Le parole della Costituzione

di Stefano Penge

0. Le domanda chiave

Come avrete letto nell'Introduzione, la nostra Costituzione è stata, ed è ancora, al centro di tante riflessioni educative. Si parte dalla Costituzione per ricostruire l'identità del cittadino futuro, e si torna alla Costituzione quando ci sono dei dubbi sui doveri e i diritti di quello presente.

In questo capitolo vogliamo proporre un'operazione un po' diversa: vogliamo usare gli strumenti digitali per interrogare la Costituzione. Vogliamo porle delle domande a cui sarebbe difficile rispondere con una lettura tradizionale. Per esempio, quanto è complessa la Costituzione? Di cosa parla? Quanto è leggibile oggi?

Per farlo, dobbiamo considerare la Costituzione non come una raccolta di concetti, ma come un documento concreto, come un testo da analizzare. Il fatto che i dati da cui partire siano parole, anziché numeri come nelle altre attività di questo volume, non deve trarci in inganno. Dato non significa solo numero; i numeri possono essere il risultato di un'operazione su qualsiasi parte del nostro universo, dagli insetti alle stelle, dai suoni alle parole. Esiste persino una branca della disciplina informatica, che si chiama in Italia *informatica umanistica*, che tratta proprio di queste operazioni.

Se consideriamo la Costituzione come un documento, prima di iniziare dobbiamo porci qualche domanda sulla legittimità del nostro lavoro. Sono domande che sembrano banali, ma non lo

sono mai quando si inizia un lavoro che parte da documenti di cui non siamo noi stessi gli autori.[1]

1. La Costituzione è un documento pubblico. Che significa esattamente "pubblico"? Gratis?
2. Chi è l'autore della Costituzione? Una sola persona o più persone? In questi casi, si applica la norma sul diritto d'autore?
3. Che operazioni si possono fare sul testo della Costituzione? Ci sono operazioni consentite e altre vietate?
4. Ne possono esistere versioni diverse, o solo l'originale fa fede?
5. La Costituzione vale solo se è scritta in Italiano? O si può tradurre in altre lingue?

Quanto è difficile la Costituzione?

Per capire un testo, come questo che state leggendo, bisogna conoscere il significato delle parole da cui è composto. Ma non basta: bisogna riuscire a collegare le parole tra loro e grazie ai meccanismi della grammatica e della sintassi capire il significato che si crea dalla relazione delle parole. Se un testo è rivolto a tutti gli Italiani, e vuole essere capito dalla maggior parte di questi, dovrà essere scritto in modo semplice, chiaro, senza inutili giri di parole, senza parole troppo lunghe e difficili.

Possiamo dire la stessa cosa della Costituzione? E come verificarlo?

La più famosa della analisi del testo della Costituzione dal punto di vista linguistico è quella di Tullio De Mauro, che nella sua "Introduzione alla Costituzione" cita delle cifre: la Costituzione è lunga 9369 parole, ma è scritta con soli 1.357 vocaboli diversi; di questi, ben 1.002 appartengono al

1 https://www.aib.it/attivita/2020/78571-pubblico-dominio-istruzioni-per-luso-frequently-asked-questions/

vocabolario di base e da soli occupano il 92,13% del totale. Inoltre, la lunghezza media per frase è inferiore alle 20 parole.[2]

Ma come sempre succede, una cosa è leggere delle affermazioni, anche se espresse da qualcuno di attendibile come il professor De Mauro (linguista emerito, prima che ministro dell'Istruzione), un'altra è arrivarci da soli, partendo da strumenti semplici e disponibili a tutti.

Per capire cosa dice De Mauro occorre sapere che cos'è il *vocabolario di base*,[3] e quale sia la lunghezza media di una frase in Italiano.

Per questo, una delle attività che proponiamo in questo capitolo sarà quella di approfondire cosa significa leggibile, come si fa a verificare se un testo è leggibile, e infine qual è la leggibilità della Costituzione.

Di cosa parla la Costituzione?

Beh, è facile: ci sono i titoli. Diritti e doveri, rapporti etico-sociali, rapporti economici, rapporti politici… Ma più in dettaglio?

Ci sono tante analisi dal punto di vista storico e giuridico della Costituzione che la spiegano e la interpretano, la confrontano con i documenti corrispondenti in altri Paesi (la Costituzione Indiana[4]) o in altre epoche (lo Statuto Albertino).

Ma non c'è il rischio di soggettività nell'interpretazione? Non potrebbe darsi che a seconda delle competenze, della cultura, dell'esperienza del lettore o semplicemente del momento storico in cui la si legge la Costituzione dica cosa diverse? In fondo, non si dice che il significato di un testo dipende dall'interpretazione soggettiva del lettore?

---

2 Ci sono molti altri studi sulla lingua della costituzione. Ad esempio si può consigliare e discutere insieme quello di Bice Mortara Garavelli, "L'italiano della Repubblica: caratteri linguistici della Costituzione" che non si limita ad analisi quantitative.

3 https://www.internazionale.it/opinione/tullio-de-mauro/2016/12/23/il-nuovo-vocabolario-di-base-della-lingua-italiana

4 Qui un confronto interessante tra la Carta Costituzionale indiana e quella italiana: https://losbuffo.com/2018/03/09/costituzione-italia-india/ ; invece su Wikipedia trovate una lista di costituzioni con l'anno di entrata in vigore e alcuni link per l'approfondimento: https://it.wikipedia.org/wiki/Lista_delle_costituzioni

Certo la Costituzione vuole essere esattamente il *contrario*: un testo che ha un'interpretazione unica, oggettiva, chiara per tutti. Ma ci riesce davvero?

Una delle possibilità offerta dall'informatica umanistica è proprio quella di scoprire gli argomenti di cui parla un testo attraverso la categorizzazione le parole. E' un po' quello che fanno i software che profilano gli utenti sulla base delle ricerche che fanno, dei siti che visitano, dei messaggi che inviano.

