MACHINE LEARNING IN NATURAL LANGUAGE PROCESSING

[redacted]a; Grupa 341

ABSTRACT

Two concepts, one mission: to make machines understand humans. Natural Language Processing (NLP) and Machine Learning (ML) are all the rage right now, but people tend to mix them up. In this paper, I will present the distinction between these two different, but complementary terms in the field of Artificial Intelligence and how to apply machine learning to solve problems in natural language processing and text analytics.

INTRODUCERE

Domeniul Inteligenței Artificiale reprezintă, fără doar și poate, viitorul, iar frenezia, cu care este înconjurat, la momentul actual, este mai mult decât bine meritată, raportându-ne nu doar la descoperirile din ultimii ani, dar și la potențialul încă neexplorat al acestei ramuri informatice. Dacă până de curând, **AI** (eng. original: *Artificial Intelligence*) reprezenta tematica filmelor Science Fiction și părea un subiect futurist, care nu ar fi putut să se dezvolte în secolul nostru, acum vorbim de o realitate tangibilă, care a luat amploare în anii recenți.

Împreună cu acest concept, 3 alte terminologii continuă să se evidențieze în discuțiile de specialitate și anume: **ML** (eng. original: *Machine Learning*), **DL** (eng. original: *Deep Learning*) și **NLP** (eng. original: *Natural Language Processing*). Oamenii au tendința de a confunda acești 4 termeni și este ușor să înțelegem de ce dacă ne gândim cât de interconectate aceste domenii sunt în realitate.

Pentru a putea să discutăm despre subiectul ales, trebuie să începem cu înțelegerea conceptelor de bază, care sunt diferențele dintre ele, dar și ce relații există între acestea, pentru a ne forma un punct de plecare în analizarea problemei puse în discuție.

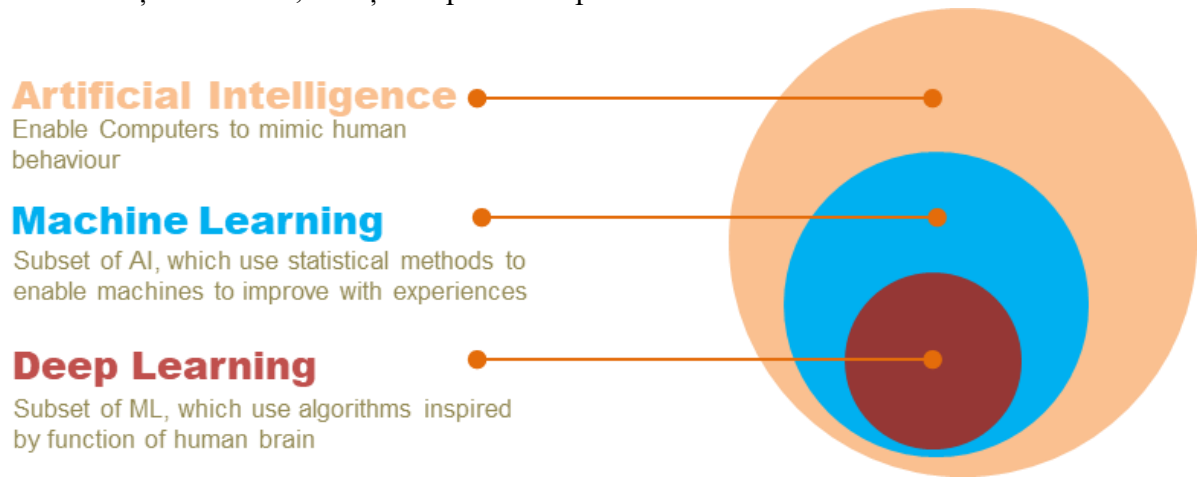
AI vs ML vs DL: SCURT ISTORIC, DIFERENȚE ȘI RELAȚII

