

Importing the Required Libraries

```
!pip install matplotlib
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (3.7.1)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.24.3)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (0.11.0)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (8.4.0)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (4.39.3)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (3.0.9)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.4)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (23.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.0.7)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
```

```
!pip install --upgrade pandas
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (2.0.1)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2022.7.1)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist-packages (from pandas) (1.24.3)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2023.3)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
```

```
import json
import glob
import nltk
import spacy
import gensim
import gensim.corpora as corpora
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from gensim.utils import simple_preprocess
from gensim.models import CoherenceModel
from nltk.corpus import stopwords
from wordcloud import WordCloud
```

Preparing the Data

```
df = pd.read_excel('/content/drive/MyDrive/Project Data.xls')
print(df.head(5))
```

```

      ConversationId \
0  AAQkAGVjMmUyMDljLTAYMzYtNGU0MC1iOGFmLWI3NWQzM2...
1  AAQkAGVjMmUyMDljLTAYMzYtNGU0MC1iOGFmLWI3NWQzM2...
2  AAQkAGVjMmUyMDljLTAYMzYtNGU0MC1iOGFmLWI3NWQzM2...
3  AAQkAGVjMmUyMDljLTAYMzYtNGU0MC1iOGFmLWI3NWQzM2...
4  AAQkAGVjMmUyMDljLTAYMzYtNGU0MC1iOGFmLWI3NWQzM2...

      Subject \
0      5130 Browns Pt Blvd Unit D - Landscaping
1  Fwd: 5130 Browns Pt Blvd Unit D - Landscaping
2      RE: 5130 Browns Pt Blvd Unit D - Landscaping
3  File Number-3645595-Address-12526 SE 32nd St U...
4  RE: File Number-3645595-Address-12526 SE 32nd ...

      Body      Category \
0  Dear Propvivo, \r\nThank you for the recent le...  Maintenance
1  Sorry , here are the pictures. [cid:0DD370C3-6...  Maintenance
2  Dear Russ & Shanna,\r\n\r\nWe Thank You for yo...  Maintenance
3  \r\n\r\n\r\n\r\n\r\n\r\nFile No.: 4244 3645595\r\n...  Other
4  Hi,\r\n\r\n\r\nCan you please confirm if this a Ru...  Payment

      HasAttachment  DateTimeReceived  UnitNumber
0      False  2019-10-09 20:02:06  Unit # 5130D
1      False  2019-10-09 20:10:03  Unit # 5130D
2      False  2019-10-09 22:31:00  Unit # 5130D
3      True  2021-01-29 05:56:13  NaN
4      False  2021-01-29 21:19:00  NaN
```

```

def load_data(link):
    df = pd.read_excel(link)
    return df

nltk.download('stopwords')
stopwords = stopwords.words("english")

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!

print (stopwords)

['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself',
<img alt="Horizontal scrollbar" data-bbox="100 225 950 238"/>

data = load_data('/content/drive/MyDrive/Project Data.xls')["Body"]

#cleaning the data
# Remove empty strings
data = data.replace('', pd.NA)

# Drop rows with null values
data = data.dropna()

print (data[0][0:90])

Dear Propvivo,
Thank you for the recent letter regarding the dog poop remains in the way

def lemmatization(texts, allowed_postags=["NOUN", "ADJ", "VERB", "ADV"]):
    nlp = spacy.load("en_core_web_sm", disable=["parser", "ner"])
    texts_out = []
    for text in texts:
        # if isinstance(text, str) and not pd.isna(text): # Check for NaN and empty strings
        doc = nlp(text)
        new_text = []
        for token in doc:
            if token.pos_ in allowed_postags:
                new_text.append(token.lemma_)
        final = " ".join(new_text)
        texts_out.append(final)
    return texts_out

lemmatized_body = lemmatization(data)
print(lemmatized_body[0][0:90])

thank recent letter regard dog poop remain way landscaper make sure future remain pick imm

def gen_words(texts):
    final = []
    for text in texts:
        new = gensim.utils.simple_preprocess(text, deacc=True)
        final.append(new)
    return (final)

data_words = gen_words(lemmatized_body)

print (data_words[0][0:20])

['thank', 'recent', 'letter', 'regard', 'dog', 'poop', 'remain', 'way', 'landscaper', 'make', 'sure', 'future', 'remain', 'pick', 'immed
<img alt="Horizontal scrollbar" data-bbox="100 810 950 823"/>

id2word = corpora.Dictionary(data_words)

corpus = []
for text in data_words:
    new = id2word.doc2bow(text)
    corpus.append(new)

print (corpus[0][0:20])

