In [71]:
```
!pip install modernize
```

```
Requirement already satisfied: modernize in f:\anaconda\lib\site-packages
(0.8.0)
Requirement already satisfied: fissix in f:\anaconda\lib\site-packages (from mo
dernize) (24.4.24)
Requirement already satisfied: appdirs>=1.4.4 in f:\anaconda\lib\site-packages
(from fissix->modernize) (1.4.4)
```

In [72]:
```
!python.exe -m pip install --upgrade pip
```

```
Requirement already satisfied: pip in f:\anaconda\lib\site-packages (25.3)
```

In [74]:
```
## DOWNLOADING RESOURSES
import nltk
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')

!pip install dateparser
!pip install num2words
```

```
[nltk_data] Downloading package punkt to C:\Users\NYB
[nltk_data]     COMPUTER\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to C:\Users\NYB
[nltk_data]     COMPUTER\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to C:\Users\NYB
[nltk_data]     COMPUTER\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
Requirement already satisfied: dateparser in f:\anaconda\lib\site-packages
(1.2.2)
Requirement already satisfied: python-dateutil>=2.7.0 in f:\anaconda\lib\site-p
ackages (from dateparser) (2.8.2)
Requirement already satisfied: pytz>=2024.2 in f:\anaconda\lib\site-packages (f
rom dateparser) (2025.2)
Requirement already satisfied: regex>=2024.9.11 in f:\anaconda\lib\site-package
s (from dateparser) (2025.11.3)
Requirement already satisfied: tzlocal>=0.2 in f:\anaconda\lib\site-packages (f
rom dateparser) (2.1)
Requirement already satisfied: six>=1.5 in f:\anaconda\lib\site-packages (from
python-dateutil>=2.7.0->dateparser) (1.16.0)
Requirement already satisfied: num2words in f:\anaconda\lib\site-packages
(0.5.14)
Requirement already satisfied: docopt>=0.6.2 in f:\anaconda\lib\site-packages
(from num2words) (0.6.2)
```

In [76]:
```
## importing all necessary libraries
import dateparser
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from num2words import num2words
```

```
import re
```

Converting time to words

In [80]:
```
def convert_time_to_words(time_str):
    time_obj = dateparser.parse(time_str)
    hour = time_obj.hour
    minute = time_obj.minute
    am_pm = "AM" if hour < 12 else "PM"
    hour_12 = hour % 12 if hour != 0 else 12
    if minute == 0:
        time_words = f"{num2words(hour_12)} o'clock {am_pm}"
    else:
        time_words = f"{num2words(hour_12)} {num2words(minute)} {am_pm}"
    return time_words
```

Converting Date to words

In [84]:
```
def convert_date_to_words(date_string):
    try:
        date_obj = dateparser.parse(date_string)
        if date_obj:
            return date_obj.strptime("%B") + " " + num2words(date_obj.day, ord
    except Exception:
        pass
    return date_string
```

Text Normalizing

In [88]:
```
def normalize_text(text):
    tokens = word_tokenize(text)  ## tokenizing words

    normalized_tokens = []
    for token in tokens:
        try:
            date_obj = dateparser.parse(token)
            if date_obj:
                if ":" in token:
                    normalized_tokens.append(convert_time_to_words(token))
                else:
                    normalized_tokens.append(convert_date_to_words(token))
                continue
        except Exception as e:
            pass
        date_formats = ["%Y/%m/%d", "%Y\\%m\\%d", "%Y-%m-%d"]
        for date_format in date_formats:
            try:
                date_obj = dateparser.parse(token, date_formats=[date_format])
```

```
                if date_obj:
                    normalized_tokens.append(convert_date_to_words(token))
                    break
            except Exception as e:
                pass
        else:
            if token.isdigit() and 0 <= int(token) <=1000:
                normalized_tokens.append(num2words(int(token)))
            else:
                if token.isalpha():  ## removing puctuation and special charac
                    token = token.lower()  # converting to lower case
                    lemmatizer = WordNetLemmatizer()
                    token = lemmatizer.lemmatize(token)
                    normalized_tokens.append(token)
                else:
                    cleaned_token = re.sub(r'[^\w\s]', '', token)
                    if cleaned_token:
                        cleaned_token = cleaned_token.lower()
                        lemmatizer = WordNetLemmatizer()
                        cleaned_token = lemmatizer.lemmatize(cleaned_token)
                        normalized_tokens.append(cleaned_token)
                    else:
                        normalized_tokens.append(token)

    return ' '.join(normalized_tokens)
```

In [90]:
```
text = "This is an example sentence with 123 numbers on 01/01/1996 at 14:30!"
print(normalize_text(text))
```

```
this is an example sentence with one hundred and twenty-three number on 01/01/1
996 at two thirty PM !
```

In [ ]:

In [ ]: