NLP Lab(1)

Email Meta Data

Name	Sec	BN	Code
Ahmed Hany Farouk	1	10	9202213
Basma Hatem Elhoseny	1	16	9202381

We have tried to make our regex as simple as possible 😉😉

Getting Mail:

Every mail start with this pattern From r …

```
From r  Wed Oct 30 21:41:56 2002
X-Sieve: cmu-sieve 2.0
Return-Path: <james_ngola2002@maktoob.com>
Message-Id: <200210310241.g9V2fNm6028281@cs.CU>
From: "MR. JAMES NGOLA." <james_ngola2002@maktoob.com>
Reply-To: james_ngola2002@maktoob.com
To: webmaster@aclweb.org
Date: Thu, 31 Oct 2002 02:38:20 +0000
Subject: URGENT BUSINESS ASSISTANCE AND PARTNERSHIP
X-Mailer: Microsoft Outlook Express 5.00.2919.6900 DM
MIME-Version: 1.0
Content-Type: text/plain; charset="us-ascii"
Content-Transfer-Encoding: 8bit
X-MIME-Autoconverted: from quoted-printable to 8bit by sideshowmel.si.UM id g9V2foW24311
Status: O

FROM:MR. JAMES NGOLA.
CONFIDENTIAL TEL: 233-27-587908.
E-MAIL: (james_ngola2002@maktoob.com).

URGENT BUSINESS ASSISTANCE AND PARTNERSHIP.


DEAR FRIEND,

I AM ( DR.) JAMES NGOLA, THE PERSONAL ASSISTANCE TO THE LATE CONGOLESE (PRESIDENT LAURENT KABILA)
```

```python
# ######################### TODO: write the pattern to split the emails on in the raw string below ######################
start_email_pattern = r"^From\sr\s"
# ######################################################################################################################

# contents = re.split(start_email_pattern, f)
contents = re.split(start_email_pattern, f, flags=re.MULTILINE)
contents.pop(0)
assert len(contents) == 3976, "Your pattern is wrong"
```

Note: We have added the multi-line flag to the split function so that we capture From at the beginning of lines only not to be in the body of the mail it self or so

Sender Information:

The sender's information is extracted from the From tag in the meta data of the mail

```
From r  Wed Oct 30 21:41:56 2002
Return-Path: <james_ngola2002@maktoob.com>
X-Sieve: cmu-sieve 2.0
Return-Path: <james_ngola2002@maktoob.com>
Message-Id: <200210310241.g9V2fNm6028281@cs.CU>
From: "MR. JAMES NGOLA." <james_ngola2002@maktoob.com>
Reply-To: james_ngola2002@maktoob.com
To: webmaster@aclweb.org
Date: Thu, 31 Oct 2002 02:38:20 +0000
Subject: URGENT BUSINESS ASSISTANCE AND PARTNERSHIP
X-Mailer: Microsoft Outlook Express 5.00.2919.6900 DM
MIME-Version: 1.0
Content-Type: text/plain; charset="us-ascii"
Content-Transfer-Encoding: 8bit
X-MIME-Autoconverted: from quoted-printable to 8bit by sideshowmel.si.UM id g9V2foW24311
Status: O

FROM:MR. JAMES NGOLA.
CONFIDENTIAL TEL: 233-27-587908.
E-MAIL: (james_ngola2002@maktoob.com).

URGENT BUSINESS ASSISTANCE AND PARTNERSHIP.
```

```python
def senderEmailAndName(email_item):
    ##################### TODO: write regex that can match the whole line containing the sender name and email #########
    # sender_name_email_line_regex = r"From:.*@.*"
    # sender_name_email_line_regex = r"From(.*<.+>|.*@.*)"
    sender_name_email_line_regex = r"(?i)^From:(.*<.+>|.*@.*)"
    ##########################################################################################################
    sender = re.search(sender_name_email_line_regex, email_item, flags=re.MULTILINE)

    # This check is in case there is no sender found (We don't know if all emails contain this info)
    if sender is not None:
        # Since we have the line You need to do the following
        # You can capture the email by using only one regex (only one re.search())
        # To capture the name safely you can write regex that captures all characters from the end of the start line pattern
        # to the pattern the starts the email

        ##################### TODO: write regex that captures the email, and the name ##########################
        # email regex: this will capture the email exactly
        # email_regex = r"[\w\._%+-]+@[\w\._%+-]+"
        # email_regex = r"\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,7}\b"
        email_regex = r"\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,7}\b"
        # name regex: this will capture the name besides some characters before and after it that will be removed later
        # name_regex=r"(\".+\"|(?<=:).*(?=<))"
        # name_regex = r"(\".*\"|:.*<)"
        name_regex = r"(\".*\"|:.*<)"
        ########################################################################################################
        s_email = re.search(email_regex, sender.group())
        s_name = re.search(name_regex, sender.group())
    else:
```

