Some Thoughts on Integrating Acoustic Speech Data into Formal Ontologies and Knowledge Management and Representation Systems

Alireza Dehbozorgi

6 min read · Just now

Listen Share More

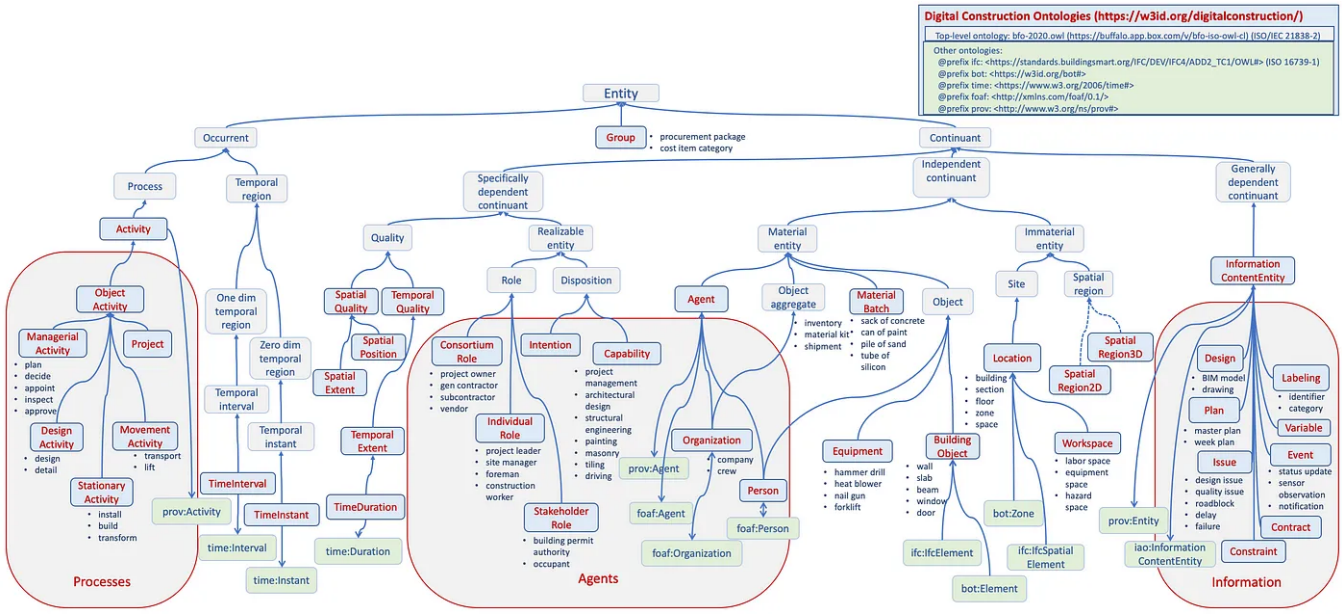


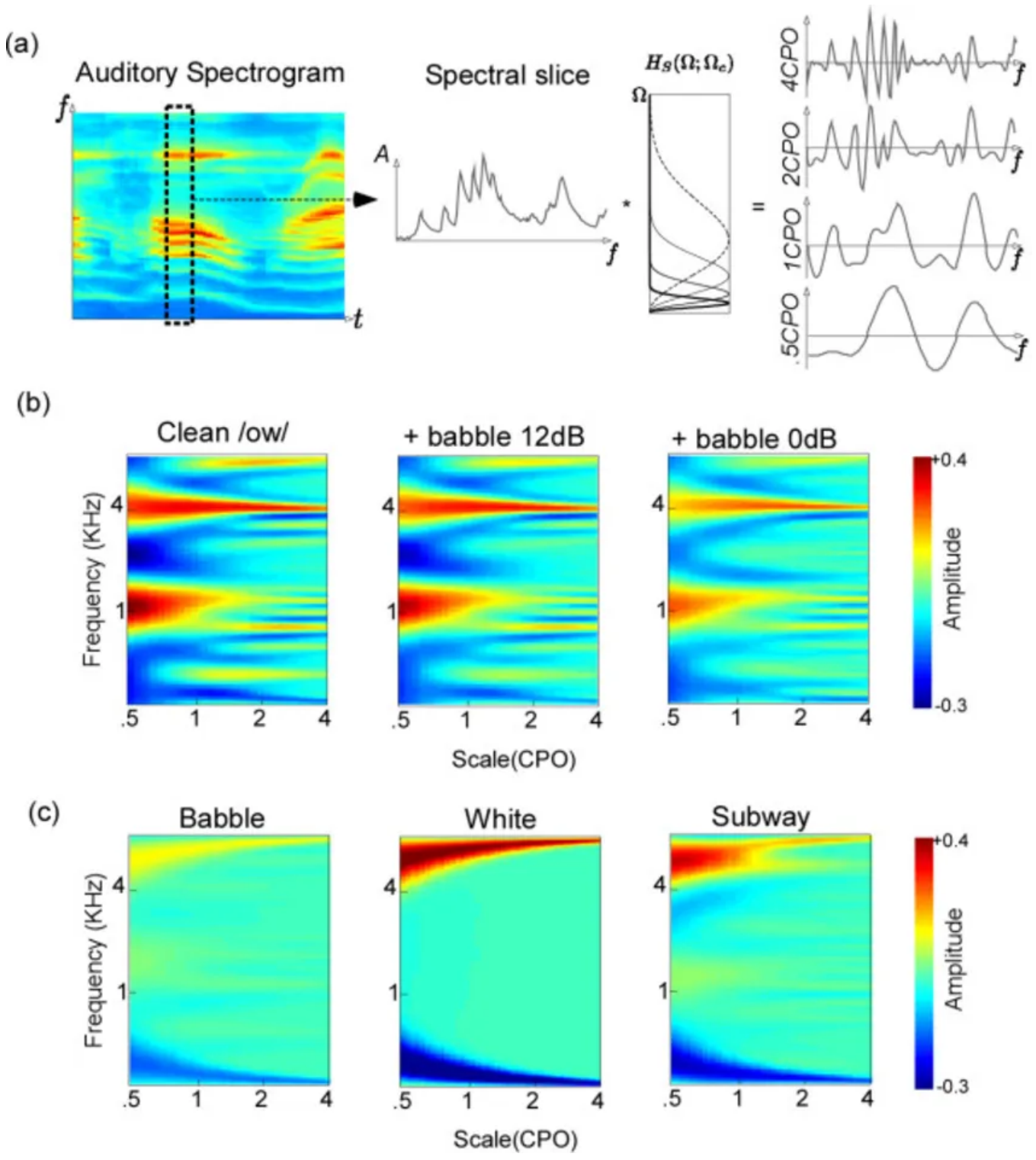
Image Source: https://digitalconstruction.github.io/v/0.3/index.html

Acoustic analysis can be used to integrate speech cues and data into formal ontologies (1) by identifying and extracting acoustic features from speech signals, which can then be mapped to ontology concepts and relationships. Here are some ways in which acoustic analysis can be applied:

1. Pitch analysis: Pitch analysis is a technique for analyzing the fundamental frequency of speech signals, which can be used to identify intonation and stress patterns in speech. This information can be used to map speech cues to ontology

concepts and relationships, such as emotional states or emphasis on particular concepts.

2. Spectral analysis: Spectral analysis is a technique for analyzing the frequency content of speech signals, which can be used to identify spectral features, such as formants and harmonics. These features can be used to map speech data to ontology concepts and relationships, such as phonemes and speech sounds.



https://www.researchgate.net/figure/Details-of-the-speech-spectral-analysis-a-The-speech-spectrogram-is-analyzed_fig2_257879534

3. Duration analysis: Duration analysis is a technique for analyzing the length of speech signals, which can be used to identify pauses and silences in speech. This information can be used to map speech cues to ontology concepts and relationships, such as turn-taking in conversation or the timing of events.

. . .

Integrating speech data into formal ontologies and knowledge management and representation systems can be a challenging task, but it can yield significant benefits for natural language processing applications. In this Medium blog, we will explore some of the techniques and tools that can be used to integrate speech data into formal ontologies and knowledge management systems. We will also provide some code snippets and references to help you get started.

1. Acoustic Analysis

Acoustic analysis is a technique for analyzing the acoustic features of speech signals, such as pitch, spectral content, and duration. These features can be used to identify phonemes, words, and phrases in speech, which can be mapped to ontology concepts and relationships. Some common acoustic analysis techniques include pitch analysis, spectral analysis, duration analysis, and mel-frequency cepstral coefficients (MFCCs).

