UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II

Facoltà di Ingegneria Informatica

PROGETTO INTELLIGENZA ARTIFICIALE: Analisi Testuale.

Professoressa:
Amato Flora

A.A. 2018/2019

Studenti:

COLANTUONO Antonio	***N46003371***
COPPOLA Vincenzo	***N46003356***
DELLA TORCA Salvatore	***N46003208***
IZZO Alberto	***N46003425***

INDICE

1. Analisi Testuale

L'analisi del contenuto è un insieme di tecniche, manuali o assistite da computer, che hanno come obiettivo:

- interpretare documenti provenienti da processi di comunicazione in senso proprio, come ad esempio testi;
- assegnare un significato a tracce e manufatti per poter produrre inferenze valide e attendibili.

Una volta definito l'insieme dei testi da analizzare, è necessario eseguire una serie di operazioni al fine di curare l'organizzazione interna e la trascrizione.
In particolare, bisogna verificare:

- *la comparabilità dei testi;*
- *la disponibilità di una o più caratteristiche da associare a ciascun frammento;*
- *che i testi siano sufficientemente lunghi al fine di rendere vantaggioso il ricorso a tecniche automatiche di analisi.*

Il punto di partenza è l'analisi delle parole che compongono il corpus al fine di individuare le cosiddette "*parole tema*", che consentono di cogliere immediatamente gli argomenti principali del testo.
I passi di analisi che consentono di descrivere in modo semi automatico il contenuto di un testo sono i seguenti:

- *analisi dei segmenti ripetuti*: sono quelle forme composte, costituite da parole che compaiono nel corpus con la stessa sequenza, che aiutano a fornire una rappresentazione sintetica dei contenuti del corpus e a individuare rapidamente attori, oggetti e azioni su cui è strutturato il testo;

- *analisi delle co-occorrenze*: consente di studiare le associazioni tra parole, individuando quelle parole che compaiono più spesso vicine tra loro;

- *analisi delle parole caratteristiche*: consente di differenziare le diverse parti di un testo, evidenziando quelle parole che sono sovra-rappresentate nel linguaggio di una certa categoria di autori, in modo da caratterizzare il linguaggio*;*

- *analisi del linguaggio peculiare*: a ciascuna parola è associata una frequenza che serve a indicare l'uso atteso di ogni parola nella comunità linguistica a cui il lessico è riferito;

- *analisi delle parole con caratteristiche grammaticali omogenee:* utile perché, ad esempio, l'insieme dei verbi ricondotti al lemma può fornire una graduatoria delle azioni menzionate nel testo, l'insieme degli aggettivi fornisce elementi per valutare il tono di un testo, oppure l'insieme e il tipo di pronomi può dar conto del tipo di interazione presente tra soggetti che caratterizza il testo;

- *analisi delle concordanze*: tecnica che consente di analizzare il contesto d'uso di una parola di interesse o di gruppi di parole con la stessa radice visualizzando le n parole precedenti e le n successive alla parola in analisi, tutte le volte che questa compare nel corpus al fine di risolvere alcune delle ambiguità semantiche e di ricostruire per ogni parola i riferimenti tematici a cui questa rinvia, definendo una mappa concettuale tra parole e temi affrontati;

- *analisi delle corrispondenze lessicali*: consente di sintetizzare l'informazione contenuta in una grossa matrice di dati testuali, visualizzando l'associazione tra le parole all'interno del testo in analisi e cercando la migliore rappresentazione simultanea degli elementi in modo da studiare le dipendenze tra caratteri.

2. Specifiche di progetto e strumenti utilizzati

Si vuole utilizzare la libreria SPACY effettuare l'analisi testuale. In particolare si vogliono analizzare i "tweet" prelevati dall'applicazione Twitter, al fine di effettuare le operazioni di analisi che sono state elencate in precedenza.