```

```
word = id2word[[0][:1][0]]
print (word)
```

```
[(0, 1), (1, 1), (2, 1), (3, 2), (4, 1), (5, 1), (6, 1), (7, 2), (8, 1), (9, 4), (10, 1), (11, 1), (12, 1), (13, 2), (14, 1), (15, 1), (16, 1), (17, 1), (18, 1), (19, 1), (20, 1), (21, 1), (22, 1), (23, 1), (24, 1), (25, 1), (26, 1), (27, 1), (28, 1), (29, 1), (30, 1), (31, 1), (32, 1), (33, 1), (34, 1), (35, 1), (36, 1), (37, 1), (38, 1), (39, 1), (40, 1), (41, 1), (42, 1), (43, 1), (44, 1), (45, 1), (46, 1), (47, 1), (48, 1), (49, 1), (50, 1), (51, 1), (52, 1), (53, 1), (54, 1), (55, 1), (56, 1), (57, 1), (58, 1), (59, 1), (60, 1), (61, 1), (62, 1), (63, 1), (64, 1), (65, 1), (66, 1), (67, 1), (68, 1), (69, 1), (70, 1), (71, 1), (72, 1), (73, 1), (74, 1), (75, 1), (76, 1), (77, 1), (78, 1), (79, 1), (80, 1), (81, 1), (82, 1), (83, 1), (84, 1), (85, 1), (86, 1), (87, 1), (88, 1), (89, 1), (90, 1), (91, 1), (92, 1), (93, 1), (94, 1), (95, 1), (96, 1), (97, 1), (98, 1), (99, 1), (100, 1), (101, 1), (102, 1), (103, 1), (104, 1), (105, 1), (106, 1), (107, 1), (108, 1), (109, 1), (110, 1), (111, 1), (112, 1), (113, 1), (114, 1), (115, 1), (116, 1), (117, 1), (118, 1), (119, 1), (120, 1), (121, 1), (122, 1), (123, 1), (124, 1), (125, 1), (126, 1), (127, 1), (128, 1), (129, 1), (130, 1), (131, 1), (132, 1), (133, 1), (134, 1), (135, 1), (136, 1), (137, 1), (138, 1), (139, 1), (140, 1), (141, 1), (142, 1), (143, 1), (144, 1), (145, 1), (146, 1), (147, 1), (148, 1), (149, 1), (150, 1), (151, 1), (152, 1), (153, 1), (154, 1), (155, 1), (156, 1), (157, 1), (158, 1), (159, 1), (160, 1), (161, 1), (162, 1), (163, 1), (164, 1), (165, 1), (166, 1), (167, 1), (168, 1), (169, 1), (170, 1), (171, 1), (172, 1), (173, 1), (174, 1), (175, 1), (176, 1), (177, 1), (178, 1), (179, 1), (180, 1), (181, 1), (182, 1), (183, 1), (184, 1), (185, 1), (186, 1), (187, 1), (188, 1), (189, 1), (190, 1), (191, 1), (192, 1), (193, 1), (194, 1), (195, 1), (196, 1), (197, 1), (198, 1), (199, 1), (200, 1), (201, 1), (202, 1), (203, 1), (204, 1), (205, 1), (206, 1), (207, 1), (208, 1), (209, 1), (210, 1), (211, 1), (212, 1), (213, 1), (214, 1), (215, 1), (216, 1), (217, 1), (218, 1), (219, 1), (220, 1), (221, 1), (222, 1), (223, 1), (224, 1), (225, 1), (226, 1), (227, 1), (228, 1), (229, 1), (230, 1), (231, 1), (232, 1), (233, 1), (234, 1), (235, 1), (236, 1), (237, 1), (238, 1), (239, 1), (240, 1), (241, 1), (242, 1), (243, 1), (244, 1), (245, 1), (246, 1), (247, 1), (248, 1), (249, 1), (250, 1), (251, 1), (252, 1), (253, 1), (254, 1), (255, 1), (256, 1), (257, 1), (258, 1), (259, 1), (260, 1), (261, 1), (262, 1), (263, 1), (264, 1), (265, 1), (266, 1), (267, 1), (268, 1), (269, 1), (270, 1), (271, 1), (272, 1), (273, 1), (274, 1), (275, 1), (276, 1), (277, 1), (278, 1), (279, 1), (280, 1), (281, 1), (282, 1), (283, 1), (284, 1), (285, 1), (286, 1), (287, 1), (288, 1), (289, 1), (290, 1), (291, 1), (292, 1), (293, 1), (294, 1), (295, 1), (296, 1), (297, 1), (298, 1), (299, 1), (300, 1), (301, 1), (302, 1), (303, 1), (304, 1), (305, 1), (306, 1), (307, 1), (308, 1), (309, 1), (310, 1), (311, 1), (312, 1), (313, 1), (314, 1), (315, 1), (316, 1), (317, 1), (318, 1), (319, 1), (320, 1), (321, 1), (322, 1), (323, 1), (324, 1), (325, 1), (326, 1), (327, 1), (328, 1), (329, 1), (330, 1), (331, 1), (332, 1), (333, 1), (334, 1), (335, 1), (336, 1), (337, 1), (338, 1), (339, 1), (340, 1), (341, 1), (342, 1), (343, 1), (344, 1), (345, 1), (346, 1), (347, 1), (348, 1), (349, 1), (350, 1), (351, 1), (352, 1), (353, 1), (354, 1), (355, 1), (356, 1), (357, 1), (358, 1), (359, 1), (360, 1), (361, 1), (362, 1), (363, 1), (364, 1), (365, 1), (366, 1), (367, 1), (368, 1), (369, 1), (370, 1), (371, 1), (372, 1), (373, 1), (374, 1), (375, 1), (376, 1), (377, 1), (378, 1), (379, 1), (380, 1), (381, 1), (382, 1), (383, 1), (384, 1), (385, 1), (386, 1), (387, 1), (388, 1), (389, 1), (390, 1), (391, 1), (392, 1), (393, 1), (394, 1), (395, 1), (396, 1), (397, 1), (398, 1), (399, 1), (400, 1), (401, 1), (402, 1), (403, 1), (404, 1), (405, 1), (406, 1), (407, 1), (408, 1), (409, 1), (410, 1), (411, 1), (412, 1), (413, 1), (414, 1), (415, 1), (416, 1), (417, 1), (418, 1), (419, 1), (420, 1), (421, 1), (422, 1), (423, 1), (424, 1), (425, 1), (426, 1), (427, 1), (428, 1), (429, 1), (430, 1), (431, 1), (432, 1), (433, 1), (434, 1), (435, 1), (436, 1), (437, 1), (438, 1), (439, 1), (440, 1), (441, 1), (442, 1), (443, 1), (444, 1), (445, 1), (446, 1), (447, 1), (448, 1), (449, 1), (450, 1), (451, 1), (452, 1), (453, 1), (454, 1), (455, 1), (456, 1), (457, 1), (458, 1), (459, 1), (460, 1), (461, 1), (462, 1), (463, 1), (464, 1), (465, 1), (466, 1), (467, 1), (468, 1), (469, 1), (470, 1), (471, 1), (472, 1), (473, 1), (474, 1), (475, 1), (476, 1), (477, 1), (478, 1), (479, 1), (480, 1), (481, 1), (482, 1), (483, 1), (484, 1), (485, 1), (486, 1), (487, 1), (488, 1), (489, 1), (490, 1), (491, 1), (492, 1), (493, 1), (494, 1), (495, 1), (496, 1), (497, 1), (498, 1), (499, 1), (500, 1), (501, 1), (502, 1), (503, 1), (504, 1), (505, 1), (506, 1), (507, 1), (508, 1), (509, 1), (510, 1), (511, 1), (512, 1), (513, 1), (514, 1), (515, 1), (516, 1), (517, 1), (518, 1), (519, 1), (520, 1), (521, 1), (522, 1), (523, 1), (524, 1), (525, 1), (526, 