Cases Handled in sender's info regex:

1. From can be small or capital
2. Sender Name isn't enclosed in ""
3. The email isn't enclosed inside <>
4. Sometimes there is no sender name only mail in this case we assume sender name is none
5. `If From Tag isn't found then the senders's info is all Nones`

Missed Counts:

	Count
Sender	
Nones in Lines of Sender_info	369
Nones in sender_mail	377
Nones in sender_name	790

Receipt Information:

The receipt's information is extracted from the To tag in the meta data of the mail

```
From r  Wed Oct 30 21:41:56 2002
Return-Path: <james_ngola2002@maktoob.com>
X-Sieve: cmu-sieve 2.0
Return-Path: <james_ngola2002@maktoob.com>
Message-Id: <200210310241.g9V2fNm6028281@cs.CU>
From: "MR. JAMES NGOLA." <james_ngola2002@maktoob.com>
Reply-To: james_ngola2002@maktoob.com
To: webmaster@aclweb.org
Date: Thu, 31 Oct 2002 02:38:20 +0000
Subject: URGENT BUSINESS ASSISTANCE AND PARTNERSHIP
X-Mailer: Microsoft Outlook Express 5.00.2919.6900 DM
MIME-Version: 1.0
Content-Type: text/plain; charset="us-ascii"
Content-Transfer-Encoding: 8bit
X-MIME-Autoconverted: from quoted-printable to 8bit by sideshowmel.si.UM id g9V2foW24311
Status: O

FROM:MR. JAMES NGOLA.
CONFIDENTIAL TEL: 233-27-587908.
E-MAIL: (james_ngola2002@maktoob.com).

URGENT BUSINESS ASSISTANCE AND PARTNERSHIP.
```

```python
def recipientEmailAndName(email_item):
    ######################## TODO: write regex that can match the whole line containing the recipient name and email #########
    recipient_name_email_line_regex = r"(?i)^to:.*@.*"
    ###################################################################################################################
    recipient = re.search(recipient_name_email_line_regex, email_item, flags=re.MULTILINE)

    # This check is in case there is no sender found (We don't know if all emails contain this info)
    if recipient is not None:
        # Since we have the line You need to do the following
        # You can capture the email by using only one regex (only one re.search())
        # To capture the name safely you can write regex that captures all characters from the end of the start line pattern
        # to the pattern the starts the email

        ######################### TODO: write regex that captures the email, and the name ############################
        # email regex: this will capture the email exactly
        # email_regex = r"[\w\.-]+@[\w\.-]+|<([^>]+)>"
        email_regex = r"[^\s]+@[^\s]+|<([^>]+)>"
        # name regex: this will capture the name besides some characters before and after it that will be removed later
        # name_regex = r"[\w\.-]+@"
        name_regex = r"[^\s]+@"
        ###############################################################################################################
```

Cases Handled in receipt's info regex:

1. To can be small or capital
2. Receipt Name is extracted from the mail as the first part before @
3. Assumed to have no receiver in cases below
    a. To: undisclosed-recipients: ;
    b. To: N/A <>, N/A <>
    c. To: N/A <>
4. If To Tag isn't found then the receipt info is all Nones

Missed Counts: This large no are nearly the same in the three filed this mean that these mails already don't have the To Tag so the info can't be extracted 😁

Recipt	
Nones in Lines of Reciept_info	1503
Nones in Reciept_mail	1505
Nones in Reciept_name	1505

Date Information:

The date information is extracted from the Date tag in the meta data of the mail

```
From r  Wed Oct 30 21:41:56 2002
Return-Path: <james_ngola2002@maktoob.com>
X-Sieve: cmu-sieve 2.0
Return-Path: <james_ngola2002@maktoob.com>
Message-Id: <200210310241.g9V2fNm6028281@cs.CU>
From: "MR. JAMES NGOLA." <james_ngola2002@maktoob.com>
Reply-To: james_ngola2002@maktoob.com
To: webmaster@aclweb.org
Date: Thu, 31 Oct 2002 02:38:20 +0000
Subject: URGENT BUSINESS ASSISTANCE AND PARTNERSHIP
X-Mailer: Microsoft Outlook Express 5.00.2919.6900 DM
MIME-Version: 1.0
Content-Type: text/plain; charset="us-ascii"
Content-Transfer-Encoding: 8bit
X-MIME-Autoconverted: from quoted-printable to 8bit by sideshowmel.si.UM id g9V2foW24311
Status: O

FROM:MR. JAMES NGOLA.
CONFIDENTIAL TEL: 233-27-587908.
E-MAIL: (james_ngola2002@maktoob.com).

URGENT BUSINESS ASSISTANCE AND PARTNERSHIP.
```

```python
# This function takes a full email and returns a string with the date

def getDate(email_item):
    ########################## TODO: write regex to match the whole line containing the date ######################
    line_date_regex = r"(?i)^date:.*"
    ###############################################################################################################
    date_field = re.search(line_date_regex, email_item, flags=re.MULTILINE)

    if date_field is not None:
        ################################### TODO: write regex to match only the date ##############################
        # e.g. If the date is: Thu, 31 Oct 2002 22:17:55 +0100
        # you need to match only 31 Oct 2002
        # date_only_regex = r"(?:\d{2}|\d{1})\s\w{3}\s(\d{4}|\d{2})"
        date_only_regex = r"(?:\d{2}|\d{1})(th|st|nd|rd)?\s\w{3}.?\s(\d{4}|\d{2})"
        ###########################################################################################################
        date = re.search(date_only_regex, date_field.group())
    else:
        date = None

    if date is not None:
        date_sent = date.group()
```

Cases Handled in date's info regex:

1. Date can be small or capital
2. long date
    a. Date: Wed, 14 Jul 04 16:24:45 ﷺ
3. Abbreviated date
    a. DATE:13th Oct. 2003
4. Year sometimes are written in 2 digit instead of 4
    a. DATE:13th Oct. 2003
5. If Date Tag isn't found then the date is None

Missed Counts: This large no are nearly the same in the 2 fields this means that these mails already doesn't have the Date Tag so the info can't be extracted 😁

Date	
Nones in Lines of Date_info	547
Nones in Date	547

Subject Information:

The subject information is extracted from the Subject tag in the meta data of the mail

```
From r  Wed Oct 30 21:41:56 2002
Return-Path: <james_ngola2002@maktoob.com>
X-Sieve: cmu-sieve 2.0
Return-Path: <james_ngola2002@maktoob.com>
Message-Id: <200210310241.g9V2fNm6028281@cs.CU>
From: "MR. JAMES NGOLA." <james_ngola2002@maktoob.com>
Reply-To: james_ngola2002@maktoob.com
To: webmaster@aclweb.org
Date: Thu, 31 Oct 2002 02:38:20 +0000
Subject: URGENT BUSINESS ASSISTANCE AND PARTNERSHIP
X-Mailer: Microsoft Outlook Express 5.00.2919.6900 DM
MIME-Version: 1.0
Content-Type: text/plain; charset="us-ascii"
Content-Transfer-Encoding: 8bit
X-MIME-Autoconverted: from quoted-printable to 8bit by sideshowmel.si.UM id g9V2foW24311
Status: O

FROM:MR. JAMES NGOLA.
CONFIDENTIAL TEL: 233-27-587908.
E-MAIL: (james_ngola2002@maktoob.com).

URGENT BUSINESS ASSISTANCE AND PARTNERSHIP.
```

```python
# This function takes a full email item and returns a string with the email subject
def getSubject(email_item):
    ############################ Done: write regex to match the whole line containing the subject ################
    line_subject_regex = r"(?i)^subject: .*"
    ###########################################################################################################
    subject_field = re.search(line_subject_regex, email_item, flags=re.MULTILINE)

    if subject_field is not None:
        ################### Done: write regex with the pattern to be removed to capture the subject only #########
        remove_pattern = r"(?i)subject: "
        ###########################################################################################################
        subject = re.sub(remove_pattern, "", subject_field.group())
    else:
        subject = None

    # return subject
    s_line=None
    if subject:
        s_line=subject_field.group()

    return s_line,subject
```

Cases Handled in date's info regex:

1. Subject can be small or capital
2. If Subject Tag is Empty then the value is None
3. If Subject Tag isn't found then the date is None

Missed Counts: 39 Meaning that Simply we count get all the Subjects in the mails that have the subject tag else they aren't present in other words so empty subject was found

Subject	
Nones in Lines of Subject_info	39
Nones in Subject	39

Note: Attached with this pdf 1_10_1_16_Stat.csv with the statistics of the Nones in the tag line itself 😊