2.1 spaCy

spaCy è una libreria software open-source per l'elaborazione avanzata del linguaggio naturale, scritta nei linguaggi di programmazione Python e Cython.
Le operazioni possibili utilizzando la libreria *spaCy* sono molteplici:

- *tokenizzazione;*
- *named entity recognition;*
- *grammatical tagging;*
- *text classification;*
- *deep learning integration.*

2.2 Python

Python è un linguaggio di programmazione ad alto livello, orientato agli oggetti e adatto a sviluppare applicazioni distribuite e scripting. È un linguaggio multi-paradigma che ha tra i principali obiettivi dinamicità, semplicità e flessibilità e inoltre supporta il paradigma object oriented, la programmazione strutturata e molte caratteristiche di programmazione funzionale e riflessione.
Le caratteristiche più immediatamente riconoscibili di Python sono le variabili non tipizzate e l'uso dell'indentazione per la definizione delle specifiche.
Altre caratteristiche distintive sono l'overloading di operatori e funzioni tramite delegation, la presenza di un ricco assortimento di tipi e funzioni di base e librerie standard.
Il controllo dei tipi è forte (*strong typing*) e viene eseguito a runtime (*dynamic typing*): una variabile è un contenitore a cui viene associata un'etichetta che rappresenta il nome della variabile e che può essere associata a diversi contenitori anche di tipo diverso durante il suo tempo di vita.
La gestione della memoria è effettuata dai garbage collector che sono responsabili di liberare automaticamente la memoria cancellando tutte quelle risorse che sono inutilizzate.

3. Descrizione analisi

N.B: È stato utilizzato il sistema operativo Linux Mint con la versione 2.7 di Python.
Per l'installazione di spaCy è stato prima installato il comando ***pip*** *e poi successivamente è stato eseguito il comando*

pip install -U spaCy

Al termine dell'installazione è stato eseguito il comando per scaricare la lingua inglese

python -m spaCy download en_core_web_sm

Per prima cosa sono stati scaricati dei tweet pubblicati da *Donald Trump*.
Il numero di tweet analizzati è **19**, ma con lo script realizzato è in realtà possibile analizzare un qualsiasi numero di tweet.
Lo script realizzato ha come obiettivo soddisfare tutte i punti che compongono una *pipeline NLP:*

- *pretrattamento del corpus*: è stata realizzata una correzione manuale dei caratteri utilizzati nei tweet al fine di poter rendere tutti i caratteri riconoscibili, secondo il codice ASCII;
- *annotazione morfo-semantica:* è stata assegnata ad ogni termine la categoria di appartenenza e, successivamente, è stato individuato il lemma;
- *analisi lessico-metrica:* Noun Chunks e Navigating the Parse Tree individuano per ogni termine tutte le parole che sono ad esse associate e le inseriscono all'interno di una lista;
- *estrazione dei termini peculiari:* la Named Entity Recognition identifica i nomi e i concetti con maggiore importanza;
- *identificazione delle relazioni:* è stato utilizzato un approccio ***endogeno*** in quanto le informazioni sono state estratte dai corpus stessi.

In primis tutti i tweet sono stati inseriti all'interno di un unico documento di testo.
Utilizzando poi il linguaggio python è stato realizzato lo script di analisi che adesso andremo ad analizzare.