1), (527, 1), (528, 1), (529, 1), (530, 1), (531, 1), (532, 1), (533, 1), (534, 1), (535, 1), (536, 1), (537, 1), (538, 1), (539, 1), (540, 1), (541, 1), (542, 1), (543, 1), (544, 1), (545, 1), (546, 1), (547, 1), (548, 1), (549, 1), (550, 1), (551, 1), (552, 1), (553, 1), (554, 1), (555, 1), (556, 1), (557, 1), (558, 1), (559, 1), (560, 1), (561, 1), (562, 1), (563, 1), (564, 1), (565, 1), (566, 1), (567, 1), (568, 1), (569, 1), (570, 1), (571, 1), (572, 1), (573, 1), (574, 1), (575, 1), (576, 1), (577, 1), (578, 1), (579, 1), (580, 1), (581, 1), (582, 1), (583, 1), (584, 1), (585, 1), (586, 1), (587, 1), (588, 1), (589, 1), (590, 1), (591, 1), (592, 1), (593, 1), (594, 1), (595, 1), (596, 1), (597, 1), (598, 1), (599, 1), (600, 1), (601, 1), (602, 1), (603, 1), (604, 1), (605, 1), (606, 1), (607, 1), (608, 1), (609, 1), (610, 1), (611, 1), (612, 1), (613, 1), (614, 1), (615, 1), (616, 1), (617, 1), (618, 1), (619, 1), (620, 1), (621, 1), (622, 1), (623, 1), (624, 1), (625, 1), (626, 1), (627, 1), (628, 1), (629, 1), (630, 1), (631, 1), (632, 1), (633, 1), (634, 1), (635, 1), (636, 1), (637, 1), (638, 1), (639, 1), (640, 1), (641, 1), (642, 1), (643, 1), (644, 1), (645, 1), (646, 1), (647, 1), (648, 1), (649, 1), (650, 1), (651, 1), (652, 1), (653, 1), (654, 1), (655, 1), (656, 1), (657, 1), (658, 1), (659, 1), (660, 1), (661, 1), (662, 1), (663, 1), (664, 1), (665, 1), (666, 1), (667, 1), (668, 1), (669, 1), (670, 1), (671, 1), (672, 1), (673, 1), (674, 1), (675, 1), (676, 1), (677, 1), (678, 1), (679, 1), (680, 1), (681, 1), (682, 1), (683, 1), (684, 1), (685, 1), (686, 1), (687, 1), (688, 1), (689, 1), (690, 1), (691, 1), (692, 1), (693, 1), (694, 1), (695, 1), (696, 1), (697, 1), (698, 1), (699, 1), (700, 1), (701, 1), (702, 1), (703, 1), (704, 1), (705, 1), (706, 1), (707, 1), (708, 1), (709, 1), (710, 1), (711, 1), (712, 1), (713, 1), (714, 1), (715, 1), (716, 1), (717, 1), (718, 1), (719, 1), (720, 1), (721, 1), (722, 1), (723, 1), (724, 1), (725, 1), (726, 1), (727, 1), (728, 1), (729, 1), (730, 1), (731, 1), (732, 1), (733, 1), (734, 1), (735, 1), (736, 1), (737, 1), (738, 1), (739, 1), (740, 1), (741, 1), (742, 1), (743, 1), (744, 1), (745, 1), (746, 1), (747, 1), (748, 1), (749, 1), (750, 1), (751, 1), (752, 1), (753, 1), (754, 1), (755, 1), (756, 1), (757, 1), (758, 1), (759, 1), (760, 1), (761, 1), (762, 1), (763, 1), (764, 1), (765, 1), (766, 1), (767, 1), (768, 1), (769, 1), (770, 1), (771, 1), (772, 1), (773, 1), (774, 1), (775, 1), (776, 1), (777, 1), (778, 1), (779, 1), (780, 1), (781, 1), (782, 1), (783, 1), (784, 1), (785, 1), (786, 1), (787, 