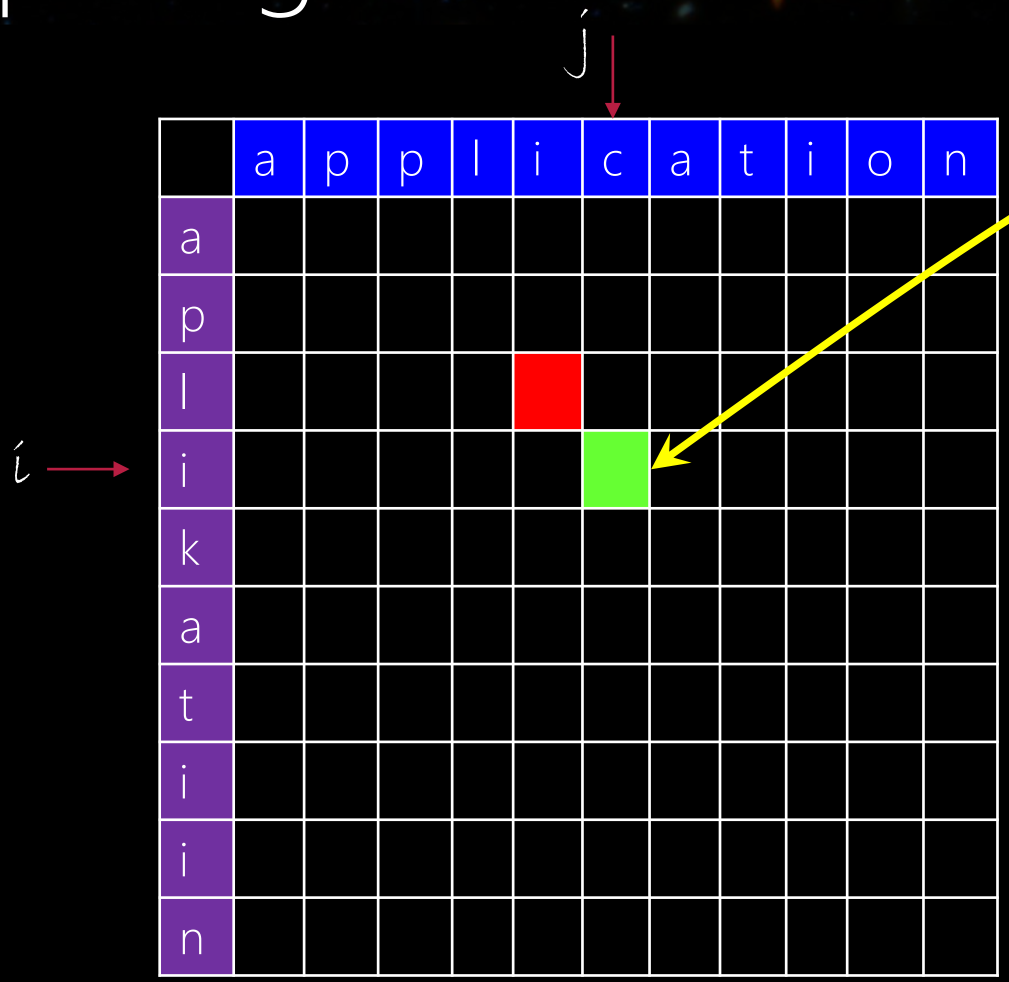
Spelling Correction

via

Minimum Edit Distance : Word Similarity in *Surface*

Text Normalization

Spelling Correction via Minimum Edit Distance



$$D(x[0:i], y[0:j]) = \min\{$$

$$D(x[0:i-1], y[0:j-1]) + x[i] \leftrightarrow y[j]$$

$$\}$$

$x[i] \neq y[j] \rightarrow 2$

$x[i] = y[j] \rightarrow 0$

Text Normalization

Spelling Correction via Minimum Edit Distance

	a	p	p	l	i	c	a	t	i	o	n
a											
p											
l											
i											
k											
a											
t											
i											
i											
n											