Una delle attività che proponiamo è perciò quella di provare a categorizzare i temi della Costituzione attraverso un'analisi delle frequenze delle parole e attraverso un'analisi delle concordanze, cioè delle presenza nel testo di coppie di parole particolarmente significative.

1. Raccogliere i dati

1.1 Procurarsi il testo

Visto che la nostra principale fonte di dati sarà la Costituzione, è il caso prima di tutto di procurarsela e *leggerla*. A partire dall'indice.

La Costituzione Italiana ha una struttura articolata in Parti, Titoli, Sezioni e Articoli.[5]

- PRINCIPI FONDAMENTALI (artt. 1-12)
- PARTE PRIMA: Diritti e doveri dei cittadini (artt. 13-54):
  - Titolo I - Rapporti civili
  - Titolo II - Rapporti etico-sociali
  - Titolo III - Rapporti economici
  - Titolo IV - Rapporti politici
- PARTE SECONDA: Ordinamento della Repubblica (artt. 55-139):
  - Titolo I - Il Parlamento
    - SEZIONE I - Le Camere
    - SEZIONE II - La formazione delle leggi
  - Titolo II - Il Presidente della Repubblica
  - Titolo III - Il Governo
    - SEZIONE I - Il Consiglio dei Ministri
    - SEZIONE II - La pubblica amministrazione
    - SEZIONE III - Gli organi ausiliari
  - Titolo IV - La magistratura
    - SEZIONE I - Ordinamento giurisdizionale
    - SEZIONE II - Norme sulla giurisdizione
  - Titolo V - Le Regioni, le Province, i Comuni
  - Titolo VI - Garanzie costituzionali
    - SEZIONE I - La Corte costituzionale
    - SEZIONE II - Revisione della Costituzione

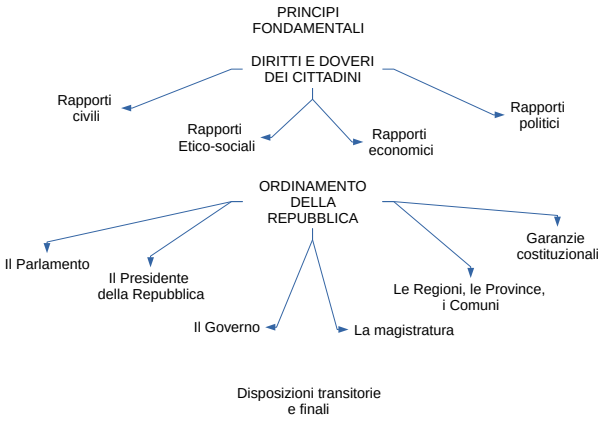
5 Nel seguito, per evitare fraintendimenti, useremo la forma maiuscola (Articolo) per differenziarla dagli altri significati della parola: articolo come *particella grammaticale* e come *testo giornalistico*.

    - Leggi costituzionali
- Disposizioni transitorie e finali

Probabilmente un diverso modo di visualizzarne la struttura aiuterebbe a capirla meglio. Possiamo provare a realizzare a mano un grafo, in cui i nodi sono le parti e i titoli (volendo, anche le sezioni) e gli archi hanno il significato di composto-da. In questo modo emergerebbe una struttura simmetrica che forse l'indice nascondeva.

Oppure, ma questo solo dopo una lettura analitica, potremmo provare a realizzare una *mappa* visuale in cui ogni articolo è un nodo e gli archi sono i collegamenti semantici, con il significato di vedi-anche.

Un altro tipo di visualizzazione potrebbe essere fatto a partire dalle parole, oppure dei temi generali, magari nella forma di "word cloud"; per costruire questo tipo di mappe, però, abbiamo bisogno di un supporto da parte del computer in termini di conteggio delle frequenze o di categorizzazione delle parole, e quindi le rimandiamo all'ultimo capitolo.



Per gli scopi di questa attività, ci limitiamo a studiare solo i primi 12 Principî e i seguenti 42 Articoli, cioè un totale di 54 Articoli. Questa scelta deriva da due considerazioni:

1. i primi 54 Articoli sono quelli che regolano nelle forme più generali i rapporti tra i cittadini e la Repubblica, e probabilmente sono i più rilevanti per l'Educazione Civica;
2. per lo stesso motivo, questi articoli sono omogenei per linguaggio, mentre nella seconda parte compaiono termini tecnici (come “giurisdizione”, “promulgare”, “impugnazione”) che pongono problemi ulteriori di leggibilità.

Si tratta però di una scelta nostra, che può essere modificata dal docente e dalla classe al momento della realizzazione concreta dell’attività.

1.2 Preparare il testo

Per lavorare sul testo della Costituzione in maniera automatica, per analizzarlo ed elaborarlo, dobbiamo averlo in un formato digitale e aperto, cioè leggibile da un software.

Non ci interessano gli aspetti tipografiche del testo (caratteri, colori, stili, margini) ma solo il contenuto, cioè le parole di cui è composto, intese proprio come sequenze di lettere, segni di punteggiatura e spazi.

In informatica non esistono le parole, ma i bit; però con i bit si possono rappresentare tante cose diverse, comprese le lettere dell'alfabeto. Ma in che modo? A quanti bit corrisponde una lettera? E in che ordine? A questa domanda l'informatica ha risposto definendo e pubblicando degli standard, cioè delle regole che stabiliscono appunto come devono essere disposti i bit per essere interpretati come lettere, o numeri, o immagini e così via: sono i formati.

Uno dei formati più semplici per i testi è il TXT,[6] che è una sequenza di simboli, ognuno dei quali corrisponde ad una lettera o una cifra, ad un segno di punteggiatura oppure ad un terminatore di riga. Si tratta di un formato *aperto*, di cui è nota la

6 Vedi Appendice

struttura, ed è leggibile da moltissimi software disponibili su qualsiasi sistema operativo. E' insomma il formato ideale per scambiarsi documenti testuali quando non sia tanto interessante l'aspetto ma solo il contenuto, come nel nostro caso.