Here is a concise sample code snippet in Python that demonstrates how to extract MFCCs from a speech signal using the librosa library:

```
import librosa

# Load speech signal
signal, sr = librosa.load('speech.wav', sr=None)

# Extract MFCCs
mfccs = librosa.feature.mfcc(signal, sr=sr, n_mfcc=13)
```

2. Named Entity Recognition

Named entity recognition (NER) is a technique for identifying and extracting named entities from text, such as people, places, and organizations. This technique can also be applied to speech data to identify named entities, which can be mapped to ontology concepts and instances.Here is a code snippet in Python that demonstrates how to perform NER on speech data using the Google Cloud Speech-to-Text API:

```
from google.cloud import speech_v1
from google.cloud.speech_v1 import enums

# Load speech file
with open('speech.wav', 'rb') as audio_file:
    content = audio_file.read()

# Initialize speech client
client = speech_v1.SpeechClient()

# Configure speech recognition request
config = speech_v1.types.RecognitionConfig(
    encoding=enums.RecognitionConfig.AudioEncoding.LINEAR16,
    sample_rate_hertz=16000,
    language_code='en-US',
    enable_word_time_offsets=True,
    enable_automatic_punctuation=True
)
audio = speech_v1.types.RecognitionAudio(content=content)

# Perform speech recognition
response = client.recognize(config=config, audio=audio)

# Extract named entities
for result in response.results:
    for word_info in result.alternatives[0].words:
        if word_info.speaker_tag == 1:
            print('Speaker {}:'.format(word_info.speaker_tag))
        print('Word: {}'.format(word_info.word))
        print('Entity: {}'.format(word_info.entity_tag))
```

3. Ontology Mapping

Once speech data has been analyzed and named entities have been identified, the next step is to map the extracted information to ontology concepts and relationships. This can be done using a variety of techniques, such as rule-based mapping, machine learning-based mapping, or a combination of both.

Here is an example of how ontology mapping can be done using the RDFLib library in Python:

```
from rdflib import Graph, Namespace, URIRef, Literal
from rdflib.namespace import RDF, RDFS, OWL, XSD

# Initialize ontology graph
g = Graph()

# Load ontology file
g.parse('ontology.owl')

# Define ontology namespaces
ns = Namespace('http://example.com/ontology#')
g.bind('ex', ns)

# Define speech data namespaces
speech_ns = Namespace('http://example.com/speech#')
g.bind('speech', speech_ns)

# Create speech data instance
speech_instance = URIRef('http://example.com/speech/speech1')
g.add((speech_instance, RDF.type, ns.Speech))

# Extract named entities from speech data
named_entities = ['John', 'New York', 'Google']

# Map named entities to ontology concepts
for named_entity in named_entities:
    if named_entity.lower() == 'john':
        g.add((speech_instance, ns.spokenBy, ns.John))
    elif named_entity.lower() == 'new york':
        g.add((speech_instance, ns.spokenIn, ns.NewYork))
    elif named_entity.lower() == 'google':
        g.add((speech_instance, ns.mentioned, ns.Google))

# Save ontology graph to file
g.serialize('speech_data.rdf',format='xml')
```

4. Knowledge Representation:

Once speech data has been mapped to ontology concepts and relationships, it can be represented and stored in a knowledge management system, such as a triple store or a semantic database. These systems allow for efficient querying and retrieval of

information, and can be used to power a variety of natural language processing applications.

Here is an example of how speech data can be stored in a triplestore using the SPARQL protocol:

```
from SPARQLWrapper import SPARQLWrapper, JSON

# Initialize SPARQL endpoint
sparql = SPARQLWrapper('http://example.com/sparql')

# Define SPARQL query
query = """
PREFIX ex: <http://example.com/ontology#>
PREFIX speech: <http://example.com/speech#>

INSERT DATA {
    speech:speech1 a ex:Speech ;
        ex:spokenBy ex:John ;
        ex:spokenIn ex:NewYork ;
        ex:mentioned ex:Google .
}
"""

# Execute SPARQL query
sparql.setQuery(query)
sparql.setMethod('POST')
sparql.query()

# Query the triplestore for speech data
query = """
PREFIX ex: <http://example.com/ontology#>
PREFIX speech: <http://example.com/speech#>

SELECT ?speaker ?location ?mention
WHERE {
    speech:speech1 ex:spokenBy ?speaker .
    speech:speech1 ex:spokenIn ?location .
   speech:speech1 ex:mentioned ?mention .
}
"""

sparql.setQuery(query)
sparql.setReturnFormat(JSON)
results = sparql.query().convert()

# Print results
for result in results['results']['bindings']:
    speaker = result['speaker']['value']
```

```
location = result['location']['value']
mention = result['mention']['value']
print('Speaker: {}'.format(speaker))
print('Location: {}'.format(location))
print('Mention: {}'.format(mention))
```

. . .