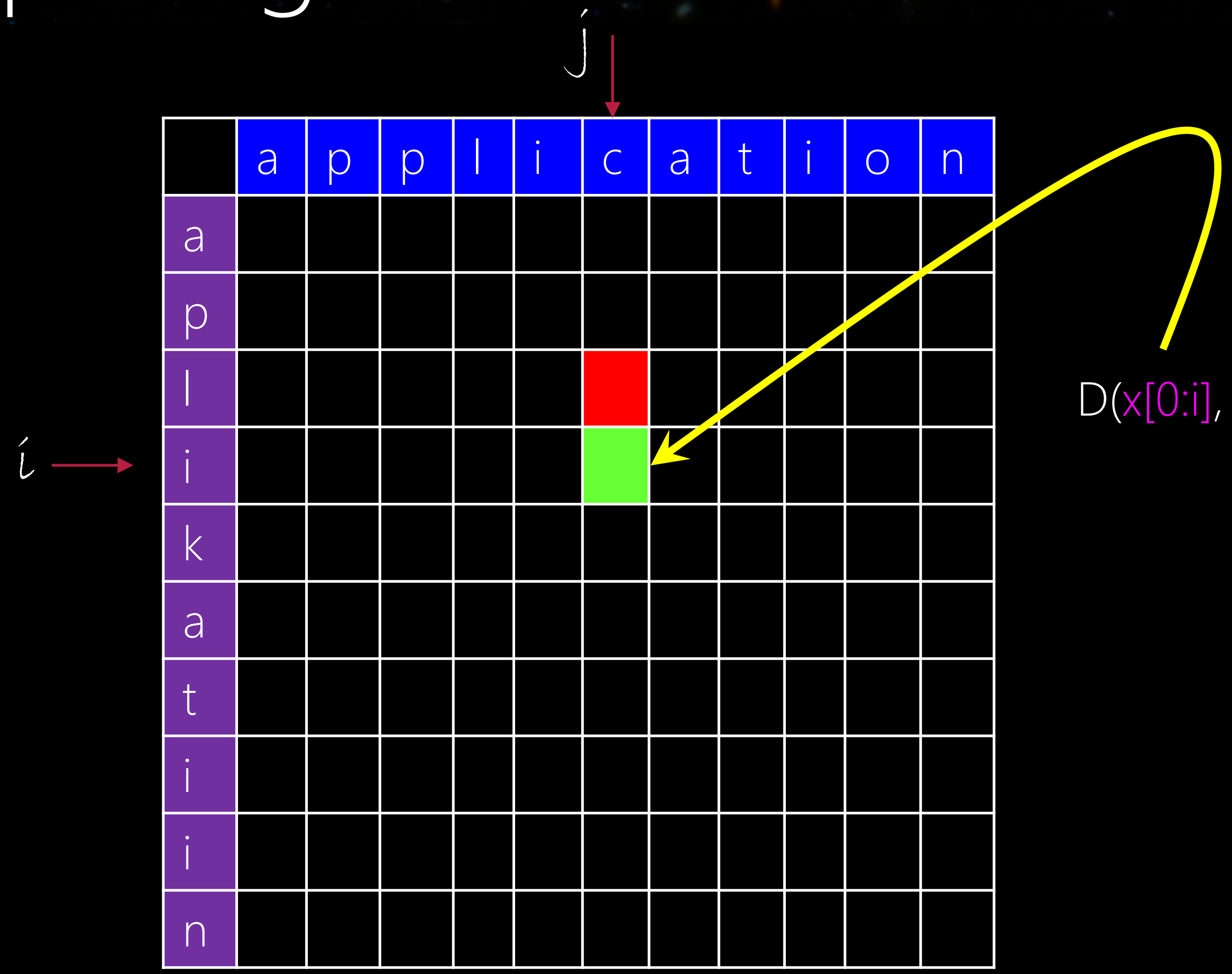
D(x[0:i], y[0:j]) = min{

D(x[0:i], y[0:j-1]) + 1 Insert

}

Text Normalization

Spelling Correction via Minimum Edit Distance

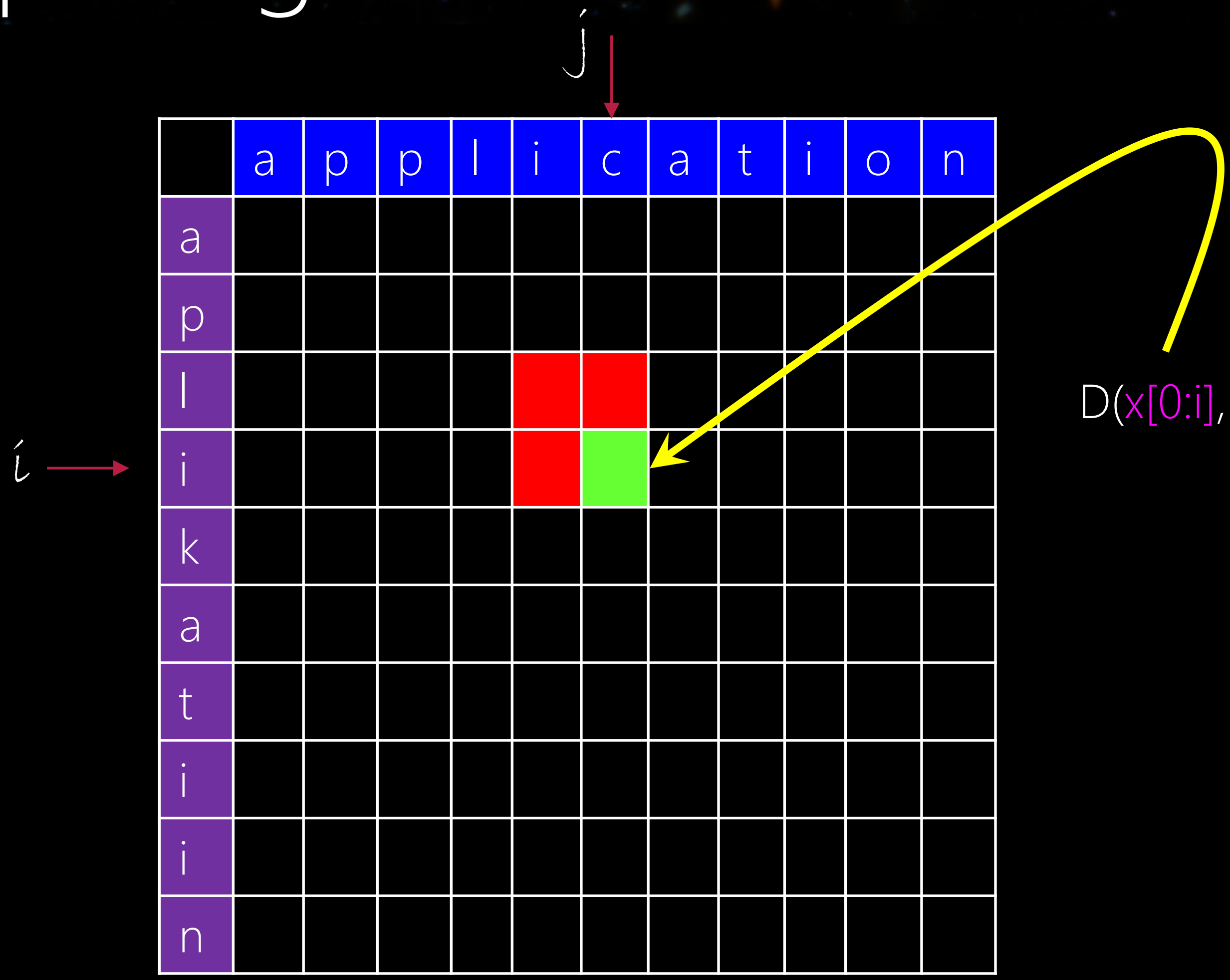


D(x[0:i], y[0:j]) = min{

D(x[0:i-1], y[0:j]) + 1 Delete
}

Text Normalization

Spelling Correction via Minimum Edit Distance



$$
\begin{aligned}
D(x[0{:}i], y[0{:}j]) = \min\{ & \\
& D(x[0{:}i-1], y[0{:}j-1]) + x[i] \leftrightarrow y[j] \\
& D(x[0{:}i], y[0{:}j-1]) + 1 \quad \text{Insert} \\
& D(x[0{:}i-1], y[0{:}j]) + 1 \quad \text{Delete} \\
& \}
\end{aligned}
$$

