Trattamento Automatico del Linguaggio

Rachele Sprugnoli – rachele.sprugnoli@unicatt.it

Centro Interdisciplinare di Ricerche per la Computerizzazione
dei Segni dell'Espressione (CIRCSE)

UNIVERSITÀ
CATTOLICA
del Sacro Cuore

LINGUISTICA COMPUTAZIONALE

versus

TRATTAMENTO AUTOMATICO DEL LINGUAGGIO

Computational linguistics and natural language processing […] are sometimes used interchangeably to describe the field concerned with the processing of human language by computers

- **Computational Linguistics** is used to describe research interested in answering linguistic questions using computational methodology
- **Natural Language Processing** describes research on automatic processing of human language for practical applications

Bender, Emily M. 2016. "Linguistic Typology in Natural Language Processing". Linguistic Typology 20(3), 645-660.

MA…

Il computer, di per sé, **NON** conosce il linguaggio naturale!

Il **Trattamento Automatico del Linguaggio** (TAL) ha lo scopo di dotare il computer di conoscenze linguistiche, di creare macchine che capiscano (e addirittura riproducano) il linguaggio naturale, di sviluppare programmi che assistano l'essere umano in compiti (*task*) linguistici:

- riconoscimento automatico del parlato
- sintesi automatica della voce
- traduzione automatica
- analisi automatica del sentimento

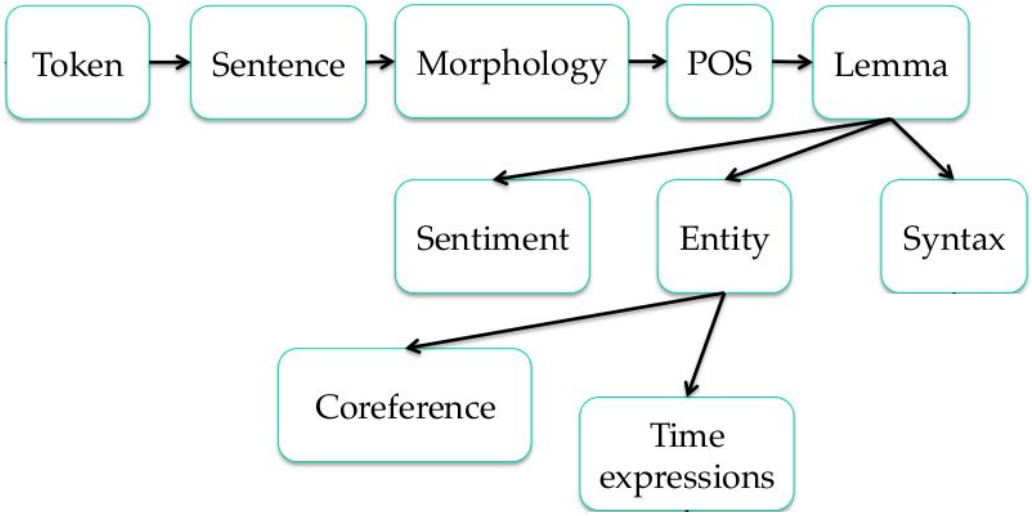
Google Translate

APPLICAZIONI

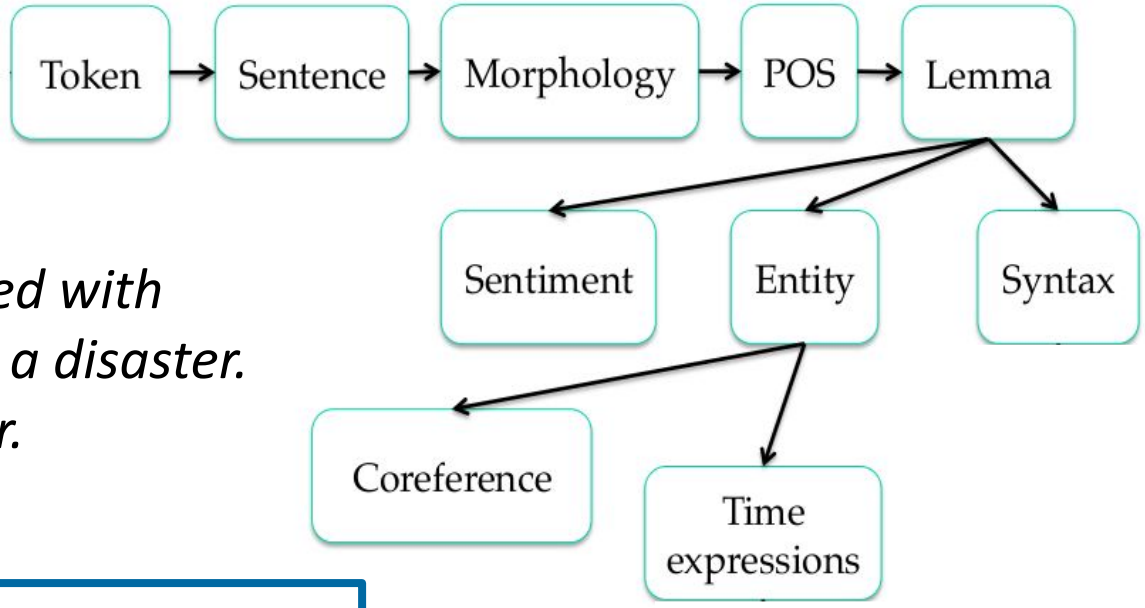
- BIBLIOTECHE ed EDITORIA: riconoscere autori/riferimenti bibliografici, individuare articoli pertinenti, suggerire percorsi di lettura, monitorare l'opinione dei lettori
- STORIA: estrarre eventi dalle fonti, individuare fonti su argomenti simili, migliorare la qualità dell'OCR per la digitalizzazione delle fonti
- LETTERATURA: identificare caratteristiche linguistiche e stilistiche
- MUSEI: generare in modo (semi-)automatico le descrizioni di opere, arricchire le descrizioni, identificare opere simili, creare percorsi museali personalizzati
  - https://pro.europeana.eu/project/ai-in-relation-to-glams
  - https://sites.google.com/view/ai4lam