Purtroppo non è facile trovare una versione in testo semplice della Costituzione: le versioni pubblicate sui siti istituzionali sono in PDF, che è un formato molto diffuso per rappresentare documenti, ma più complesso. Si può partire da una di queste versioni e trasformarla in TXT usando delle strategie artigianali (il "copia e incolla") oppure usando dei convertitori appositi. Nell'appendice trovate alcuni suggerimenti in proposito. In ogni caso, nella sezione "Risorse" del sito di questo testo trovate la versione in TXT della Costituzione già pronta per l'uso.

Una volta ottenuto il testo, può essere una buona idea “ripulirlo”. Che significa?

Prima di tutto assicuriamoci che non compaiano caratteri strani.[7]

Poi potremmo togliere gli a-capo inutili e gli spazi e tabulazioni all’inizio della riga. Forse noi umani non ci pensiamo, ma quelli che noi interpretiamo come *vuoti* sono comunque dei simboli come gli altri. Fanno parte degli accorgimenti tipografici che servono per facilitare la lettura; ma per un programma non sono significativi e quindi li possiamo togliere.

Infine c’è del numero dell’Articolo: se lo lasciamo, entrerà a far parte degli elementi lessicali da contare; se lo togliamo perdiamo la separazione tra Articoli nel caso di Articoli composti da più periodi. Una soluzione intermedia potrebbe essere quella di togliere tutti gli a-capo *tranne* quelli che separano gli Articoli

7 Che potrebbero derivare da un'errata codifica del file, soprattutto se è vecchio. Vedi Appendice

2 Elaborare i dati

2.1 Dimensione

Una volta ottenuto il documento in formato TXT, ci sono alcune operazioni semplici che possiamo fare aprendolo con un qualsiasi Word Processor, come Microsoft Word, o Libre Office Writer, o altro ancora.

Per prima cosa, possiamo *misurarne le dimensioni*. La nostra Costituzione è un documento relativamente *piccolo*. Se lo stampiamo su fogli A4, con un carattere corpo 11, un'interlinea singola, un margine di 2 cm a destra e sinistra, sta tutta in 20 pagine. Per essere il documento più importante su cui si basa tutta la vita politica e civile della Repubblica, non è davvero molto.

Ma quanto è lunga *in sostanza*? Ovvero, quanto è lungo il testo, indipendentemente dalle scelte tipografiche (cioè la dimensione della pagina, del carattere, interlinea, margini)? Possiamo saperlo immediatamente guardando in basso, nella riga bianca in fondo allo schermo, dove di solito sono presenti alcune informazioni generali sul documento aperto, tra cui appunto il numero di pagine, il numero di parole e il numero di caratteri,[8] cioè di lettere.

Questi numeri, combinati tra loro, ci dicono anche un'altra cosa: la lunghezza media delle parole è 6,8 lettere. Lo possiamo calcolare con una divisione:

```
numeroCaratteri / numeroParole
```

Questo valore è leggermente superiore alla media della lingua Italiana, che è circa 6.[9] Tiene conto di tutte le parole e di tutte le

8 La parola 'carattere' usata in questo contesto non c'entra con la definizione di un aspetto della personalità, ma si riferisce ai segni che corrispondono a lettere, numeri o segni di interpunzione. Esiste un altro significato di 'carattere' che è relativo allo stile grafico di questi segni; per non confondere i due significati, spesso per il secondo si usa il termine *font*, di derivazione tipografica.

9 Non c'è un valore ufficialmente riconosciuto. Dipende naturalmente dal tipo di testo, dal periodo storico, dallo stile. Potete fare la prova con i Promessi Sposi, che si scarica da Liber Liber, un archivio che contiene tutti i testi di pubblico dominio: https://www.liberliber.it/online/autori/autori-m/alessandro-francesco-tommaso-

lettere, ovunque si trovino: titoli, titoletti, articoli, preposizioni, numeri. Più avanti cercheremo un modo più preciso di definire la lunghezza media di una parola.

La media è un indicatore che ha un problema intrinseco: è rappresentativa solo per le distribuzioni simmetriche; quindi senza una misura della deviazione standard non è utilizzabile. Potremmo usare, invece della media, un indicatore diverso, come la mediana o la moda. Per far questo, però, non ci basta più il Word Processor ma dobbiamo passare ad altro.

Ci sono software appositi dedicati all'analisi lessicale,[10] ma sono esagerati per quello che ci serve. Un passo avanti possiamo però farlo usando semplicemente un foglio di calcolo.

Utilizzare un foglio di calcolo per trattare testi non è forse la prima idea che viene in mente, perché di solito si pensa che Excel (che è per molti sinonimo di 'foglio di calcolo', come Word lo è di 'programma di videoscrittura') sia uno strumento utile soltanto per trattare numeri. Ma se ci pensiamo un attimo, un foglio di calcolo contiene centinaia di funzioni, non solo matematiche ma anche statistiche, alcune delle quali sono applicabili anche ai testi.