Text Normalization

Spelling Correction via Minimum Edit Distance

m

	a	p	p	l	i	c	a	t	i	o	n
a											
p											
l											
i											
k											
a											
t											
i											
i											
n											

n

$$D(x[0:i], y[0:j]) = \min\{$$
$$D(x[0:i-1], y[0:j-1]) + x[i] \leftrightarrow y[j]$$
$$D(x[0:i], y[0:j-1]) + 1 \quad \text{Insert}$$
$$D(x[0:i-1], y[0:j]) + 1 \quad \text{Delete}$$
$$\}$$

$$D(x[0:n], y[0:m])$$

Text Normalization

Spelling Correction via Minimum Edit Distance

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. Cybernetics and Control Theory (1965).

Pioneer in the theory of error correcting codes, Vladimir I. Levenshtein, known as the father of coding theory in Russia.

```
function MIN-EDIT-DISTANCE(source, target) returns min-distance

  n ← LENGTH(source)
  m ← LENGTH(target)
  Create a distance matrix distance[n+1,m+1]

  # Initialization: the zeroth row and column is the distance from the empty string
      D[0,0] = 0
      for each row i from 1 to n do
         D[i,0] ← D[i-1,0] + del-cost(source[i])
      for each column j from 1 to m do
         D[0,j] ← D[0, j-1] + ins-cost(target[j])

  # Recurrence relation:
  for each row i from 1 to n do
      for each column j from 1 to m do
         D[i, j] ← MIN( D[i−1,j] + del-cost(source[i]),
                        D[i−1,j−1] + sub-cost(source[i], target[j]),
                        D[i,j−1] + ins-cost(target[j]))
  # Termination
  return D[n,m]
```

Text Normalization

Spelling Correction via Minimum Edit Distance

https://phiresky.github.io/levenshtein-demo/

	a	p	p	l	i	c	a	t	i	o	n
a	0	1	2	3	4	5	6	7	8	9	10
p	1										
l	2										
i	3										
k	4										
a	5										
t	6										
i	7										
i	8										
n	9										

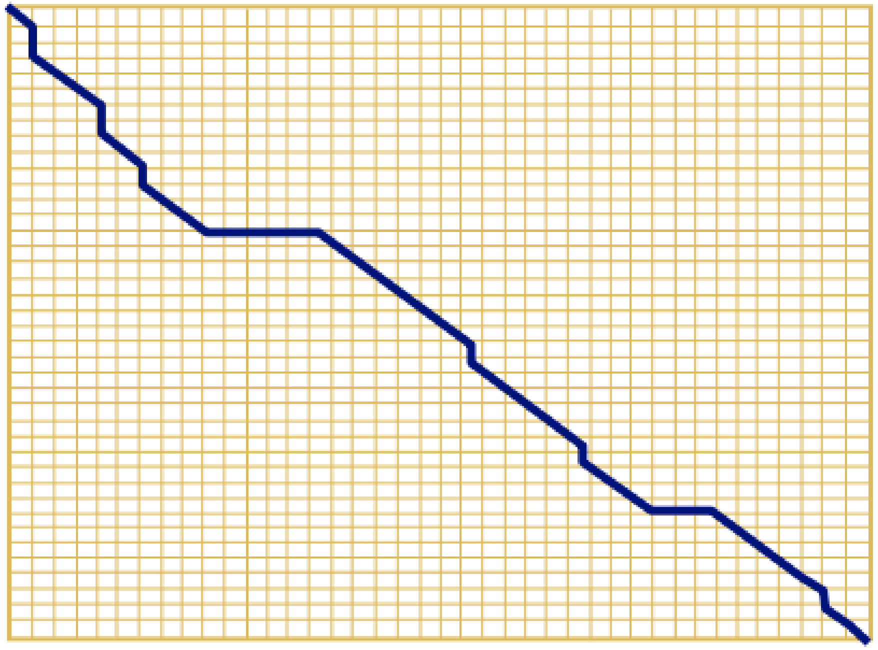
	a	p	p	l	i	c	a	t	i	o	n
a	0	1	2	3	4	5	6	7	8	9	10
p	1	0	1	2	3	4	5	6	7	8	9