COME ANALIZZARE IL LINGUAGGIO

- Struttura a **PIPELINE**: catena i cui moduli descrivono ognuno un diverso livello di analisi linguistica e dove l'output di un modulo diventa l'input per il modulo successivo. Esempio:



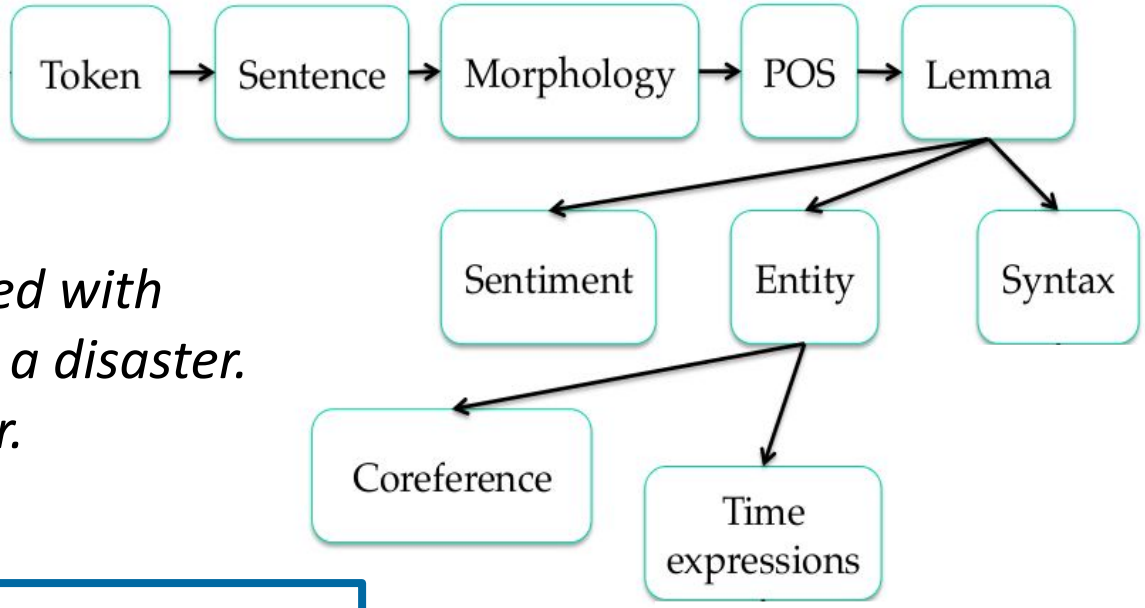
Le analisi presentate nelle prossime slide sono l'output della pipeline di Stanford CoreNLP
- demo online: http://corenlp.run/



When you see what happened with crooked Hillary today, it was a disaster. A disaster. She had a disaster.
Trump, 2016-08-05

TOKEN - SENTENCE - PART OF SPEECH

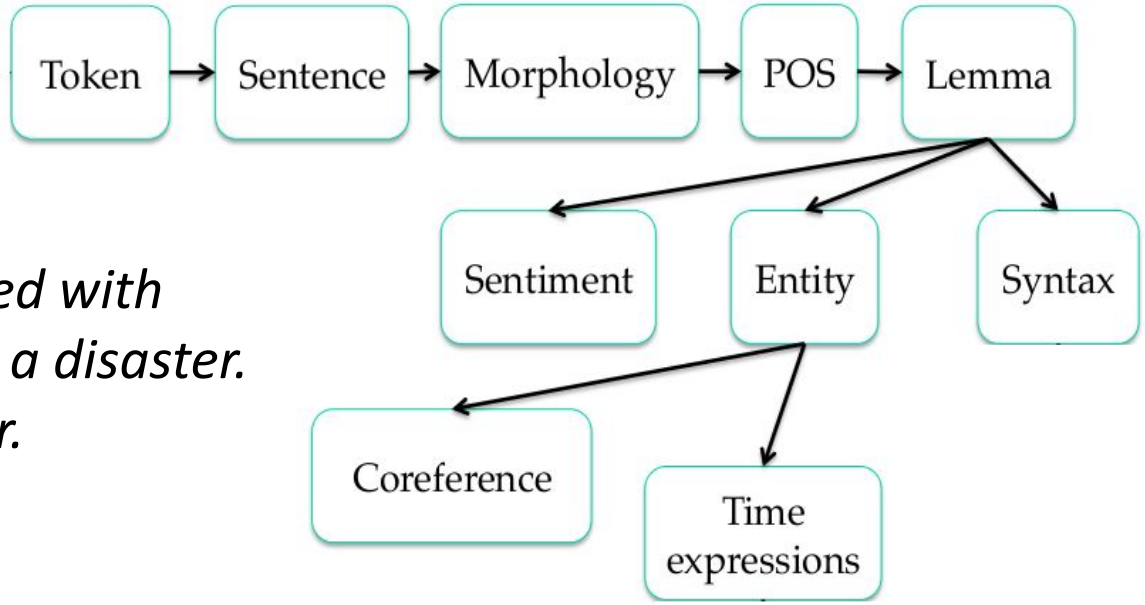
	WRB	PRP	VBP	WP	VBD	IN	JJ	NNP	NN	,	PRP	VBD	DT	NN	.
1	When	you	see	what	happened	with	crooked	Hillary	today	,	it	was	a	disaster	.
	DT	NN	.												
2	A	disaster	.												
	PRP	VBD	DT	NN	.										
3	She	had	a	disaster	.										