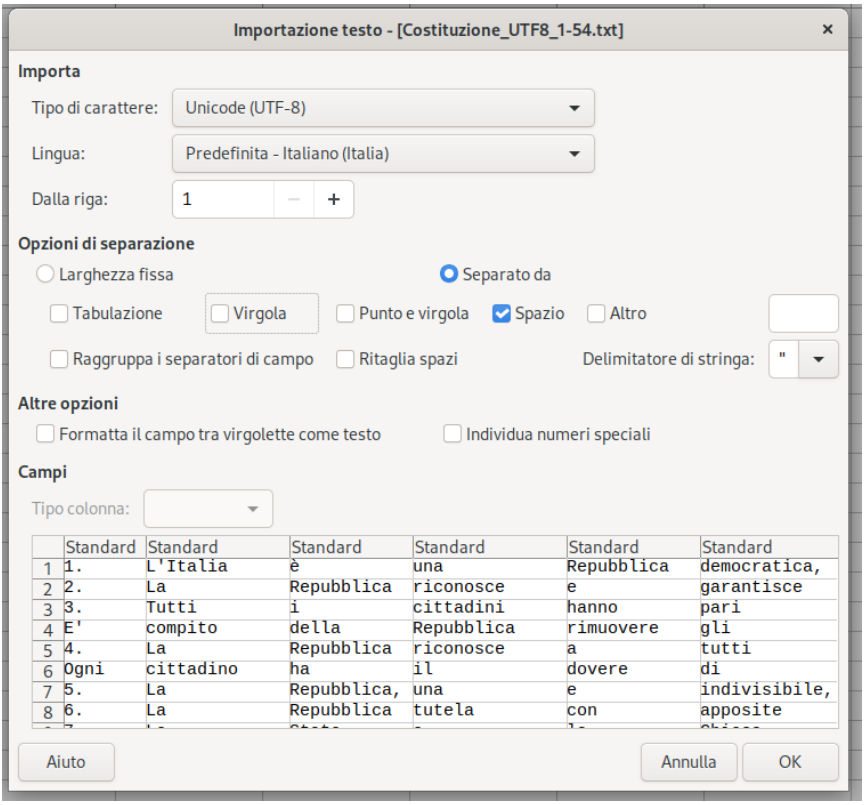
Possiamo cominciare con importare la nostra Costituzione_1-54.txt in un foglio di Libre Office Calc.[11] Al momento dell'importazione dobbiamo specificare la regola secondo la quale vogliamo dividere le parole e inserirle dentro le celle del foglio di calcolo. Per i nostri scopi, non ci serve inserire un intero articolo dentro una cella, ma abbiamo bisogno di qualcosa di più dettagliato: vogliamo che in ogni cella ci sia solo una parola. Allora usiamo come separatore di valore lo *spazio*. In questo modo otterremo una tabella in cui in ogni cella c'è una parola – o quasi.

---

manzoni/i-promessi-sposi-edizione-a-mondadori-1985/

10 Per esempio: https://korpus.cz/quitaup/ ; https://www.webtools.services/text-analyzer ; https://www.madeintext.com/text-analyzer/

11 Tutti gli esempi sono basati sui software della suite Libre Office, che è libera e gratuita; le stesse attività si possono eseguire utilizzando altre suite, come Microsoft Office nelle varie versioni, o altri fogli di calcolo disponibili sui diversi sistemi operativi.



Come vedete questa soluzione non è perfetta. Si vede che tra le parole figurano "L'Italia" e anche "democratica," che non sono parole. Dobbiamo fare un po' di pulizia, perché le celle contengono anche alcuni segni di punteggiatura: le regole di scrittura italiane vogliono che non ci sia nessuno spazio tra il segno e la parola precedente (questo non è vero in altre lingue, dove a volte si usa uno spazio sia prima che dopo). Quindi dobbiamo cancellare punti, punti e virgola, due punti e virgole. Fermi: non va fatto a meno, per fortuna possiamo usare la funzione "cerca e sostituisci e possiamo sostituire tutti questi caratteri che non ci servono con un valore nullo, il vuoto.

Aggiungiamo una colonna subito dopo quella con il numero dell'Articolo, e nella prima cella scriviamo questa formula:[12]

12 Nelle celle dei fogli di calcolo si possono inserire dati oppure formule. Per distinguere, le formule sono sempre precedute dal segno =.

```
=CONTA.VALORI(C2:FZ2)
```

Questo ci permette di avere il numero di celle che contengono qualcosa, dalla colonna C fino alla colonna FZ, che corrisponde al numero massimo di parole per Articolo.

Ricopiamo la formula per tutte le righe, oppure trasciniamo verso il basso la selezione.

Aggiungiamo una riga per le intestazioni; nella prima cella scriviamo "ARTICOLO" e nella seconda "PAROLE". A questo punto dovremmo avere una situazione come questa:

	A	B	C	D	E	F
1	ARTICOLO	PAROLE				
2	1	23	L'Italia	è	una	Repubblica
3	2	34	La	Repubblica	riconosce	e
4	3	74	Tutti	i	cittadini	hanno
5	4	48	La	Repubblica	riconosce	a
6	5	38	La	Repubblica	una	e
7	6	9	La	Repubblica	tutela	con
8	7	36	Lo	Stato	e	la
9	8	50	Tutte	le	confessioni	religiose
10	9	24	La	Repubblica	promuove	lo
11	10	69	L'ordinamento	giuridico	italiano	si
12	11	58	L'Italia	ripudia	la	guerra

In fondo agli Articoli, dopo la riga 55, possiamo inserire le formule per calcolare degli indicatori complessivi basandoci sulla colonna B: *somma, media, massimo, minimo e mediana*. Sono tutte funzioni già disponibili in Calc.

2747	somma parole
Per ogni Articolo:	
50,8703703703704	media
180	max
9	min
48	mediana

Come si vede, a parte alcuni estremi (l'Articolo 6 con 9 parole e l'Articolo 21 con 180 parole[13]) la distribuzione è piuttosto simmetrica, con la mediana e la media molto vicine.

Attenzione: siccome il file iniziale che abbiamo usato (Costituzione_1-54.txt) riportava ogni Articolo in una riga separata, e in più abbiamo eliminato i segni di interpunzione, non abbiamo modo di ragionare sulla lunghezza dei *periodi*, per quegli Articoli che sono più complessi (l'Articolo 21, appunto). Possiamo però rifare tutto il lavoro a partire da una versione della Costituzione in cui ogni periodo corrisponda ad una riga separata. Perdiamo in questo modo il riferimento agli Articoli originali, ma abbiamo la possibilità di misurare la complessità delle frasi. Vedremo più avanti che si potrà fare di meglio usando un linguaggio di programmazione, che è più flessibile e più manipolabile.

Per contare la lunghezza delle parole dobbiamo creare un nuovo foglio, che farà da specchio del primo. In pratica, in ogni cella del secondo foglio vogliamo mettere la lunghezza della cella corrispondente del primo foglio (cioè delle parole).