1), (788, 1), (789, 1), (790, 1), (791, 1), (792, 1), (793, 1), (794, 1), (795, 1), (796, 1), (797, 1), (798, 1), (799, 1), (800, 1), (801, 1), (802, 1), (803, 1), (804, 1), (805, 1), (806, 1), (807, 1), (808, 1), (809, 1), (810, 1), (811, 1), (812, 1), (813, 1), (814, 1), (815, 1), (816, 1), (817, 1), (818, 1), (819, 1), (820, 1), (821, 1), (822, 1), (823, 1), (824, 1), (825, 1), (826, 1), (827, 1), (828, 1), (829, 1), (830, 1), (831, 1), (832, 1), (833, 1), (834, 1), (835, 1), (836, 1), (837, 1), (838, 1), (839, 1), (840, 1), (841, 1), (842, 1), (843, 1), (844, 1), (845, 1), (846, 1), (847, 1), (848, 1), (849, 1), (850, 1), (851, 1), (852, 1), (853, 1), (854, 1), (855, 1), (856, 1), (857, 1), (858, 1), (859, 1), (860, 1), (861, 1), (862, 1), (863, 1), (864, 1), (865, 1), (866, 1), (867, 1), (868, 1), (869, 1), (870, 1), (871, 1), (872, 1), (873, 1), (874, 1), (875, 1), (876, 1), (877, 1), (878, 1), (879, 1), (880, 1), (881, 1), (882, 1), (883, 1), (884, 1), (885, 1), (886, 1), (887, 1), (888, 1), (889, 1), (890, 1), (891, 1), (892, 1), (893, 1), (894, 1), (895, 1), (896, 1), (897, 1), (898, 1), (899, 1), (900, 1), (901, 1), (902, 1), (903, 1), (904, 1), (905, 1), (906, 1), (907, 1), (908, 1), (909, 1), (910, 1), (911, 1), (912, 1), (913, 1), (914, 1), (915, 1), (916, 1), (917, 1), (918, 1), (919, 1), (920, 1), (921, 1), (922, 1), (923, 1), (924, 1), (925, 1), (926, 1), (927, 1), (928, 1), (929, 1), (930, 1), (931, 1), (932, 1), (933, 1), (934, 1), (935, 1), (936, 1), (937, 1), (938, 1), (939, 1), (940, 1), (941, 1), (942, 1), (943, 1), (944, 1), (945, 1), (946, 1), (947, 1), (948, 1), (949, 1), (950, 1), (951, 1), (952, 1), (953, 1), (954, 1), (955, 1), (956, 1), (957, 1), (958, 1), (959, 1), (960, 1), (961, 1), (962, 1), (963, 1), (964, 1), (965, 1), (966, 1), (967, 1), (968, 1), (969, 1), (970, 1), (971, 1), (972, 1), (973, 1), (974, 1), (975, 1), (976, 1), (977, 1), (978, 1), (979, 1), (980, 1), (981, 1), (982, 1), (983, 1), (984, 1), (985, 1), (986, 1), (987, 1), (988, 1), (989, 1), (990, 1), (991, 1), (992, 1), (993, 1), (994, 1), (995, 1), (996, 1), (997, 1), (998, 1), (999, 1), (1000, 1)]
```

```
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                             id2word=id2word,
                                             num_topics=30,
                                             random_state=100,
                                             update_every=1,
                                             chunksize=100,
                                             passes=10,
                                             alpha="auto")
```