When you see what happened with crooked Hillary today, it was a disaster. A disaster. She had a disaster.
Trump, 2016-08-05

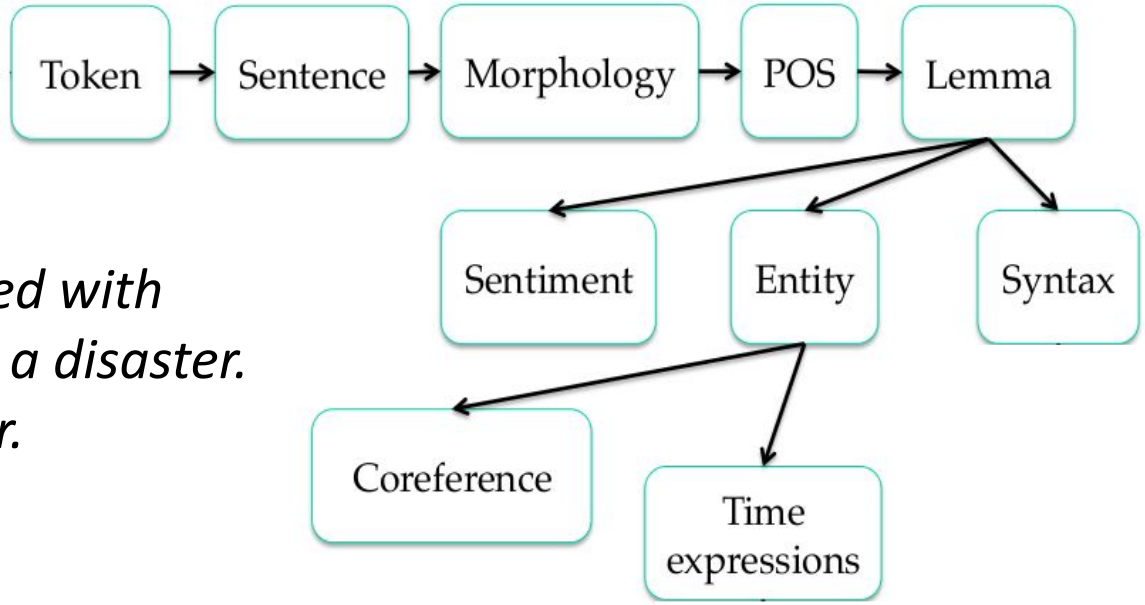
TOKEN - SENTENCE - PART OF SPEECH

C’era una volta un pezzo di legno.
C’era | una | volta | un | pezzo | di | legno.
C’ | era | una | volta | un | pezzo | di | legno | .



When you see what happened with crooked Hillary today, it was a disaster. A disaster. She had a disaster.
Trump, 2016-08-05

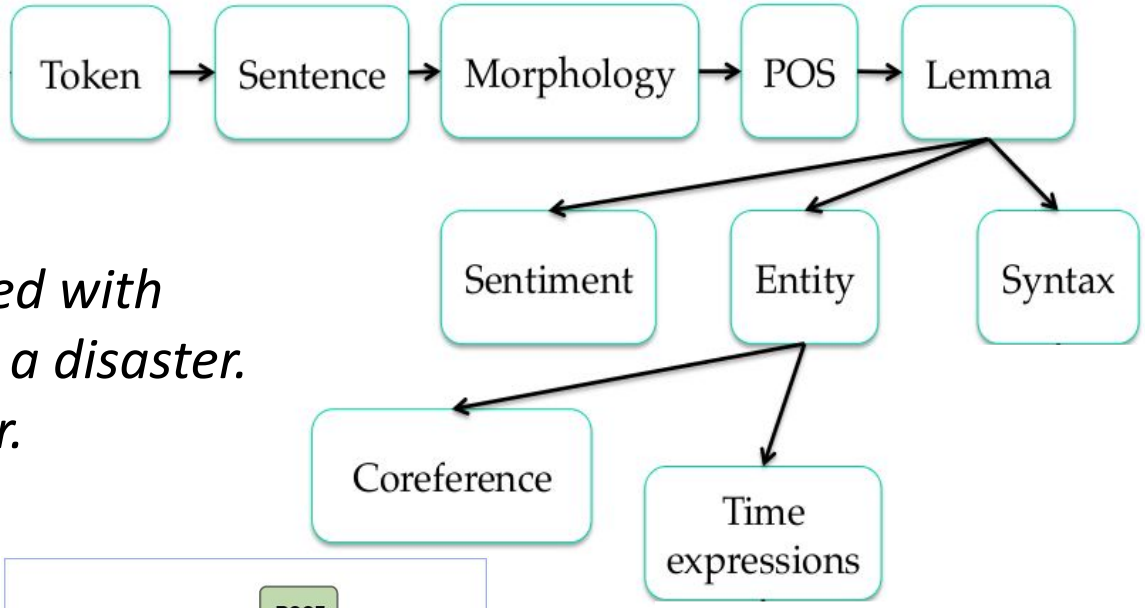
MORPHOLOGY



When you see what happened with crooked Hillary today, it was a disaster. A disaster. She had a disaster.
Trump, 2016-08-05

LEMMA

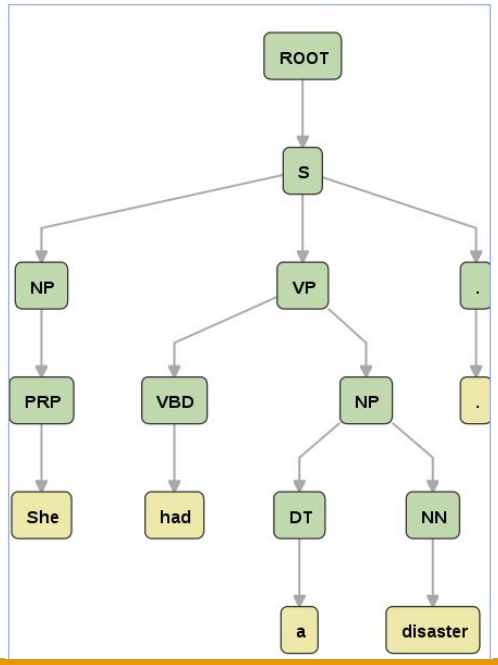
1	when you see what happen with crooked Hillary today , it be a disaster .	When you see what happened with crooked Hillary today , it was a disaster .
2	a disaster .	A disaster .
3	she have a disaster .	She had a disaster .