Potremmo pensare di usare questa formula:

```
=LUNGHEZZA($Foglio1.C2);
```

ma ci accorgeremmo presto di un problema: gli Articoli, come abbiamo appena scoperto, hanno lunghezze diverse; quindi nella parte finale di alcune righe ci sono delle celle vuote. Ma la lunghezza del valore di una cella vuota è 0, che è comunque un valore numerico. Quindi le funzioni statistiche che andremo ad applicare userebbero questo 0 come se fosse un valore.

Dobbiamo perciò usare una formula un po' più complessa che restituisce NULL nel caso in cui la cella corrispondente non contenga testo:

```
=SE(LUNGHEZZA($Foglio1.C2)>0;LUNGHEZZA($Foglio1.C2);
                                    "")
```

13 Perché questa differenza di lunghezza tra Articoli? Dipende dalla complessità delle materia dell'Articolo? Dal rischio di ambiguità? Porsi esplicitamente queste domande durante il lavoro con gli studenti è sicuramente l'aspetto più produttivo didatticamente di un lavoro su documenti.

che dice: *se la lunghezza della cella corrispondente è maggiore di 0, usa quel valore; altrimenti restituisci una stringa vuota* (cioè: "").

Copiamo la formula e incolliamola in tutte le celle del foglio. Apparirà in ogni cella la lunghezza della parola della cella corrispondente del foglio precedente. Inseriamo una nuova colonna per la somma delle lunghezze delle parole di ogni Articolo, con la formula:

```
=SOMMA(G2:GD2)
```

Dovremmo trovarci in questa situazione:

	A	B	C	D	E	F
1	ARTICOLO	SOMMA				
2	1	128	8	1	3	10
3	2	207	2	10	9	1
4	3	423	5	1	9	5
5	4	260	2	10	9	1
6	5	224	2	10	3	1
7	6	57	2	10	6	3
8	7	212	2	5	1	2
9	8	295	5	2	11	9
10	9	130	2	10	8	2
11	10	454	13	9	8	2
12	11	326	8	7	2	6

Se vogliamo, possiamo inserire delle nuove colonne per calcolare la lunghezza media, la mediana e la lunghezza massima delle parole per ogni singolo Articolo; oppure possiamo farlo per tutto il testo:

TOTALE LETTERE	15337
MEDIA	5,58318165271205
MAX	18
MEDIANA	5
DEV.STANDARD	3,42337531400691

Che significato hanno questi numeri?

Che la lunghezza media[14] delle parole è 5,58 (la mediana è 5), il che ci conforta perché è un po' più basso della media che avevamo indicato sopra. Ricordate: stiamo analizzando solo la prima parte della Costituzione.

Ci sono parole molto lunghe, addirittura 18 lettere (quali? basta andare a vedere la cella corrispondente del foglio 1) che portano in alto la media. Perché ci sono? Si potevano sostituire con sinonimi più corti?

E quante sono le parole lunghe il doppio della media, ovvero più lunghe di 11 lettere (ad esempio "democratica")? Basta contarle, o meglio farle contare al foglio di calcolo.

In una cella (ad esempio, nella D64) inseriamo questa espressione

```
>11
```

e in un'altra l'espressione:

```
=CONTA.SE(G2:GD55;D64)
```

cioè: *conta le celle i cui valori soddisfano l'espressione contenuta nella cella D64.*

Sono 147, su 2.747; ovvero il 5,35% delle parole sono più lunghe di 11 lettere.

La deviazione standard ci serve a capire la variabilità della distribuzione, cioè quante parole si collocano intorno alla media. Il valore trovato (3,4233...) ci dice che *non* c'è una grandissima dispersione dei dati.

Per verificare in maniera più empirica, possiamo provare a contare la parole più lunghe di 2 caratteri e più corte di 9. Abbiamo bisogno di due celle contenenti le espressioni (D64 e D65) e una nuova funzione:

```
=CONTA.PIÙ.SE(G2:GD55;D64;G2:GD55;D65)
```

Verifichiamo così che le parole con lunghezza compresa tra 2 e 9 sono oltre il 50% del totale.

---

14 In statistica quando si calcola la media di oggetti indivisibili, come appunto le parole, si preferisce dire che la media è "tra 5 e 6" perché non esiste una parola lunga 5,58 lettere. In questo caso riteniamo più utile riportare il risultato esatto perché ci permette di confrontare testi diversi. Grazie a Morena De Poli per la precisazione.

Tutti questi numeri diventano sensati se li confrontiamo con quelli che si ottengono con l'analisi di altri documenti, fatti esattamente con la stessa metodologia sperimentale. Possiamo cioè prendere, ad esempio:

- un articolo di giornale
- un capitolo di un libro di testo scolastico
- un capitolo di un romanzo italiano "classico"
- una sentenza del tribunale
- un decreto della Presidenza del Consiglio dei Ministri
- una circolare del Dirigente Scolastico

caricarli nel nostro foglio di calcolo e vedere quali numeri restituiscono le funzioni che abbiamo descritto sopra. Solo a quel punto potremmo dare un senso a indicatori come media, deviazione etc.

Un'ultima annotazione metodologica: il testo che abbiamo analizzato contiene evidentemente molte parole corte: articoli, preposizioni, congiunzioni; ma anche le voci verbali "è" (29 volte) e "ha" (7 volte). Dobbiamo considerarle oppure no?

Da un lato queste parolette contribuiscono ad abbassare la media della lunghezza, e non hanno un grande contenuto; dall'altro, contribuiscono a facilitare la lettura. Un testo costruito senza queste paroline sarebbe evidentemente poco leggibile. E' un argomento che riprenderemo più avanti a proposito dell'analisi della leggibilità.

Ma se vogliamo fare qualcosa di più, ci conviene lasciare il nostro foglio di calcolo e utilizzare un linguaggio di programmazione.

2.2 Frequenze

Probabilmente tutti i linguaggi moderni sono in grado di prendere un testo (cioè una sequenza di caratteri), dividerlo in corrispondenza degli spazi o dei segni di punteggiatura e costruire una lista di parole. A partire da quella lista si possono

fare molte cose: contare la lunghezza delle parole, ordinare la lista, escludere le parole troppo corte, eliminare i duplicati, etc.