Domeniului **AI** își are originile încă din Antichitate, când filosofii clasici încercau să descrie gândirea umană printr-un sistem simbolic. Oficial, această subramură a informaticii a fost fondată în **1956**, la o conferință a Universității Dartmouth, din New Hampshire, SUA, când termenul de „*inteligență artificială*" a fost inventat și folosit pentru prima oară. **Inteligența Artificială** reprezintă procesul de încorporare a inteligenței umane în mașini sau sisteme informatice, astfel încât acestea să poată dezvolta capacitatea de a gândi și de a răspunde ca oamenii. ML, DL și NLP sunt toate subcâmpuri ale acestui domeniu. Majoritatea exemplelor de AI despre care auzim în zilele noastre, de la computere care joacă șah, până la mașini care se conduc singure, toate se bazează pe învățare adâncă (DL) și procesarea limbajului natural (NLP). Folosind aceste tehnologii, computerele pot fi instruite pentru a îndeplini sarcini specifice prin procesarea unor cantități mari de date și recunoașterea tiparelor care apar. Pe scurt, scopul principal al inteligenței artificiale este de a dezvolta mașini care să poată lua decizii inteligente pe baza experiențelor din trecut și a propriilor abilități de învățare și să transmită soluția oamenilor.

Termenul de „*machine learning*" a fost inventat în **1959** de Arthur Samuel, pionier în domeniul inteligenței artificiale, dar și a jocurilor video. **Învățarea Automată** este o ramură a inteligenței artificiale, bazată pe idea că sistemele informatice pot învăța prin utilizarea observațiilor (date, de exemplu) sau a experienței anterioare și pot lua decizii cu o intervenție umană minimă. Modelele produse folosind această tehnică au capacitatea de a oferi previziuni viitoare și de a veni cu soluții inteligente.

În **1986**, Rina Dechter a introdus termenul de „*deep learning*". **Învățarea profundă** este un tip de învățare automată, care antrenează un computer pentru a îndeplini sarcini similare omului, cum ar fi recunoașterea vorbirii, identificarea imaginilor sau efectuarea de predicții. Algoritmii de DL sunt inspirați de metodele de procesare a informațiilor folosite de creirul uman. În loc să organizeze datele de intrare pentru a rula pe ecuații predefinite, învățarea profundă stabilește parametrii de bază pentru input și antrenează computerul să învețe singur prin recunoașterea modelelor folosite.

Relația dintre AI, ML și DL poate fi reprezentată astfel:

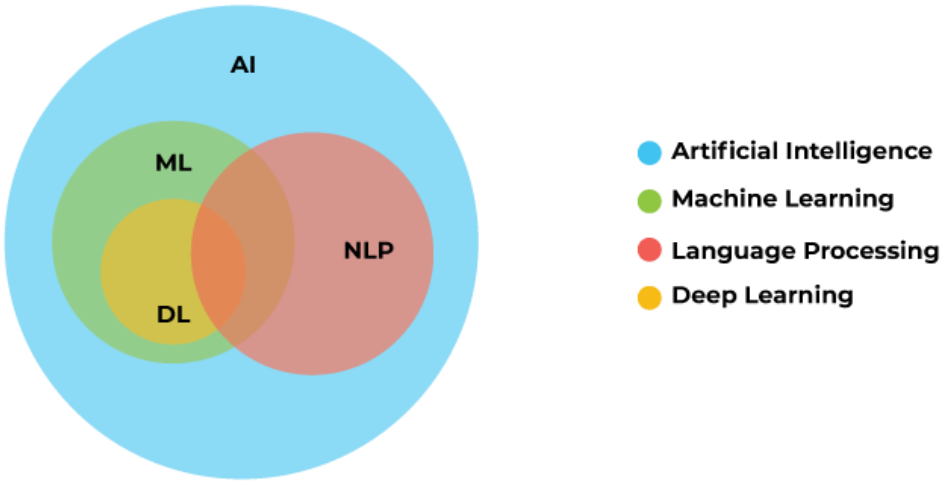


~ Figură 1 – AI vs ML vs DL (Banerjee, 2020) ~

NLP: CE ESTE ȘI UNDE SE ÎNCADREAZĂ ÎN LUMEA AI?