import spacy è un'istruzione che serve per importare la libreria scaricata in precedenza.
È stato poi caricato il modello della lingua inglese attraverso la funzione *spacy.load()* ed è stato aperto in lettura il file contenente i tweet.
Dato che ogni tweet occupa una sola riga del documento di testo, sfruttando una variabile indice, è stato possibile tener conto di quale fosse il tweet analizzato.

```
1 President Obama just had a news conference, but he doesn't have a clue. Our country is a divided crime scene, and it will only get worse!
2 If it were up to goofy Elizabeth Warren, we'd have no jobs in America-she doesn't have a clue.
3 David Brooks, of the New York Times, is closing in on being the dumbest of them all. He doesn't have a clue.
4 Mitt Romney is a mixed up man who doesn't have a clue. No wonder he lost!
5 I am going to save Medicare and Medicaid, Carson wants to abolish, and failing candidate Gov. John Kasich doesn't have a clue - weak!
6 Just watched Marco Rubio on television. Just another all talk, no action, politician. Truly doesn't have a clue! Worst voting record in Sen.
7 Rich Lowry is truly one of the dumbest of the talking heads - he doesn't have a clue!
8 Whoopi Goldberg had better surround herself with better hosts than Nicole Wallace, who doesn't have a clue. The show is close to death!
9 You save your $. Don't invest in Karl Rove. He doesn't have a clue.
10 I am the BEST builder, just look at what I've built. Hillary can't build. Republican candidates can't build. They don't have a clue!
11 Defense Sec.Hagel has quit. Great news for our country. The guy didn't have a clue-grossly outmatched by our enemies. Couldn't even speak
12 I just had to fire someone, he didn't have a clue--he reminded me of Obama on Wednesday night.
13 I am going to keep our jobs in the U.S., and totally rebuild our crumbling infrastructure. Crooked Hillary has no clue! @Teamsters
14 I'd like to call JEB a liar, but the truth is he has no clue & never revealed that he used Eminent Domain- when criticizing me!
15 Pamela Geller is a total whack job who doesn't have a clue. Don't provoke the enemy, go get them and make them pay. No signals, just do it!
16 Why does CNN bore their audience with people like @secupp, a totally biased loser who doesn't have a clue. I hear she will soon be gone!
17 When will @CNN get some real political talent rather than political commentators like Errol Louis, who doesn't have a clue! Others bad also.
18 What a waste of time being interviewed by Anderson Cooper when he puts on really stupid talking heads likeTim O'Brien-dumb guy with no clue!
19 Our country needs a president with great leadership skills and vision, not someone like Hillary or Barack, neither of which has a clue!
```

tweets.txt × sp.py × anallizati.txt ×

```
1  import spacy
2
3  nlp = spacy.load('en_core_web_sm')
4  tweets = open("tweets.txt", "r")
5  indice = 0
6  for contenuti in tweets.readlines():
7      indice += 1
8      doc = nlp(unicode(contenuti, "utf-8"))
9      analizzato = open("anallizati.txt", 'a')
10
11     analizzato.write("**********************************************ANALIZING TWEET NUMBER " + str(indice) + "...**********************************************\n")
```

L'istruzione successiva è la *lemmatizzazione* che, per ogni termine, individua il **lemma**, cioè la forma base, il **pos**, cioè la categoria di appartenenza del termine, il **tag** che è l'etichetta dettagliata della categoria del termine e infine il **dep**, cioè la relazione sintattica tra i vari token che si ottiene attraverso tecniche di I.E.(*information extraction).*

```
analizzato.write("**********************************************ANALIZING TWEET NUMBER " + str(indice) + "...**********************")
analizzato.write("LEMMATIZZTION\n")
analizzato.write("TEXT    LEMMA    POS    TAG    DEP\n")
for token in doc:
    analizzato.write(token.text + '    ' +  token.lemma_ + '    ' + token.pos_ + '    ' + token.tag_ + '    ' + token.dep_ + '\n')
```

Successivamente sono stati analizzati i “NOUN CHUNKS”, ossia "frasi di base” che hanno un nome come testa.
In questa fase si determinano:

- il *root text,* testo originale della parola che collega il noun chunk al resto della frase;
- il *root dep,* relazione di dipendenza che collega la radice alla sua testa;
- il *root head text,* cioè il testo della testa del token della radice.

```
17    analizzato.write("\nNOUN CHUNKS\n")
18    analizzato.write("TEXT    ROOT_TEXT    ROOT_DEP    ROOT_HEAD_TEXT\n")
19    for chunk in doc.noun_chunks:
20        analizzato.write(chunk.text + '    ' + chunk.root.text + '    ' + chunk.root.dep_ + '    ' + chunk.root.head.text + '\n')
21
```

Navigating the parse tree vuole invece analizzare l’albero delle dipendenze, connettendo i termini che dipendono tra loro con un arco.
In questa fase di analisi possiamo allora individuare il **dep**, cioè la relazione sintattica che collega il “figlio” alla testa, l’ **head text**, cioè il testo originale della testa del token, l’**head pos**, ossia la categoria di appartenenza della testa, ed infine il **children**, cioè il figlio (inteso come relazione di dipendenza) della testa.

```
analizzato.write("\nNAVIGATING THE PARSE TREE\n")
analizzato.write("TEXT    DEP    HEAD_TEXT    HEAD_POS    CHILDREN\n")
for token in doc:
    analizzato.write(token.text + '    ' + token.dep_ + '    ' + token.head.text + '    ' + token.head.pos_ + '    ' + str([child for child
```