Per consentire a tutti, anche a chi non abbia una grande pratica di programmazione, abbiamo scelto come ambiente di lavoro Snap!. Si tratta di un ambiente di programmazione visuale, a blocchetti, che assomiglia a (ed è in effetti derivato da) Scratch, ma che si è evoluto fino a diventare uno strumento che può accompagnare i ragazzi dalla primaria alla secondaria superiore, e nelle intenzioni dei progettisti anche all'Università.

Nell'area dedicata del sito web trovate un tutorial in Italiano su Snap!. Può essere il caso di dargli un'occhiata prima di iniziare a svolgere le attività che presentiamo di seguito; oppure consultarlo ogni volta che appaia una difficoltà. Qui supponiamo che la classe sia in grado di aprire Snap! in qualche maniera,[15] scegliere e agganciare dei blocchetti, crearne di nuovi ed eseguirli.

L'unica cosa che vogliamo dire qui è che Snap!, pur essendo un ambiente visuale - in cui cioè si usano dei blocchetti di varie forme per costruire un programma - non permette solo di fare animazioni e raccontare storie come il cugino Scratch, ma ha almeno due caratteristiche che lo rendono molto utile anche in un contesto più impegnativo come il notro. La prima è che permette di lavorare con le *liste*. Liste di cosa? Liste di numeri, lettere, suoni, parole, sprite… ma anche liste di liste.

Una lista è una cosa di questo tipo:

```
(mela,arancia,susina)
```

Una lista di liste è invece questo:

```
(mela,4),(arancia,7),(susina,6)
```

Le liste di liste assomigliano alle tabelle:

1	mela	4
2	arancia	7
3	susina	6

e si possono trattare allo stesso modo, scorrendole riga per riga. Per esempio: (arancia,7) è il secondo elemento della lista. E' però possibile usare questo particolare tipo di lista come un

15 Snap! è utilizzabile direttamente dentro un browser, ma è anche possibile scaricarlo e aprirlo sul proprio PC senza connessione a Internet.

dizionario, in cui se chiediamo la *chiave* (in questo caso, la parola) ci viene restituito il *valore* (qui la sua lunghezza). Vedremo che questa possibilità ci permetterà di interrogare il nostro archivio.

Le liste hanno anche il vantaggio di non essere limitate a righe e colonne, nel senso che ogni elemento di una lista può a sua volta essere una lista; quindi si possono avere tabelle a 3, 4, 5 dimensioni e così via.

Questo discorso può sembrare astratto, ma se facciamo un esempio con il testo della Costituzione diventa più chiaro.

Cominciamo con costruire delle coppie formate dalle parole con la loro lunghezza:

```
(l,1)
(Italia,6)
(è,1)
...
```

A questo punto mettiamo tutte queste coppie (che sono liste) in una lista più grande, che corrisponde al primo Articolo:

```
(1,(l,1),(Italia,6),(è,1),(una,3),(repubblica,10),
(democratica,11)(fondata,7),(sul,3),(lavoro,6))
```

È come se la tabella fosse fatta così:

	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
1	l	Italia	è	una	repubblica	democratica	fondata	sul	lavoro
	1	6	1	3	10	11	7	3	6

Questa modalità è molto comoda perché in una sola struttura di dati possiamo tenere sia il testo originale dell'Articolo, sia i risultati delle analisi.

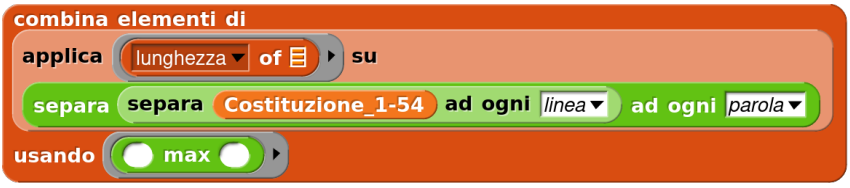
La seconda caratteristica particolare e molto utile di Snap! è che le funzioni messe a disposizione si possono applicare a *qualsiasi* elemento del linguaggio. Detto in altro modo, se abbiamo una funzione che conta, possiamo usarla sui numeri, ma anche sulle stringhe di caratteri, sulle liste, sui pixel di un’immagine o sui campioni di una registrazione audio.

In particolare, le liste si possono costruire e manipolare facilmente grazie a delle funzioni apposite che chiameremo 'blocchi magici'. Si tratta di blocchi che consentono di applicare una funzione a tutti gli elementi di una lista in una sola volta, per ottenere una nuova lista oppure un valore unico.

La prima cosa da fare è caricare il nostro file di testo Costituzione_1-54.txt dentro all'ambiente di programmazione. Nel caso di Snap!, è molto facile: basta prenderlo con il mouse (o con il dito se si usa un tablet; lo smartphone è un po' troppo piccolo) e trascinarlo nell'area di disegno. Viene creata automaticamente una variabile che contiene le 54 righe del testo, una per Articolo. Su questa variabile possiamo fare delle operazioni, come ad esempio:

1. creare una lista di Articoli
2. dividere ogni Articolo in parole
3. contare il numero di parole per Articolo
4. scoprire di quante parole consta l'Articolo più lungo.

Sembra lungo e complicato, e in effetti lo sarebbe se usassimo il foglio di calcolo. Il vantaggio di avere a disposizione un vero linguaggio di programmazione come Snap! è che tutte queste operazioni si possono fare con *un solo* blocchetto magico come questo:



Ovvero:

1. creare una lista di Articoli (blocco verde chiaro)
2. dividere ogni Articolo in parole (blocco verde scuro)
3. contare il numero di parole per Articolo (blocco arancione chiaro)
4. scoprire di quante parole consta l'Articolo più lungo (blocco arancione scuro con inserto verde)

Ogni volta che clicchiamo su questo blocchetto, appare in un fumetto il risultato (ovvero 133), che è il numero di parole dell'Articolo 33, che è il più lungo. Se vogliamo riusare questo blocchetto per altre analisi ci basta trascinarlo e agganciarlo ad altri blocchetti; oppure possiamo dargli un nome (per esempio: “trovaArticoloLungo”) e richiamarlo tutte le volte che ci serve.