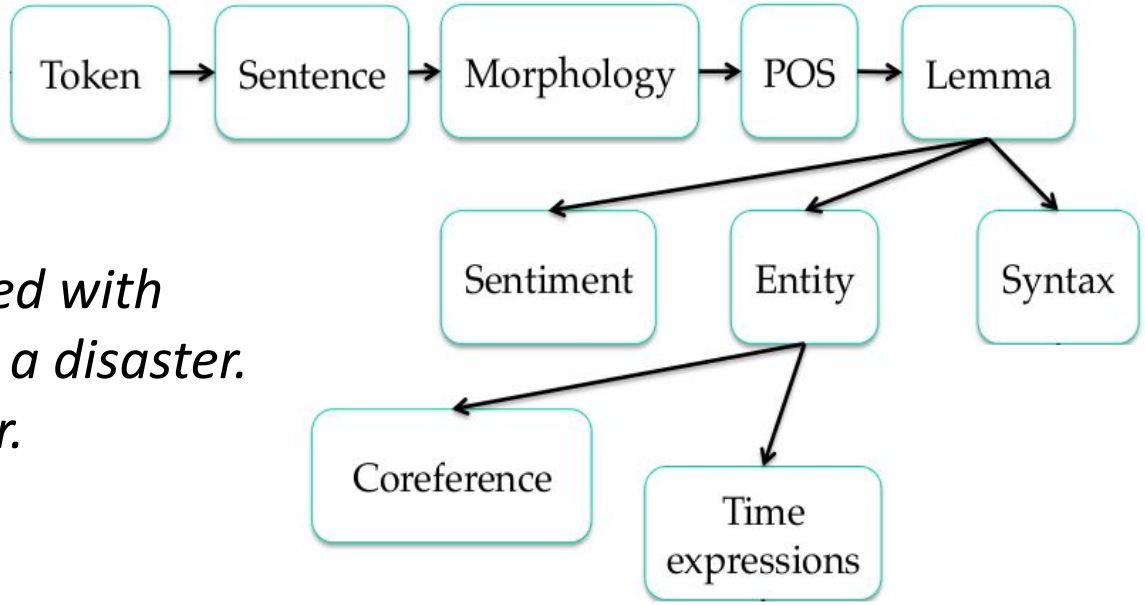


When you see what happened with crooked Hillary today, it was a disaster. A disaster. She had a disaster.
Trump, 2016-08-05

SYNTAX / PARSING

- a costituenti

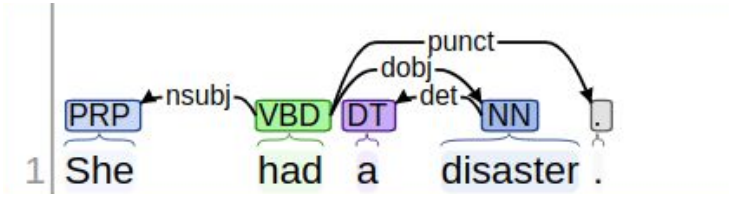


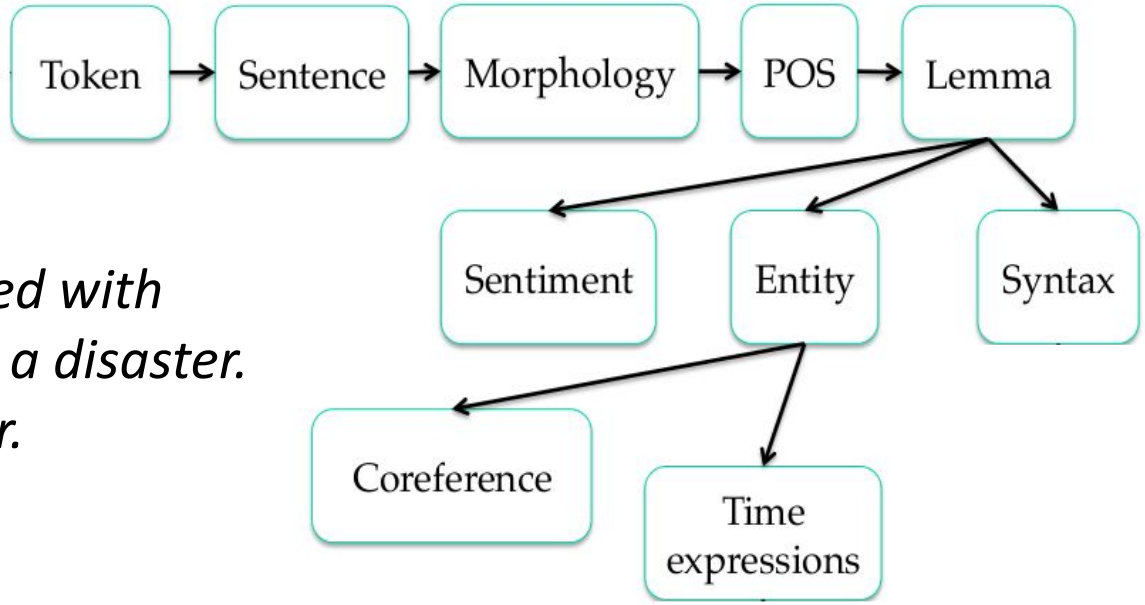


When you see what happened with crooked Hillary today, it was a disaster. A disaster. She had a disaster.
Trump, 2016-08-05