In conclusion, integrating speech data into formal ontologies and knowledge management systems requires a combination of techniques, such as acoustic analysis, named entity recognition, ontology mapping, and knowledge representation. By using these techniques, speech data can be transformed into structured data that can be mapped to ontology concepts and relationships, and stored in a knowledge management system for efficient querying and retrieval. The code snippets and references provided in this Medium blog should serve as a starting point for anyone interested in integrating speech data into formal ontologies and knowledge management systems.

. . .

References:

1. Huang, X., Acero, A., & Hon, H. W. (2001). Spoken language processing: A guide to theory, algorithm, and system development. Prentice Hall PTR.
2. Rabiner, L. R., & Juang, B. H. (1993). Fundamentals of speech recognition. Prentice Hall.
3. Stevens, K. N. (2000). Acoustic phonetics. MIT Press.
4. Stevens, K. N., & House, A. S. (2002). Acoustics and phonetics. In Handbook of the International Phonetic Association (pp. 97–100). Cambridge University Press.
5. Ladd, D. R. (2008). Intonational phonology (Vol. 32). Cambridge University Press.

6. Santiago, G., & Aquilar, J. (2022). Ontological model for the acoustic management in a smart environment. Applied Computing and Informatics (pp. 2634–1964)

7. Park, E. H., & Storey, V. C. (2023). Emotion Ontology Studies: A Framework for Expressing Feelings Digitally and its Application to Sentiment Analysis. ACM Computing Surveys, Volume 55, Issue 9. (pp 1–38), https://doi.org/10.1145/3555719

8. An artificially intelligent approach for automatic speech processing based on triune ontology and adaptive tribonacci deep neural networks (2022).

9. An Ontology-Based Approach for Knowledge Acquisition: An Example of Sustainable Supplier Selection Domain Corpus. (2022).

10. Ontology-Based Approach to Semantically Enhanced Question Answering for Closed Domain: A Review (2021).

11. Using Ontology-Based Approaches to Representing Speech Transcripts for Automated Speech Scoring (2013).

12. Using Ontology-based Approaches to Representing Speech Transcripts for Automated Speech Scoring (2012).

13. A Multi-level Acoustic Feature Extraction Framework for Transformer Based End-to-End Speech Recognition (2022).

14. A Survey on Databases for Multimodal Emotion Recognition and an Introduction to the VIRI (Visible and InfraRed Image) Database (2022).

15. Automatic Speech Recognition Using Limited Vocabulary: A Survey (2022).

16. RDFLib documentation: https://rdflib.readthedocs.io/en/stable/

17. Google Cloud Speech-to-Text API documentation: https://cloud.google.com/speech-to-text/docs

18. SPARQLWrapper documentation: https://rdflib.github.io/sparqlwrapper/

. . .

Stay tuned for more! Cheers!

Speech Recognition | Ontology | Acoustics | Knowledge Management | Knowledge Representation

Edit profile

Written by Alireza Dehbozorgi

56 Followers

I'm a linguist and AI researcher interested in mathematical/computational approaches to both human and formal languages. Twitter: @BDehbozorgi83