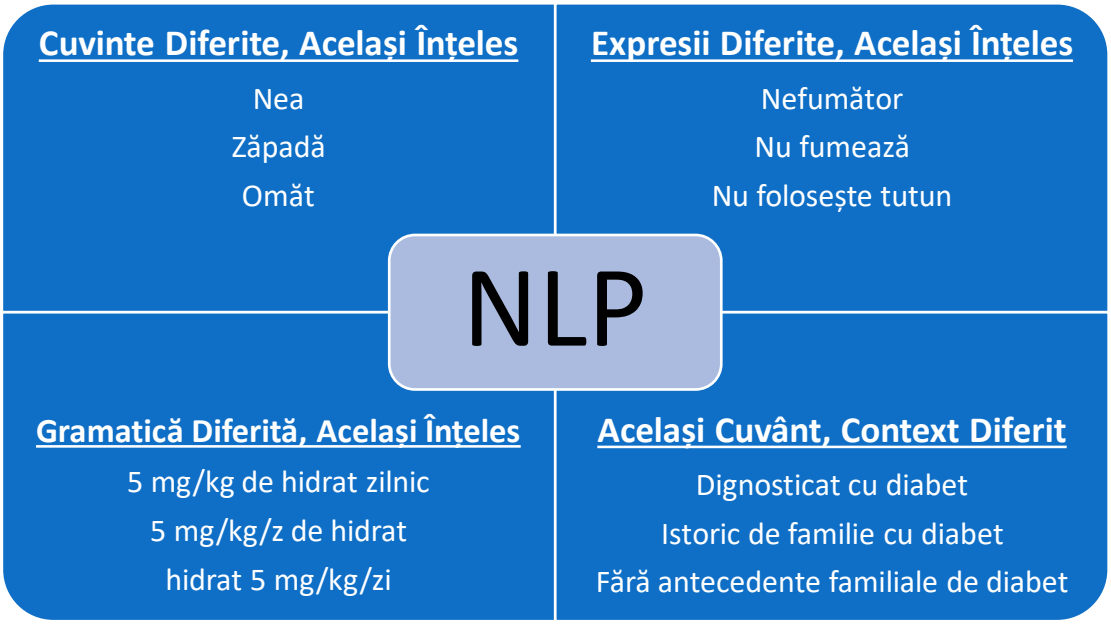
Roberts Dilts este recunoscut ca fiind cel care a inventat termenul de „*natural language processing*”, în anul **1981**. **Procesarea limbajului natural** este procesul de explicare a unei structuri sau comenzi către o mașină, folosind limbajul natural, al oamenilor. Acest limbaj va fi tradus într-un format pe care sistemul informatic să îl poată înțelege, procesa și, ulterior, genera înapoi către utilizator. Pe scurt, NLP este o ramură a inteligenței artificiale care ajută computerele să priceapă, să interpreteze și să manipuleze limbajul uman.

Pentru a evidenția mai clar relația dintre NLP și conceptele explicate anterior, prezentăm următoarea ilustrație:



~ Figură 2 - NLP vs AI (Bolaños, 2020) ~

NLP se străduiește să reducă diferența dintre mașini și oameni, permițând unui computer să analizeze ce a spus utilizatorul și să prelucreze ce a vrut să exprime acesta. După cum ne așteptăm, această sarcină s-a dovedit destul de complexă. Pentru a putea purta o conversație cu un om, sistemul informatic trebuie să înțeleagă sintaxa (gramatică), semantica (înțelesul cuvintelor), morfologia și pragmatica. Numărul de reguli care trebuie urmărite poate părea copleșitor și explică de ce încercările anterioare de NLP au dus, inițial, la rezultate dezamăgitoare.



ML în NLP

Acum că am explicat ce reprezintă fiecare concept și avem o bază de informații, putem să intuim deja de ce ar trebui să folosim învățarea automată în domeniul NLP și care sunt avantajele unei astfel de abordări. NLP, după cum spune și numele, procesează limbajul și ne oferă o analiză a textului, dar pentru a putea învăța computerul să „vorbească" cu noi, folosim ML. Din experiența anterioară și exemplele oferite de noi, sistemul informatic se va îmbunătăți și va fi capabil să înțeleagă și să folosească limbaj uman.

Când se vorbește despre analiza textului, **învățarea automată** este considerată o combinație de tehnici statistice care servește la detectarea tiparelor, sentimentelor din spatele cuvintelor (furie, ironie, fericire etc...), părților de vorbire și altor fenomene dintr-un text. Există două tipuri de proceduri de învățare automată: **supravegheată** și **nesupravegheată**. *Învățarea automată supravegheată* este procesul în care tehnicile ML pot fi exprimate ca un model care poate fi aplicat și altor texte. Există, de asemenea, câțiva algoritmi care lucrează în seturi de date extinse pentru a obține o semnificație, un înțeles general, cunoscut și sub numele de *învățare automată nesupravegheată*. Cunoașterea diferențelor dintre aceste 2 tehnici este esențială pentru a obține cele mai bune rezultate în practică:

1. ***Învățarea Supravegheată***: În acest tip de învățare automată, o serie de documente text sunt etichetate sau adnotate cu exemple referitoare la ce ar trebui să caute mașina și cum ar trebui să interpreteze acel aspect. Aceste documente sunt utilizate pentru a antrena un model statistic, căruia i se oferă apoi un text neetichetat pentru a fi analizat. Cu cât setul de date este mai mare, cu atât rezultatele sunt mai bune: fiecare model poate fi instruit de mai multe ori pentru a-și îmbunătăți învățarea. Cei mai populari algoritmi din această categorie sunt: Support Vector Machines, Bayesian Networks, Maximum Entropy, Conditional Random Field, Neural Networks/Deep Learning.
2. ***Învățarea Nesupravegheată***: Acest tip implică instruirea unui model fără etichetare prealabilă sau adnotare. Unele dintre aceste tehnici sunt surprinzător de ușor de înțeles:
    a. *Clustering*: înseamnă gruparea documentelor similare în grupuri sau seturi. Aceste clustere sunt apoi sortate în funcție de importanță și relevanță (grupare ierarhică).
    b. *Latent Semantic Indexing (LSI)*: e folosită pentru a obține acele cuvinte care apar adesea împreună în texte. Oamenii de știință utilizează LSI pentru căutări cu mai multe fațete sau pentru returnarea rezultatelor căutării care nu sunt termenul exact de căutare. Ex: dacă cuvintele „TV" și „canal" sunt corelate în multe texte, atunci veți primi foarte probabil documente care conțin „canal", chiar dacă doar căutați „TV".
    c. *Matrix Factorization*: Această tehnică folosește „factori latenți" pentru a descompune o matrice mare în două matrici mai mici. Factorii latenți sunt asemănări între elemente. Ex: „Am aruncat mingea peste munte". Cuvântul „aruncat" este asociat, mult mai probabil, cu „minge" decât cu „munte".

UN EXEMPLU CONCRET

NLP are o mulțime de utilizări practice în zilele noastre, dintre care amintim:

- Information Retrieval (Google finds relevant and similar results).
- Information Extraction (Gmail structures events from emails).
- Machine Translation (Google Translate translates language from one language to another).
- Text Simplification (Rewordify simplifies the meaning of sentences)
- Sentiment Analysis (Hater News gives us the sentiment of the user).
- Text Summarization (Smmry or Reddit's autotldr gives a summary of sentences).
- Spam Filter (Gmail filters spam emails separately).
- Auto-Predict (Google Search predicts user search results).
- Auto-Correc t(Google Keyboard and Grammarly correct words otherwise spelled wrong).
- Speech Recognition (Google WebSpeech or Vocalware).
- Question Answering (IBM Watson's answers to a query).
- Natural Language Generation (Generation of text from image or video data.)

În acest exemplu, vom construi un model pentru a înțelege recenziile vinurilor (în limbaj natural) de către experți și pentru a deduce varietatea vinului pe care îl examinează. Pentru problema noastră, vom folosi acest dataset: https://www.kaggle.com/zynicide/wine-reviews.

Problema cu care lucrăm astăzi este, în esență, o problemă de clasificare NLP. Există mai mulți algoritmi de clasificare NLP care au fost aplicați diferitelor probleme. De exemplu, Bayes naiv a fost folosit pentru detectarea spamului, iar SVM a fost folosită pentru a clasifica texte. Ar fi interesant să implementăm o versiune simplă a acestor algoritmi pentru a servi ca bază a modelului nostru de învățare profundă.

O implementare populară a Bayes naiv pentru NLP implică preprocesarea textului folosind TF-IDF și apoi rularea modelului pe ieșirile preprocesate. Aceasta permite algoritmului să fie rulat pe cele mai proeminente cuvinte dintr-un document. Putem implementa naiv Bayes după cum urmează:

```
import numpy as np
from sklearn.naive_bayes import MultinomialNB
from sklearn.feature_extraction.text import CountVectorizer
import pandas as pd
from collections import Counter
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfTransformer

df = pd.read_csv('data/wine_data.csv')

counter = Counter(df['variety'].tolist())
top_10_varieties = {i[0]: idx for idx, i in
enumerate(counter.most_common(10))}
df = df[df['variety'].map(lambda x: x in top_10_varieties)]
```

```
description_list = df['description'].tolist()
varietal_list = [top_10_varieties[i] for i in df['variety'].tolist()]
varietal_list = np.array(varietal_list)

count_vect = CountVectorizer()
x_train_counts = count_vect.fit_transform(description_list)


tfidf_transformer = TfidfTransformer()
x_train_tfidf = tfidf_transformer.fit_transform(x_train_counts)

train_x, test_x, train_y, test_y = train_test_split(x_train_tfidf,
varietal_list, test_size=0.3)

clf = MultinomialNB().fit(train_x, train_y)
y_score = clf.predict(test_x)

n_right = 0
for i in range(len(y_score)):
    if y_score[i] == test_y[i]:
        n_right += 1

print("Accuracy: %.2f%%" % ((n_right/float(len(test_y)) * 100)))
```