Non entriamo nei dettagli di come funziona questo blocco magico; potete però ritornarci sopra più avanti, dopo aver visto qualche esempio più semplice.

Adesso possiamo fare qualche elaborazione più avanzata.

1. **Creare un elenco di tutte le parole della Costituzione**

Questo è davvero facile. C'è in Snap! una funzione giù pronta che crea una lista separando un testo ad ogni interruzione di parola.

In realtà, "separa" permette di dividere un testo usando come separatore tante cose diverse: uno spazio, un ritorno a capo, una virgola, una parola, o qualsiasi altro segno che ci venga in mente. È così che l'abbiamo usata sopra, per dividere la Costituzione in paragrafi ('linee'), per poi dividere ogni paragrafo in parole.

Peccato solo che in questo modo, per i motivi che abbiamo già visto prima, le parole contengano anche i segni di interpunzione, che in Italiano si attaccano alla parola precedente; e anche gli articoli e le preposizioni articolate con elisione. Quindi avremo tra le parole, ad esempio, "L'Italia", "religione,", "dall'autorità". Questo sarà un problema sia quando andremo a contare la lunghezza delle parole, sia in generale quando andremo a confrontare, per esempio, "autorità" con "dall'autorità".

Per ovviare a questo inconveniente creiamo un blocchetto nuovo, che chiamiamo 'Pulisci', che toglie i segni di interpunzioni alla fine delle parole e, se esiste un apostrofo, toglie quello e tutto quello che lo precede. Usare questo blocchetto ci pone però un dilemma che abbiamo già incontrato nell’attività precedente: se togliamo le virgole, non saremo più in grado di distinguere le frasi subordinate all’interno di un periodo, e se togliamo punti,

due punti e punti e virgole, non saremo in grado di separare i periodi se non tramite il carattere di a-capo. Dobbiamo fare una scelta, ed essere pronti eventualmente a tornare indietro.

Per adesso, questo blocchetto lo applichiamo su tutte le parole del testo e otteniamo una lista pulita, che possiamo usare subito, oppure metterla da parte in modo da ritrovarcela disponibile e farci altre operazioni: possiamo creare una variabile globale a cui diamo il nome 'parole'.

Il contenuto del blocco pulisci lo potete trovare nell'area relativa del sito web, insieme a tutti gli altri.

2. Misurare il testo

Già che ci siamo, vogliamo misurare la lunghezza totale del testo, sommando tutte le sue parole (una volta pulite, come detto sopra). Questo lo si fa con un solo blocchetto magico:

Come funziona? Il blocchetto magico 'combina' applica una funzione a tutti gli elementi di una lista, e man mano tiene traccia del risultato parziale, fino a restituire il totale. In questo caso, la funzione che vogliamo applicare ad ogni parola è la *somma della lunghezza* di ogni parola. Sono in effetti tre funzioni, una dentro l'altra.

3. Contare quante volte ogni lemma appare nel testo

Sondaggio veloce: nella Costituzione si parla più di diritti o di doveri? Qual è la parola che ricorre di più? Per rispondere

dobbiamo contare le *occorrenze*, cioè quante volte una forma viene ripetuta.

Prima di tutto dobbiamo creare un indice ordinato delle parole. L'algoritmo per creare un indice del genere è semplice:

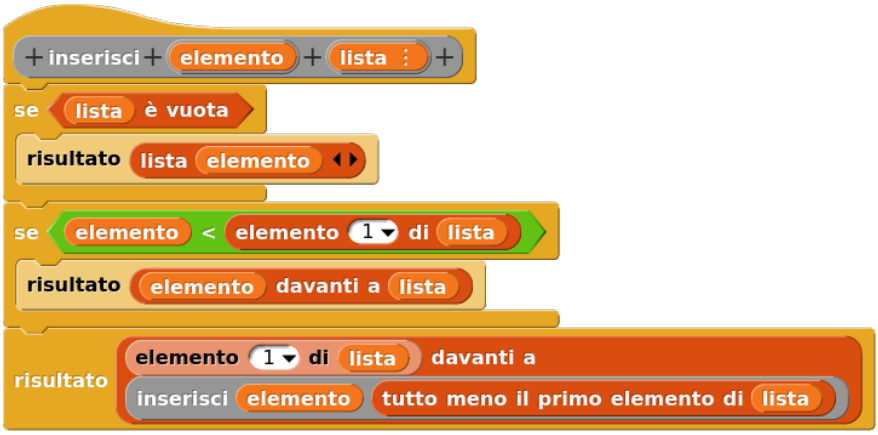
- si crea un indice di parole, cioè una lista vuota
- per ogni elemento X della lista "parole":
  - se l'indice è vuoto, si aggiunge la parola
  - se no, si inserisce la parola nella posizione alfabetica corretta

La parte più difficile è ovviamente l'inserimento della parola al posto giusto. La possiamo affidare ad un blocchetto apposito, che creiamo noi e chiamiamo "inserisci". Questo blocchetto (a differenza di quelli di Scratch, che sono sempre *procedure*, cioè comandi che fanno qualcosa, come un martello) è una *funzione*, ovvero restitusce sempre un valore alla fine della sua elaborazione. Un po' come una lavatrice in cui si inseriscono panni sporchi e sapone, e che alla fine restituisce biancheria pulita. A questo blocchetto dovremo passare due valori: la parola da inserire e la lista "indice". Il blocchetto sa fare tre cose:

1. se la lista è vuota:
   - ci mette dentro la parola e restituisce la nuova lista;
2. se la parola precede alfabeticamente il primo elemento della lista:
   - restituisce una nuova lista, costruita con la parola seguita dalla lista originale
3. negli altri casi:
   - ricomincia da capo ma con il resto della lista

In pratica, 'inserisci' divide un problema grosso (inserire una parola al posto giusto in una lista ordinata alfabeticamente) in due problemi più piccoli e risolvibili.