Vizualizing the Data

```
# Print the top 10 keywords in each topic
for topic in lda_model.show_topics(num_topics=30, num_words=10):
    print(topic)

(0, '0.267*schedule' + 0.218*confirm' + 0.053*different' + 0.028*drive' + 0.008*definitely' + 0.001*originally' + 0.000*je' + 0.0
(1, '0.118*meeting' + 0.051*proxy' + 0.037*vote' + 0.033*attend' + 0.025*send' + 0.022*enclose' + 0.021*pm' + 0.021*person' + 0.0
(2, '0.089*intend' + 0.080*message' + 0.068*information' + 0.064*notify' + 0.050*use' + 0.047*immediately' + 0.046*contain' + 0.0
(3, '0.224*cid' + 0.170*image' + 0.150*detail' + 0.118*png' + 0.096*portal' + 0.026*ad' + 0.020*ff' + 0.014*violation' + 0.014*
(4, '0.323*board' + 0.093*visit' + 0.090*available' + 0.078*much' + 0.045*answer' + 0.037*agreement' + 0.033*member' + 0.032*agr
(5, '0.074*notice' + 0.051*today' + 0.035*click' + 0.034*monthly' + 0.029*try' + 0.026*here' + 0.026*then' + 0.026*mail' + 0.022
(6, '0.127*approval' + 0.127*then' + 0.097*very' + 0.078*move' + 0.058*leave' + 0.040*remind' + 0.038*parking' + 0.034*location'
(7, '0.297*complete' + 0.081*here' + 0.075*mention' + 0.075*directly' + 0.057*dd' + 0.007*removal' + 0.007*return' + 0.000*uploa
(8, '0.193*com' + 0.133*https' + 0.094*pay' + 0.092*www' + 0.075*http' + 0.045*png' + 0.034*cc' + 0.028*jpg' + 0.022*images' +
(9, '0.060*nd' + 0.000*hqgfaceycmuha' + 0.000*fuglazyjzkvifn' + 0.000*nkcu' + 0.000*mfwkka' + 0.000*laird' + 0.000*kyunboggvdr1
(10, '0.124*need' + 0.066*make' + 0.055*time' + 0.052*take' + 0.048*go' + 0.046*include' + 0.045*day' + 0.037*change' + 0.030*f
(11, '0.067*email' + 0.062*document' + 0.030*sign' + 0.026*docusign' + 0.023*https' + 0.019*link' + 0.018*sender' + 0.016*net' +
(12, '0.061*de' + 0.042*jjtrmfo' + 0.042*profile' + 0.034*image' + 0.032*mc' + 0.031*di' + 0.030*iogfmlwi' + 0.030*fxpeqzuyqhye
(13, '0.151*week' + 0.143*homeowner' + 0.127*just' + 0.114*next' + 0.083*there' + 0.074*project' + 0.043*box' + 0.013*at' + 0.00
(14, '0.120*file' + 0.114*unit' + 0.114*fee' + 0.065*close' + 0.047*ce' + 0.043*open' + 0.038*place' + 0.034*fd' + 0.032*be' +
(15, '0.137*year' + 0.087*last' + 0.070*response' + 0.069*tree' + 0.055*come' + 0.033*roof' + 0.031*consider' + 0.030*early' +
(16, '0.106*send' + 0.085*email' + 0.053*th' + 0.037*work' + 0.032*letter' + 0.031*provide' + 0.031*owner' + 0.031*receive' +
(17, '0.108*job' + 0.062*sheet' + 0.051*order' + 0.049*track' + 0.044*email' + 0.039*receive' + 0.032*follow' + 0.027*contact' +
(18, '0.175*message' + 0.059*subject' + 0.057*recipient' + 0.037*sender' + 0.034*pass' + 0.034*com' + 0.030*securely' + 0.028*or
(19, '0.223*reply' + 0.090*mailto' + 0.078*db' + 0.034*dc' + 0.000*managemowed' + 0.000*kyunboggvdr1' + 0.000*laird' + 0.000*je'
(20, '0.162*thank' + 0.097*know' + 0.093*let' + 0.068*question' + 0.067*call' + 0.062*reach' + 0.059*kindly' + 0.032*cid' + 0.03
(21, '0.076*account' + 0.049*office' + 0.038*note' + 0.035*list' + 0.032*communication' + 0.031*violation' + 0.030*as' + 0.027*f
(22, '0.134*service' + 0.118*contact' + 0.104*ask' + 0.083*more' + 0.051*forward' + 0.048*free' + 0.048*correct' + 0.045*feel' +
(23, '0.283*payment' + 0.244*check' + 0.119*invoice' + 0.048*get' + 0.033*appreciate' + 0.025*aka' + 0.012*speak' + 0.011*https'
(24, '0.121*time' + 0.072*pm' + 0.071*hour' + 0.067*cost' + 0.053*turnaround' + 0.048*am' + 0.045*day' + 0.043*business' + 0.035
(25, '0.362*address' + 0.088*again' + 0.052*building' + 0.050*mailing' + 0.030*line' + 0.029*confirm' + 0.029*actual' + 0.026*cl
(26, '0.154*policy' + 0.059*term' + 0.058*material' + 0.046*condition' + 0.019*ongoing' + 0.015*mailbox' + 0.003*cg' + 0.000*omc
(27, '0.164*pende' + 0.083*mail' + 0.063*review' + 0.054*recipient' + 0.053*below' + 0.049*error' + 0.044*no' + 0.039*long' +
(28, '0.047*click' + 0.043*upn' + 0.036*net' + 0.032*proposal' + 0.031*ct' + 0.031*ls' + 0.030*gdw' + 0.030*ndf' + 0.030*sendgr
(29, '0.110*good' + 0.054*financial' + 0.049*sure' + 0.046*meet' + 0.042*create' + 0.038*way' + 0.035*few' + 0.030*repair' + 0.0
```

```
# Create a directory to store the images
if not os.path.exists("wordclouds"):
    os.makedirs("wordclouds")

# Create a word cloud for each topic
for i, topic in lda_model.show_topics(num_topics=30, num_words=10, formatted=False):
    topic_words = dict(topic)
    wc = WordCloud(width=800, height=400, background_color="white")
    wc.generate_from_frequencies(topic_words)
    plt.figure(figsize=(10, 5))
    plt.imshow(wc, interpolation="bilinear")
    plt.axis("off")
    plt.title("Topic " + str(i))
    plt.savefig("wordclouds/topic " + str(i) + ".png", bbox_inches="tight")
    plt.show()
```