Successivamente è stata realizzata l’iterazione rispetto all’albero delle dipendenze *locale*: per ogni termine si identificano i termini da cui essa dipende.
Gli attributi *Token.lefts* e *Token.rights* forniscono sequenze di “figli” sintattici che si verificano prima e dopo il token: vuol dire che servono per individuare le relazioni di gerarchia.

```
analizzato.write("\nITERATING AROUND THE LOCAL TREE\n")
analizzato.write("TEXT    DEP    N_LEFTS    N_RIGTHS    ANCESTORS\n")
root = [token for token in doc if token.head == token][0]
subject = list(root.lefts)[0]
for descendant in subject.subtree:
    assert subject is descendant or subject.is_ancestor(descendant)
    analizzato.write(descendant.text + '    ' + descendant.dep_ + '    ' + str(descendant.n_lefts) + '    ' + str(descendant.n_rights) + '
```

La Named Entity Recognition permette di riconoscere i “nomi” all’interno del documento: in particolare restituisce il testo originale, a che categoria appartiene e la posizione in cui si trova all’interno della frase.

```
analizzato.write("\nNAMED ENTITY RECOGNITION\n")
analizzato.write("TEXT   START   END  LABEL   DESCRIPTION\n")
for ent in doc.ents:
    analizzato.write(ent.text + '    ' + str(ent.start_char) + '    ' + str(ent.end_char) + '    ' + ent.label_)
```

Infine è stata realizzata la *Sentence Recognition,* ossia un riconoscimento delle frasi.

```
analizzato.write("\nSENTENCE RECOGNITION\n")
for sent in doc.sents:
    analizzato.write(sent.text + '\n')
```

4. Esempi di Output

Per dare un esempio di output consideriamo il primo tweet presente all'interno del documento:

President Obama just had a news conference, but he doesn't have a clue. Our country is a divided crime scene, and it will only get worse!

Eseguendo lo script che abbiamo in precedenza commentato si ottiene il seguente risultato:

************ANALIZING TWEET NUMBER 1...****************

LEMMATIZZTION

TEXT	LEMMA	POS	TAG	DEP
President	President	PROPN	NNP	compound
Obama	Obama	PROPN	NNP	nsubj
just	just	ADV	RB	advmod
had	have	VERB	VBD	ROOT
a	a	DET	DT	det
news	news	NOUN	NN	compound
conference	conference	NOUN	NN	dobj
,	,	PUNCT	,	punct
but	but	CCONJ	CC	cc
he	-PRON-	PRON	PRP	nsubj
does	do	VERB	VBZ	aux
n't	not	ADV	RB	neg
have	have	VERB	VB	conj
a	a	DET	DT	det
clue	clue	NOUN	NN	dobj
.	.	PUNCT	.	punct
Our	-PRON-	DET	PRP$	poss
country	country	NOUN	NN	nsubj
is	be	VERB	VBZ	ROOT
a	a	DET	DT	det
divided	divide	VERB	VBN	amod
crime	crime	NOUN	NN	compound
scene	scene	NOUN	NN	attr
,	,	PUNCT	,	punct
and	and	CCONJ	CC	cc
it	-PRON-	PRON	PRP	nsubj
will	will	VERB	MD	aux
only	only	ADV	RB	advmod
get	get	VERB	VB	conj
worse	bad	ADJ	JJR	acomp
!	!	PUNCT	.	punct

NOUN CHUNKS

TEXT	ROOT_TEXT	ROOT_DEP	ROOT_HEAD_TEXT
President Obama	Obama	nsubj	had
a news conference	conference	dobj	had
he	he	nsubj	have
a clue	clue	dobj	have
Our country	country	nsubj	is
a divided crime scene	scene	attr	is
it	it	nsubj	get