SYNTAX / PARSING

- a dipendenze



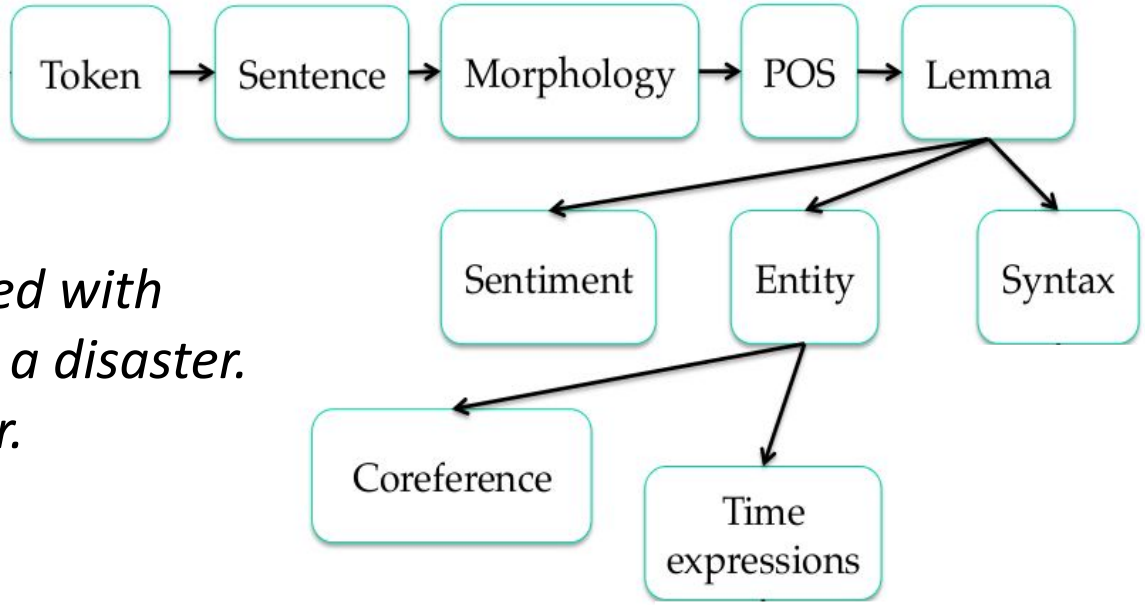


When you see what happened with crooked Hillary today, it was a disaster. A disaster. She had a disaster.
Trump, 2016-08-05

ENTITY

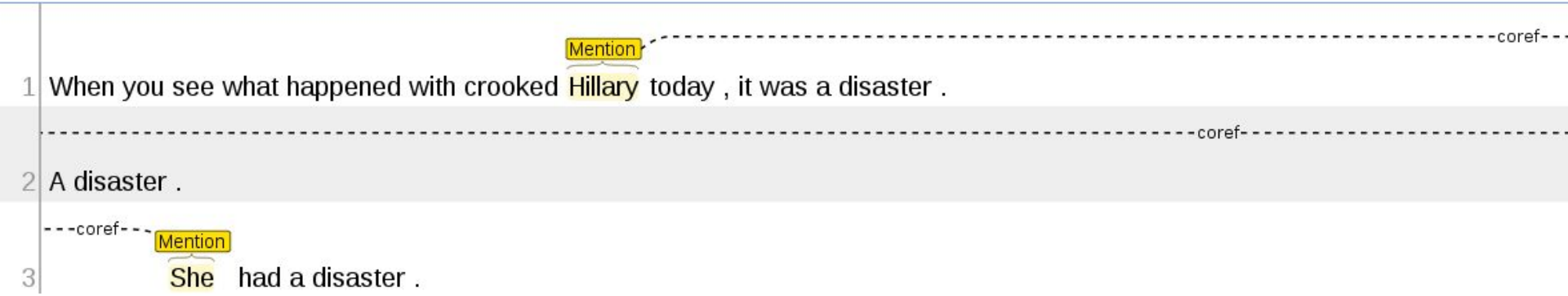
PER

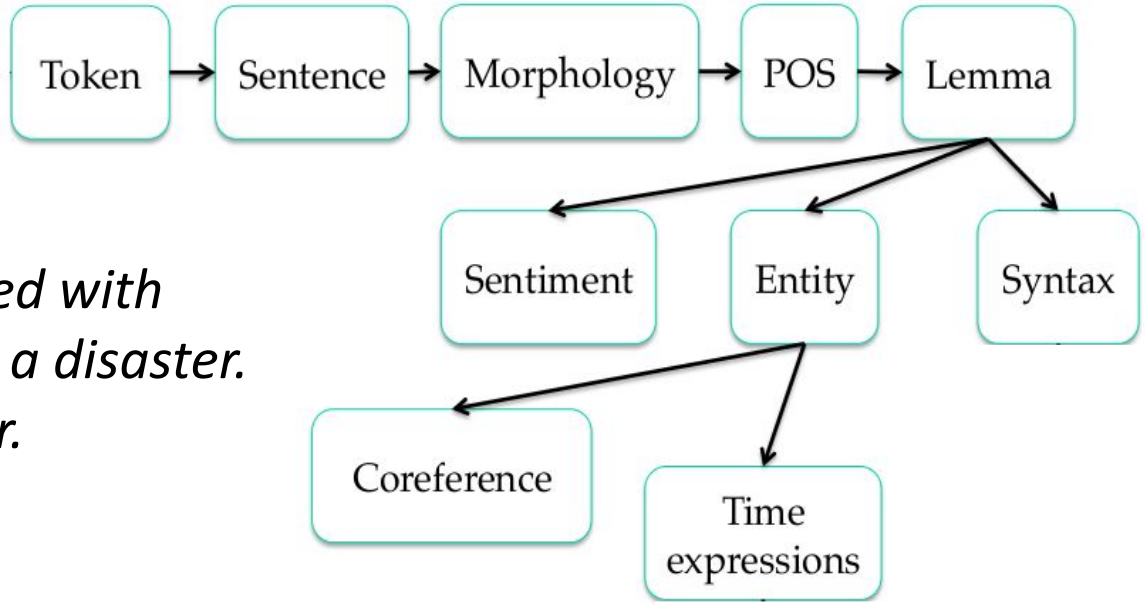
1 When you see what happened with crooked Hillary today , it was a disaster .
2 A disaster .
3 She had a disaster .