Și obținem acuratețe de 73.56%. Dacă folosim `clf = SVC(kernel='linear').fit(train_x, train_y)` acuratețea ajunge la 80.66%. Nu e foarte rău.

Provocarea noastră e să construim un model de învățare profundă care poate depăși sau cel puțin să egaleze aceste rezultate. Dacă reușim acest lucru, ar fi un indiciu excelent că modelul nostru de DL este eficient în cel puțin replicarea rezultatelor modelelor populare de învățare automată. Pentru acest lucru, vom folosi Keras cu Tenserflow pentru a ne crea modelul.

În primul rând, va trebui să restructurăm datele într-un mod care poate fi ușor procesat și înțeles de computer. Putem face acest lucru înlocuind cuvintele cu numere de identificare unice. Prepocesarea arată ceva de genul:

```
from nltk import word_tokenize
from collections import defaultdict

def count_top_x_words(corpus, top_x, skip_top_n):
    count = defaultdict(lambda: 0)
    for c in corpus:
        for w in word_tokenize(c):
            count[w] += 1
    count_tuples = sorted([(w, c) for w, c in count.items()], key=lambda x:
x[1], reverse=True)
    return [i[0] for i in count_tuples[skip_top_n: skip_top_n + top_x]]


def replace_top_x_words_with_vectors(corpus, top_x):
    topx_dict = {top_x[i]: i for i in range(len(top_x))}

    return [
        [topx_dict[w] for w in word_tokenize(s) if w in topx_dict]
        for s in corpus
```

```
    ], topx_dict


def filter_to_top_x(corpus, n_top, skip_n_top=0):
    top_x = count_top_x_words(corpus, n_top, skip_n_top)
    return replace_top_x_words_with_vectors(corpus, top_x)
```

Acum suntem gata să construim modelul. Vrem un strat de încorporare, un strat convoluțional și un strat dens pentru a profita de toate caracteristicile de învățare profundă care pot fi utile pentru aplicația noastră. Cu Keras, putem construi modelul foarte simplu, deoarece toate aceste lucruri ne sunt oferite la dispoziție:

```
from keras.models import Sequential
from keras.layers import Dense, Conv1D, Flatten
from keras.layers.embeddings import Embedding
from keras.preprocessing import sequence
from keras.utils import to_categorical
import pandas as pd
from collections import Counter
from sklearn.model_selection import train_test_split
from lib.get_top_xwords import filter_to_top_x

df = pd.read_csv('data/wine_data.csv')

counter = Counter(df['variety'].tolist())
top_10_varieties = {i[0]: idx for idx, i in
enumerate(counter.most_common(10))}
df = df[df['variety'].map(lambda x: x in top_10_varieties)]

description_list = df['description'].tolist()
mapped_list, word_list = filter_to_top_x(description_list, 2500, 10)
varietal_list_o = [top_10_varieties[i] for i in df['variety'].tolist()]
varietal_list = to_categorical(varietal_list_o)

max_review_length = 150

mapped_list = sequence.pad_sequences(mapped_list, maxlen=max_review_length)
train_x, test_x, train_y, test_y = train_test_split(mapped_list,
varietal_list, test_size=0.3)

max_review_length = 150

embedding_vector_length = 64
model = Sequential()

model.add(Embedding(2500, embedding_vector_length,
input_length=max_review_length))
model.add(Conv1D(50, 5))
model.add(Flatten())
model.add(Dense(100, activation='relu'))
model.add(Dense(max(varietal_list_o) + 1, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=['accuracy'])
model.fit(train_x, train_y, epochs=3, batch_size=64)

y_score = model.predict(test_x)
```

```
y_score = [[1 if i == max(sc) else 0 for i in sc] for sc in y_score]
n_right = 0
for i in range(len(y_score)):
    if all(y_score[i][j] == test_y[i][j] for j in range(len(y_score[i]))):
        n_right += 1

print("Accuracy: %.2f%%" % ((n_right/float(len(test_y)) * 100)))
```

Acum obținem acuratețea de 77.20%. Comparativ cu rezultatele anterioare, acesta este un scor foarte bun.