More from Alireza Dehbozorgi

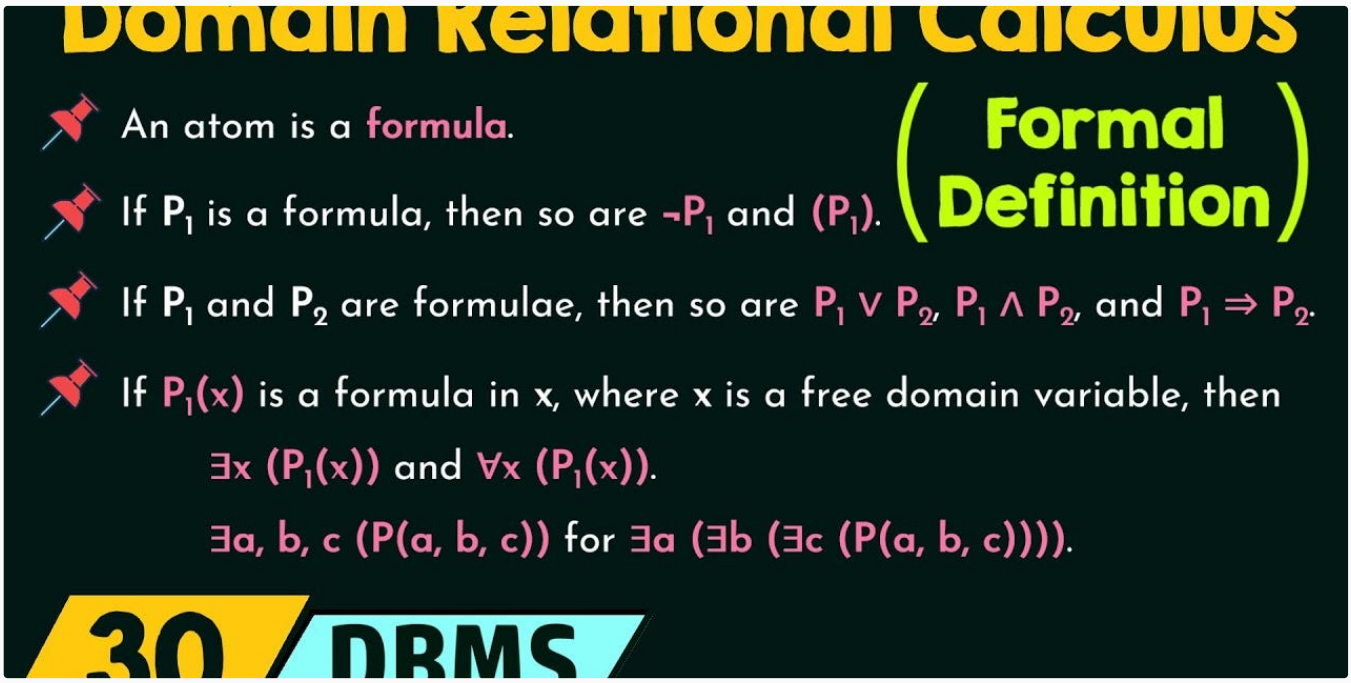
Alireza Dehbozorgi

Unraveling the Mystery of Prolog: Why it's More Relevant than Ever

Prolog, the Programming in Logic language, is a powerful tool for solving complex problems using a declarative approach. Prolog is a logic…

3 min read · 4 days ago

4 1



Alireza Dehbozorgi

Relational Calculus & Algebra (RCA) and Actionable Knowledge (Part I: Theoretical and Ethical…