When you see what happened with crooked Hillary today, it was a disaster. A disaster. She had a disaster.
Trump, 2016-08-05

COREFERENCE



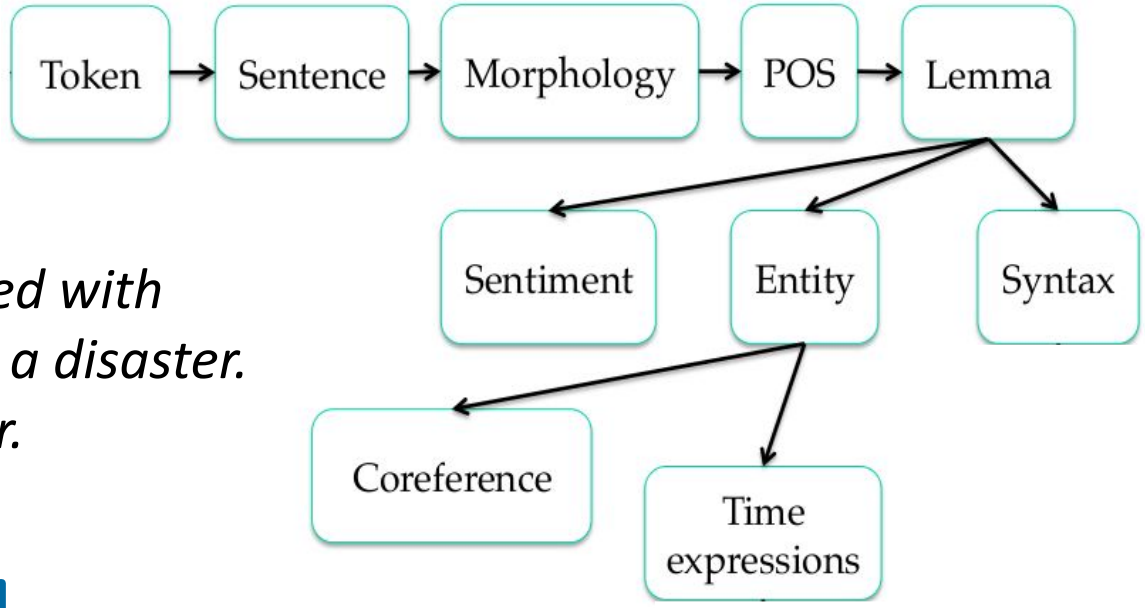


When you see what happened with crooked Hillary today, it was a disaster. A disaster. She had a disaster.
Trump, 2016-08-05

TIME EXPRESSIONS

2016-08-05

1 When you see what happened with crooked Hillary today , it was a disaster .
2 A disaster .
3 She had a disaster .



When you see what happened with crooked Hillary today, it was a disaster. A disaster. She had a disaster.
Trump, 2016-08-05

SENTIMENT

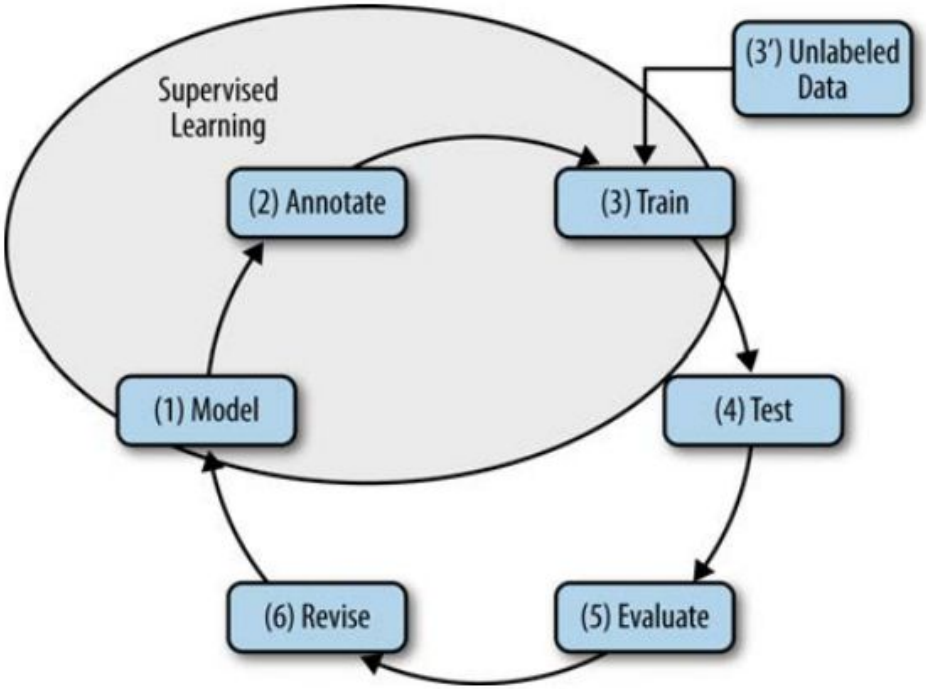
NEGATIVE
1 When you see what happened with crooked Hillary today , it was a disaster .

VERY NEGATIVE
2 A disaster .

NEGATIVE
3 She had a disaster .

COME SI SVILUPPA UN MODULO TAL

- **Sistemi di apprendimento automatico – MACHINE LEARNING (ML)**
- **ML SUPERVISIONATO**



Il ciclo MATTER

(Pustejovsky and Stubbs (2012) "Natural Language Annotation for Machine Learning". O'Reilly Media.)

COME SI SVILUPPA UN MODULO TAL

- **Sistemi di apprendimento automatico – MACHINE LEARNING (ML)**
- **ML SUPERVISIONATO**

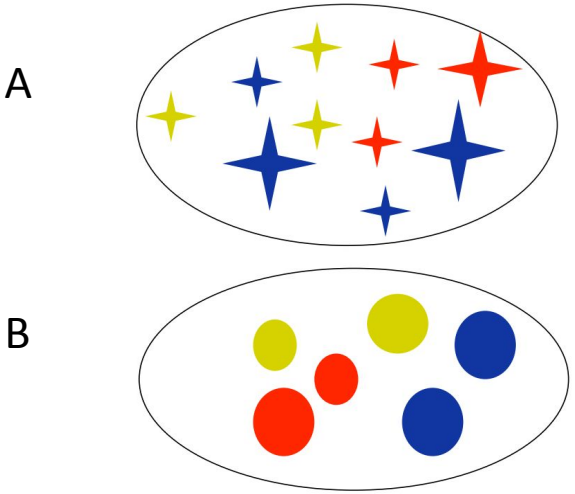
- Il ciclo MATTER**:**
  - **Model**: descrizione teorica di un fenomeno linguistico
  - **Annotate**: annotazione del corpus con uno schema di annotazione basato sul modello
  - **Train**: addestramento di un algoritmo di ML sul corpus annotato
  - **Test**: test del sistema addestrato su un nuovo campione di dati
  - **Evaluate**: valutazione delle performance del sistema
  - **Revise**: revisione del modello e dello schema di annotazione