	a	p
a	0	1
p	1	0
l	2	1
i	3	2
k	4	3
a	5	4
t	6	5
i	7	6
i	8	7
n	9	8

	a	p	p
a	0	1	2
p	1	0	1
l	2	1	2

Text Normalization

Spelling Correction via Minimum Edit Distance



from (0,0) to (M, N)

corresponds to
an alignment
of the two sequences

An optimal alignment is composed
of optimal subalignments

Dan Jurafsky

Lec02.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

Spelling Correction via Minimum Edit Distance

```python
# https://blog.paperspace.com/implementing-levenshtein-distance-word-autocomplete-autocorrect/
# Ahmed Fawzy Gad
# AI/ML engineer and a talented technical writer who authors 4 scientific books and more than 80 articles and tutorials. https://www.linkedin.com/in/ahmedfgad

import numpy
def levenshtein(token1, token2):
    distances = numpy.zeros((len(token1) + 1, len(token2) + 1))
    for t1 in range(len(token1) + 1): distances[t1][0] = t1
    for t2 in range(len(token2) + 1): distances[0][t2] = t2
    a = 0; b = 0; c = 0

    for t1 in range(1, len(token1) + 1):
        for t2 in range(1, len(token2) + 1):
              a = distances[t1][t2 - 1]
              b = distances[t1 - 1][t2]
              c = distances[t1 - 1][t2 - 1]

              if (a <= b and a <= c): distances[t1][t2] = a + 1
              elif (b <= a and b <= c): distances[t1][t2] = b + 1
              else:
                if (token1[t1 - 1] == token2[t2 - 1]): distances[t1][t2] = c
                else: distances[t1][t2] = c + 2

    print_matrix(distances, len(token1), len(token2))
    return distances[len(token1)][len(token2)]

def print_matrix(distances, token1Length, token2Length):
    for t1 in range(token1Length + 1):
        for t2 in range(token2Length + 1):
            print(str(int(distances[t1][t2])).zfill(2), end=" ")
        print()

levenshtein("application", "aplikatiion")
```

Text Normalization

Spelling Correction via Minimum Edit Distance

Levenshtein (1966)
Time Complexity: O($n \times m$)
Space Complexity: O($n \times m$)

Text Normalization

Learn to Correct Spellings

online texts (e.g., emails) depends on *keyboards*.

Misspells happens more on characters that sit next to each other on the keyboard.
Speed of typing is a source of error → Transposition: 'Desing' 'Design'

Text Normalization

Learn to Correct Spellings

Weighted Minimum Edit Distance

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

X \ Y (correct)	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
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	0	5	0	2	0	0	0	6	0	0	0	4	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

Text Normalization

Learn to Correct Spellings

Weighted Minimum Edit Distance

$$
D(x[0:i], y[0:j]) = \min\{
\begin{array}{l}
D(x[0:i-1], y[0:j-1]) + x[i] \leftrightarrow y[j] \\
D(x[0:i], y[0:j-1]) + \mathrm{insert}(y[j]) \\
D(x[0:i-1], y[0:j]) + \mathrm{delete}(x[i]) \\
\}
\end{array}
$$

$$x[i] \neq y[j] \rightarrow \mathrm{sub}(x[i], y[j])$$
$$x[i] = y[j] \rightarrow 0$$

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

X	Y (correct)																									
	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
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	0	5	0	2	0	0	0	6	0	0	0	4	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

Text Normalization

Learn to Correct Spellings

Weighted Minimum Edit Distance

Applications

Finding the closest word from Dictionary as the correct spell (Autocorrection)
Finding the closest word from Dictionary as the correct meaning!?
Finding the closest word from Dictionary as the prediction!? (Autocompletion)
Computational Biology

Aligning two sequences of protein
Daniel Jurafsky: https://www.youtube.com/watch?v=IL0-bD_e8s4