Introduction

11 min read · May 11

1

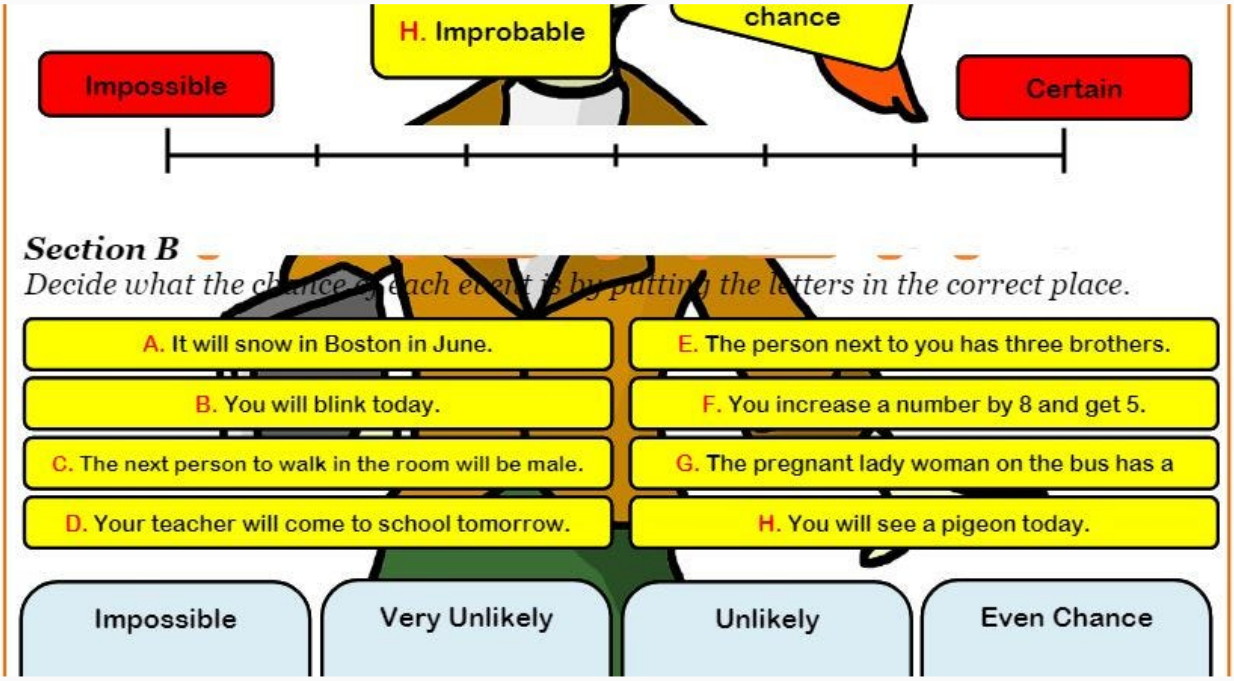
Alireza Dehbozorgi

Computational Theories of Cognition- Part II: Logic-Based Models of Cognition.

This article explains the approach to reaching the overarching scientific goal of capturing the cognition of persons in computational formal…

18 min read · May 11

2



Alireza Dehbozorgi

Language and Probability (Part I: Conceptual Ideas & Topics)

In recent years, a strong consensus has emerged that human cognition is based on probabilistic processing. (see Griffith (2011)) for a…

10 min read · Sep 21, 2022

15

See all from Alireza Dehbozorgi

Recommended from Medium

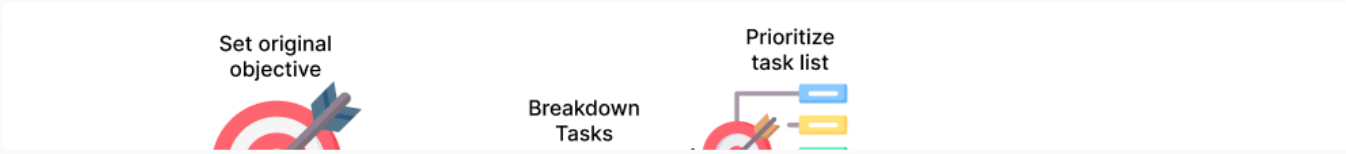
Wei-Meng Lee in Level Up Coding

Training Your Own LLM using privateGPT

Learn how to train your own language model without exposing your private data to the provider

· 7 min read · 4 days ago

448 4



Guodong (Troy) Zhao in Bootcamp

A comprehensive and hands-on guide to autonomous agents with GPT

Understand autonomous agents with LLM, from conceptual guide to hands-on tutorial

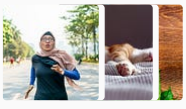
· 15 min read · May 2

498 4

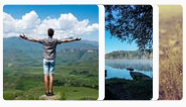
Lists

Staff Picks
311 stories · 79 saves

Stories to Help You Level-Up at Work
19 stories · 36 saves



Self-Improvement 101
20 stories · 81 saves



Productivity 101
20 stories · 80 saves

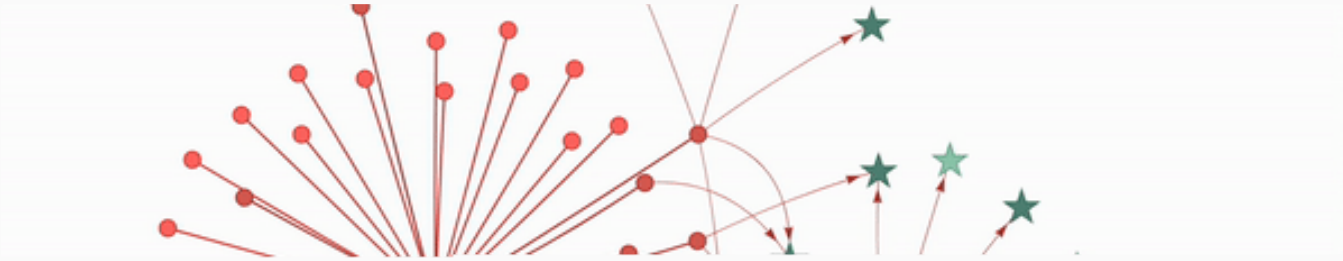
Edward Ma

Speaker Recognition from Audio

Moving from Text to Audio

· 7 min read · Dec 19, 2022

11 1



Anastasiia

Network Graph Visualisation

Exploring and Understanding Polygon Transactions using NetworkX and Pyvis

7 min read · Jan 15

47

PCMag in PC Magazine

What Are ChatGPT Plugins? The Next Phase of Conversational AI Is Here

ChatGPT Plus users can now access 70+ plugins made by individual creators and companies including Instacart and Kayak. Here's how they work…

9 min read · 6 days ago

137 3