COME SI SVILUPPA UN MODULO TAL

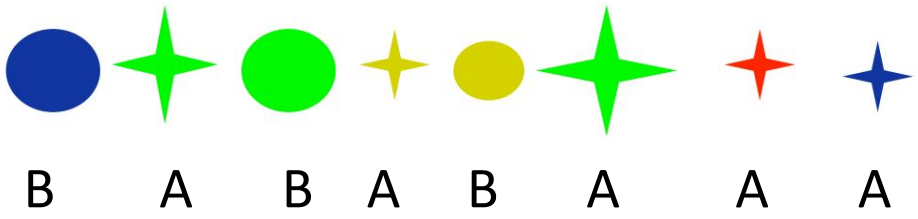
- **Sistemi di apprendimento automatico – MACHINE LEARNING (ML)**
- **ML SUPERVISIONATO, esempio**

- CLASSIFICAZIONE: dato un insieme di classi predefinite determinare a quale classe appartiene una certa entità

Input (training):



Classificazione di nuovi dati (test):



Tint - The Italian NLP Tool

- Pipeline addestrabile per i task di:

1) sentence splitting
2) tokenizzazione
3) PoS tagging
4) lemmatizzazione
5) analisi morfologica
6) dependency parsing
7) NER
8) analisi dei verbi composti
9) keyphrase extraction
10) analisi dei derivati
11) leggibilità

Sito web: **http://tint.fbk.eu/**

USIAMO TINT

- Apriamo il terminale
- Usare il comando `cd` per andare nella cartella `Tint` (che deve essere decompressa)
- Digitare il seguente comando poi premere Invio:

```
java -Dfile.encoding=UTF-8 -jar tint.jar -c
default-config.properties -i prova-news.txt -o
out-conll.conll -f conll
```

- Digitare il seguente comando poi premere Invio:

```
java -Dfile.encoding=UTF-8 -jar tint.jar -c
default-config.properties -i prova-news.txt -o
out-json.json
```

USIAMO TINT

- Leggiamo il comando

```
java -Dfile.encoding=UTF-8 -jar tint.jar -c
default-config.properties -i prova-news.txt -o
out-conll.conll -f conll
```

1. `java`: diciamo al computer che il programma è scritto in Java
2. `-Dfile.encoding=UTF-8`: specifichiamo l'encoding del testo in input: fondamentale per l'italiano!
3. `-jar`: specifichiamo estensione programma
4. `-c`: specifichiamo il file di configurazione
5. `-i`: specifichiamo il nome/percorso del file in input
6. `-o`: specifichiamo il nome/percorso del file in output
7. `-f`: specifichiamo il formato del file in output (default json)

`java -jar tint.jar -h`

L'OUTPUT DI TINT: JSON

- Apriamo il file `out-conll.conll` con un editor di testo
  - Che task include il formato `conll`?

- Apriamo il file `out-json.json` con un editor di testo
  - Che task include il formato `json`?

L'OUTPUT DI TINT: JSON

1	In	in	E	O	2	case
2	Italia	Italia	SP	LOC	5	nmod
3	"	[PUNCT]	FB	O	2	punct
4	la	la	RD	O	5	det
5	circolazione	circolazione	S	O	9	nsubj
6	dei	del	E+RD	O	7	case
7	virus	virus	S	O	5	nmod
8	influenzali	influenzale	A	O	7	amod
9	inizia	iniziare	V	O	0	ROOT
10	ad	ad	E	O	11	mark
11	intensificarsi	intensificare	V+PC	O	9	xcomp
12	"	[PUNCT]	FB	O	11	punct
13	e	e	CC	O	9	cc
14	si	si	PC	O	15	expl:impers
15	avvicina	avvicinare	V	O	9	conj
16	l'	l'	RD	O	17	det
17	inizio	inizio	S	O	15	dobj
18	del	del	E+RD	O	19	case
19	periodo	periodo	S	O	17	nmod
20	epidemico	epidemico	A	O	19	amod
21	.	[PUNCT]	FS	O	9	punct

L'OUTPUT DI TINT: JSON

COLONNE:

1) ID, identificativo numerico del token, riparte da 1 per ogni nuova frase. Le frasi sono separate da una riga vuota
2) token
3) lemma
4) PoS: https://www.corpusitaliano.it/static/documents/POS_ISST-TANL-tagset-web.pdf
5) Named Entity
6) ID della testa della parola nel parsing a dipendenze
7) etichetta della relazione a dipendenze: https://universaldependencies.org/u/dep/

L'OUTPUT DI TINT: JSON

- Lemma, PoS, morfologia,NER…

PoS tagset: https://www.corpusitaliano.it/static/documents/POS_IST-TANL-tagset-web.pdf


```
{
  "index": 7,
  "word": "virus",
  "originalText": "virus",
  "lemma": "virus",
  "characterOffsetBegin": 31,
  "characterOffsetEnd": 36,
  "pos": "S",
  "featuresText": "Gender\u003dMasc|Number\u003dPlur",
  "ner": "O",
  "full_morpho": "virus virus+n+m+plur virus+n+m+sing",
  "selected_morpho": "virus+n+m+plur",
  "guessed_lemma": false,
  "features": {
    "Gender": [
      "Masc"
    ],
    "Number": [
      "Plur"
    ]
  },
  "contentWord": true,
  "literalWord": true,
  "hyphenation": "vi-rus",
  "difficultyLevel": 4,
  "easyWord": true
},
```

L'OUTPUT DI TINT: JSON

- Verbi composti:
  - "è stata superata"