Topic 0

different drive
definitely
schedule
confirm
originally

Topic 1

member
meeting
attend
vote
enclose
proxy
send
person

Topic 2

receive use
intend
message
information
immediately
name
notify
contain
confidential

Topic 3

— | — , — • 7 — ff

```

all_keywords = []

for i in range(30):
    keywords = lda_model.get_topic_terms(i, topn=10)
    all_keywords.extend([id2word[id] for id, _ in keywords])

len(all_keywords)

300

# Get word-topic probability distribution for each topic
word_probs = []
for topic in lda_model.show_topics(num_topics=30, num_words=100, formatted=False):
    word_probs.append(dict(topic[1]))

# Sum the probabilities for each word across all topics
word_importance = {}
for topic in word_probs:
    for word, prob in topic.items():
        if word in word_importance:
            word_importance[word] += prob
        else:
            word_importance[word] = prob

# Sort the words by their importance in the model
sorted_words = sorted(word_importance.items(), key=lambda x: x[1], reverse=True)

# Create a bar chart of the top 50 words
top_words = sorted_words[:30]
words = [w[0] for w in top_words]
importance = [w[1] for w in top_words]
plt.bar(words, importance)
plt.xticks(rotation=90)
plt.xlabel("Word")
plt.ylabel("Importance")
plt.title("Word Importance in LDA Model")
plt.show()

```