NAVIGATING THE PARSE TREE
TEXT DEP HEAD_TEXT HEAD_POS CHILDREN
President compound Obama PROPN []
Obama nsubj had VERB [President]
just advmod had VERB []
had ROOT had VERB [Obama, just, conference, ,, but, have]
a det conference NOUN []
news compound conference NOUN []
conference dobj had VERB [a, news]
, punct had VERB []
but cc had VERB []
he nsubj have VERB []
does aux have VERB []
n't neg have VERB []
have conj had VERB [he, does, n't, clue, .]
a det clue NOUN []
clue dobj have VERB [a]
. punct have VERB []
Our poss country NOUN []
country nsubj is VERB [Our]
is ROOT is VERB [country, scene, ,, and, get]
a det scene NOUN []
divided amod scene NOUN []
crime compound scene NOUN []
scene attr is VERB [a, divided, crime]
, punct is VERB []
and cc is VERB []
it nsubj get VERB []
will aux get VERB []
only advmod get VERB []
get conj is VERB [it, will, only, worse, !]
worse acomp get VERB []
! punct get VERB []
! PUNCT []

ITERATING AROUND THE LOCAL TREE
TEXT DEP N_LEFTS N_RIGTHS ANCESTORS
President compound 0 0 [u'Obama', u'had']
Obama nsubj 1 0 [u'had']

NAMED ENTITY RECOGNITION
TEXT START END LABEL DESCRIPTION
Obama 10 15 PERSON

SENTENCE RECOGNITION
President Obama just had a news conference, but he doesn't have a clue.
Our country is a divided crime scene, and it will only get worse!

5. Script

```
import spacy

nlp = spacy.load('en_core_web_sm')
tweets = open("tweets.txt", "r")
indice = 0
for contenuti in tweets.readlines():
        indice += 1
        doc = nlp(unicode(contenuti, "utf-8"))
        analizzato = open("anallizati.txt", 'a')

        analizzato.write("**********ANALIZING TWEET NUMBER " + str(indice) +
"...**********\n")
        analizzato.write("LEMMATIZZTION\n")
        analizzato.write("TEXT   LEMMA   POS   TAG   DEP\n")
        for token in doc:
                analizzato.write(token.text + '  ' +  token.lemma_ + '  ' + token.pos_ + '  ' + token.tag_ + '  ' + token.dep_ + '\n')

        analizzato.write("\nNOUN CHUNKS\n")
        analizzato.write("TEXT   ROOT_TEXT   ROOT_DEP   ROOT_HEAD_TEXT\n")
        for chunk in doc.noun_chunks:
                analizzato.write(chunk.text + '  ' + chunk.root.text + '  ' + chunk.root.dep_ + '  ' + chunk.root.head.text + '\n')

        analizzato.write("\nNAVIGATING THE PARSE TREE\n")
        analizzato.write("TEXT   DEP   HEAD_TEXT   HEAD_POS   CHILDREN\n")
        for token in doc:
                analizzato.write(token.text + '  ' + token.dep_ + '  ' + token.head.text + '  ' + token.head.pos_  + '  ' + str([child for
child in token.children]) + '\n')

        analizzato.write("\nITERATING AROUND THE LOCAL TREE\n")
        analizzato.write("TEXT   DEP   N_LEFTS   N_RIGTHS   ANCESTORS\n")
        root = [token for token in doc if token.head == token][0]
        subject = list(root.lefts)[0]
        for descendant in subject.subtree:
                assert subject is descendant or subject.is_ancestor(descendant)
                analizzato.write(descendant.text + '  ' + descendant.dep_ + '  ' + str(descendant.n_lefts) + '  ' +
str(descendant.n_rights) + '  ' + str([ancestor.text for ancestor in descendant.ancestors]) + '\n')

        analizzato.write("\nNAMED ENTITY RECOGNITION\n")
        analizzato.write("TEXT   START   END   LABEL   DESCRIPTION\n")
        for ent in doc.ents:
                analizzato.write(ent.text + '  ' + str(ent.start_char) + '  ' + str(ent.end_char) + '  ' + ent.label_)

        analizzato.write("\nSENTENCE RECOGNITION\n")
        for sent in doc.sents:
                analizzato.write(sent.text + '\n')

        analizzato.write('\n\n\n')
        print('DOCUMENTO ANALLIZZATO')

analizzato.close()
tweets.close()
```