L'OUTPUT DI TINT: JSON

- Parole chiave

KD (Keyphrase Digger):
http://dhlab.fbk.eu:8080/KD_KeyDigger/


```
{
  "keyphrase": "incidenza",
  "frequency": 2,
  "score": 11.834,
  "idf": 1.0000000000751452,
  "score_boost": 1.0,
  "pattern_boost": 0.0,
  "chain_length": 1,
  "lemmas": [
    "incidenza"
  ],
  "stems": [
    "incident"
  ],
  "synonyms": [],
  "tokens": [
    "incidenza"
  ],
  "posList": [
    "S"
  ]
},
```

L'OUTPUT DI TINT: JSON

- Forme derivate

Informazione estratta dal **derIvaTario:** http://derivatario.sns.it/

Esempio: *influenzale*


```
"derivation": {
  "baseLemma": "fluire",
  "baseType": "presp",
  "phases": [
    {
      "affix": "2in",
      "allomorph": "in",
      "mt": "mt1",
      "ms": "ms2b",
      "type": "affixation"
    },
    {
      "convertionType": "v_a",
      "type": "conversion"
    },
    {
      "affix": "nza",
      "allomorph": "nza",
      "mt": "mt6",
      "ms": "ms2b",
      "type": "affixation"
    },
    {
      "affix": "ale",
      "allomorph": "ale",
      "mt": "mt1",
      "ms": "ms1",
      "type": "affixation"
    }
```

L'OUTPUT DI TINT: JSON

- Leggibilità

- **Level1:** 500 parole più facili
- **Level2:** 2500 parole più facili
- **Level3:** le 5000 parole più facili
- **TTR:** type/token ratio
- **Density:** #content words / #words
- **Deep*:** profondità albero sintattico

N.B. Parole tratte dal "Vocabolario di Base dell'Italiano" di De Mauro


```
"readability": {
  "level1WordSize": 19,
  "level2WordSize": 41,
  "level3WordSize": 47,
  "language": "it",
  "contentWordSize": 86,
  "contentEasyWordSize": 75,
  "wordCount": 145,
  "docLenWithSpaces": 909,
  "docLenWithoutSpaces": 769,
  "docLenLettersOnly": 746,
  "goodSentenceCount": 6,
  "sentenceCount": 6,
  "tokenCount": 168,
  "hyphenCount": 317,
  "hyphenWordCount": 141,
  "ttrValue": 0.78,
  "density": 0.593103448275862,
  "deepAvg": 4.833333333333333,
  "deepMax": 6.0,
  "subordinateRatio": 0.0,
  "deeps": {
    "0": 4,
    "1": 6,
    "2": 5,
    "3": 5,
    "4": 4,
    "5": 5
  }
```

L'OUTPUT DI TINT: JSON

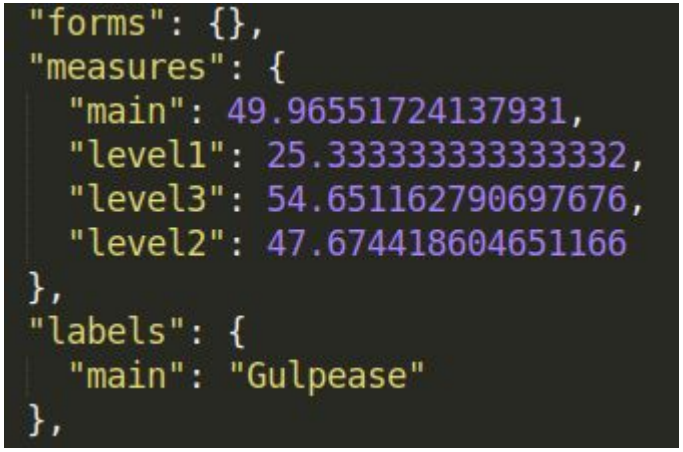
- Leggibilità:

- **Main = GULPEASE**, basato su numero frasi, lettere e parole. 100 massima leggibilità, 0 minima leggibilità

  < 80 difficile per chi ha la licenza elementare

  < 60 difficile per chi ha la licenza media

  < 40 difficile per chi ha un diploma superiore



N.B. Vale per l'italiano contemporaneo in prosa. Formula di Flesch per l'inglese.

L'OUTPUT DI TINT: JSON

- Leggibilità

- **Level1:** quanto è difficile per un lettore che conosce solo le 500 parole più facili dell'italiano
- **Level2:** quanto è difficile per un lettore che conosce solo le 2500 parole più facili dell'italiano
- **Level3:** quanto è difficile per un lettore che conosce solo le 5000 parole più facili dell'italiano


```
"forms": {},
"measures": {
  "main": 49.96551724137931,
  "level1": 25.3333333333333332,
  "level3": 54.651162790697676,
  "level2": 47.674418604651166
},
"labels": {
  "main": "Gulpease"
},
```


N.B Più il valore è basso, più è difficile.

L'OUTPUT DI TINT: JSON

- Statistiche sui POS tags

- Categorie granulari
- Macro-categorie

https://www.corpusitaliano.it/static/documents/POS_ISST-TANL-tagset-web.pdf


```
"posStats": {
  "support": {
    "CC": 4,
    "FF": 13,
    "A": 9,
    "B": 2,
    "E": 17,
    "DI": 1,
    "VA": 6,
    "FS": 6,
    "N": 10,
    "V+PC": 1,
    "RD": 13,
    "S": 34,
    "PC": 1,
    "V": 13,
    "E+RD": 13,
    "FB": 4,
    "SP": 21
  }
},
```

```
"genericPosStats": {
  "support": {
    "P": 1,
    "A": 9,
    "R": 13,
    "B": 2,
    "S": 55,
    "C": 4,
    "D": 1,
    "E": 30,
    "F": 23,
    "V": 20,
    "N": 10
  }
}
```

ALTRE PIPELINE

- UDPipe: https://lindat.mff.cuni.cz/services/udpipe/
- CoreNLP: https://corenlp.run/
  (info: https://stanfordnlp.github.io/CoreNLP/)
- LinguA: http://linguistic-annotation-tool.italianlp.it/

GRAZIE!

Email: rachele.sprugnoli@unicatt.it
Twitter: @RSprugnoli