CONCLUZII

După cum am văzut, NLP oferă un set larg de tehnici și instrumente care pot fi aplicate în toate domeniile vieții. Învățându-le și folosindu-le în interacțiunile noastre de zi cu zi, calitatea vieții noastre s-ar îmbunătăți foarte mult.

Totul este mult mai rapid și mai eficient, deoarece acum putem comunica cu mașinile, datorită tehnologiei de procesare a limbajului natural. NLP a oferit companiilor majore capacitatea de a fi flexibile în ceea ce privește deciziile lor, datorită perspectivelor pe care le-a deschis în aria de satisfacție a clienților, dar și în anticiparea schimbărilor de piață. Organizațiile inteligente iau acum decizii bazate nu numai pe date, ci și pe inteligența derivată din previziunile sistemelor informatice care folosesc această tehnologie.

Dacă există un lucru pe care îl putem garanta că se va întâmpla în viitor, este integrarea procesării limbajului natural în aproape fiecare aspect al vieții așa cum o cunoaștem. Ultimii cinci ani au fost un „*slow burn*” a ceea ce poate face NLP, grație integrării pe toate tipurile de dispozitive, de la computere și frigidere, la difuzoare și automobile. Oamenii, de asemenea, au arătat mai mult entuziasm decât antipatie pentru procesul de interacțiune om-mașină.

NLP is no longer the future. It’s already here! And it will only grow.

BIBLIOGRAFIE

Lee, A. (2019, November 24). Why NLP is important and it'll be the future — our future. Retrieved May 4, 2021, from Medium website: https://towardsdatascience.com/why-nlp-is-important-and-itll-be-the-future-our-future-59d7b1600dda

Starbridge Partners. (2020, January 22). The Future of Natural Language Processing - Starbridge Partners. Retrieved May 4, 2021, from Starbridge Partners website: http://starbridgepartners.com/2020/01/the-future-of-natural-language-processing/

Ximena Bolaños. (2020, December 9). Natural Language Processing and Machine Learning. Retrieved May 4, 2021, from Encora website: https://www.encora.com/insights/natural-language-processing-and-machine-learning

Natural Language Processing (NLP) Simplified : A Step-by-step Guide. (2021). Retrieved May 4, 2021, from Datascience.foundation website: https://datascience.foundation/sciencewhitepaper/natural-language-processing-nlp-simplified-a-step-by-step-guide

Pistoia Alliance. (2019). NLP & ML Webinar. Retrieved May 4, 2021, from Slideshare.net website: https://www.slideshare.net/pistoiaalliance/nlp-ml-webinar

Bitext. (2019). Natural Language Processing (NLP) vs. Machine Learning. Retrieved May 4, 2021, from Bitext.com website: https://blog.bitext.com/natural-language-processing-vs-machine-learning#ML

NLP, AI, and Machine Learning: What's The Difference? (2020, June 9). Retrieved May 4, 2021, from MonkeyLearn Blog website: https://monkeylearn.com/blog/nlp-ai/

Machine Learning (ML) for Natural Language Processing (NLP) - Lexalytics. (2020, September 29). Retrieved May 4, 2021, from Lexalytics website: https://www.lexalytics.com/lexablog/machine-learning-natural-language-processing#ml-vs-nlp

Five AI Technologies. (2021). Retrieved May 4, 2021, from Sas.com website: https://www.sas.com/en_us/insights/articles/analytics/five-ai-technologies.html

Artificial Intelligence – What it is and why it matters. (2021). Retrieved May 4, 2021, from Sas.com website: https://www.sas.com/en_us/insights/analytics/what-is-artificial-intelligence.html

Hira Saeed. (2016, November 4). Developing a Chatbot? Learn the Difference between AI, Machine Learning, and NLP. Retrieved May 4, 2021, from Medium website: https://chatbotslife.com/developing-a-chatbot-learn-the-difference-between-ai-machine-learning-and-nlp-40a3f745aec4

Badreesh Shetty. (2018, November 24). Natural Language Processing(NLP) for Machine Learning. Retrieved May 4, 2021, from Medium website: https://towardsdatascience.com/natural-language-processing-nlp-for-machine-learning-d44498845d5b