Più difficile a scrivere in Italiano che a fare Snap! ...



Questo modo di riusare un blocchetto *al suo interno* può sembrare illogico, ma è invece molto comodo per lavorare sulle liste, che sono strutturate proprio così: una testa con attaccata una coda, che è a sua volta una lista, che è fatta di una testa e una cosa, che a sua volta.. eccetera.

Il nostro algoritmo funziona bene, inserisce le parole al posto giusto, ma ha un solo difetto: è lentissimo. Per accelerarlo possiamo attivare la modalità 'turbo'[16] dal menù di Snap!, oppure possiamo inserire tutti i blocchetti dentro un blocco 'esegui con velocità turbo'.

Se abbiamo fretta, possiamo ordinare la lista delle parole tramite una funzione già pronta ('ordina') che non si limita a ordinare una lista di numeri, ma ordina *qualsiasi* lista sulla base di una funzione che decidiamo noi. In questo caso, la funzione di ordinamento è semplicemente X < Y, che applicata a delle parole lavora sull'ordinamento alfabetico.

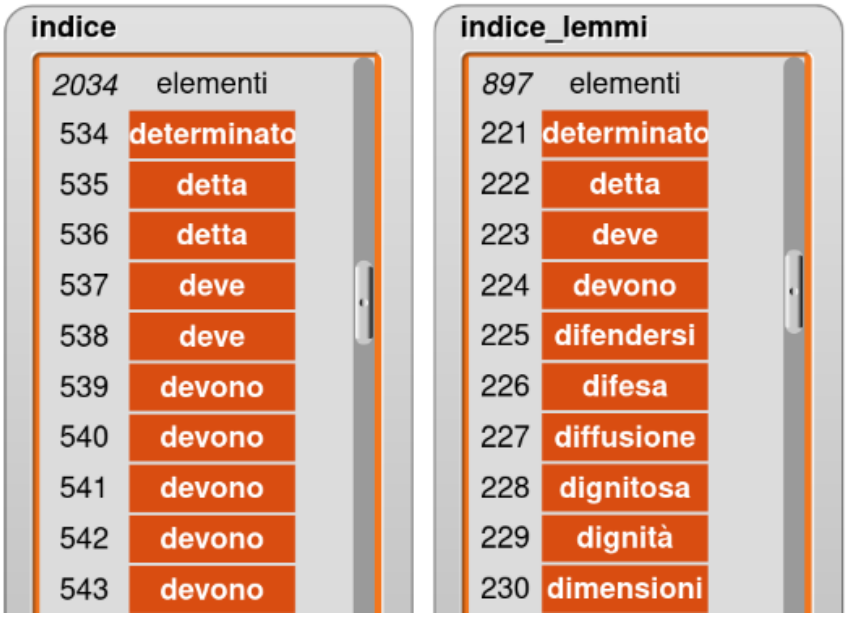
16 La velocità standard di esecuzione del codice da parte di Snap! permette di interromperlo premendo il bottone rosso in alto a destra in qualsiasi momento. È possibile rallentare ancora l'esecuzione in modo da vedere passo passo cosa succede mentre vengono eseguiti i vari blocchetti. Per scegliere questa modalità, si clicca sull'icona in alto con le orme dei passi e si sceglie una velocità di esecuzione spostando lo slider subito accanto.

Come avrete notato, la seconda lista è un po' più corta: il motivo è che abbiamo approfittato per togliere i numeri.

Attenzione al fatto che questo è l'elenco ordinato di *tutte* la parole della Costituzione, quindi può contenere – anzi contiene sicuramente – molte ripetizioni.

Possiamo eliminarle usando una funzione apposita:

ottenendo una lista più piccola, di 897 lemmi.



Ma prima di eliminare le ripetizioni, possiamo contarle per avere una nuova lista con le *frequenze* delle occorrenze delle parole. In generale, un algoritmo adatto potrebbe essere questo:

- si crea una lista di frequenze, che sarà fatta di tante piccole liste (parola, contatore)
- per ogni parola X dell'indice:
  - se non è nell'indice delle frequenze, si aggiunge una lista fatta così: (X, 0)
- se è già nell'indice delle frequenze, si incrementa il contatore relativo di 1
- alla fine si ordina l'indice delle frequenze sulla base del contatore, cioè del secondo elemento di ogni lista

E' senz'altro un esercizio interessante; ma anche in questo caso se non vogliamo perdere troppo tempo a costruire blocchetti possiamo usare una funzione già pronta che si chiama 'analizza' che fa esattamente la stessa cosa, senza errori e molto più velocemente:

Il risultato è una nuova lista di liste che mettiamo nella variabile "frequenze":

frequenze

897	A	B
234	direttamente	1
235	diretto	1
236	diritti	10
237	diritto	27
238	disciplina	2

Scorrendo la lista a mano possiamo verificare che nella Costituzione si parla più di diritti che di doveri; ma è più comodo costruire un blocco apposta, come questo:

2	A	B
1	diritto	27
2	dovere	6

Il nome della funzione 'assoc' non deve trarre in inganno: non associa qualcosa a qualcos'altro, ma a partire da una lista (che si chiama 'frequenze') e una chiave (la parola "diritto") cerca nella prima colonna la chiave e, se esiste, restituisce il valore della secondo elemento. Questo tipo di tabelle, con una colonna che funge da chiave e l'altra da valore, si chiamano 'associative': di qui il nome.

In questo modo si fa evidente un problema: ci sono molte parole che hanno la stessa radice, ma diverse desinenze e che vengono contate come parole diverse, anche se fanno riferimento ad un lemma unico. Per esempio, "diritti" e "diritto" vengono

contati separatamente, mentre probabilmente nella nostra testa andrebbero sommati (37).

Questo argomento è un po' complesso, e lo affrontiamo più avanti, nel paragrafo dedicato alla leggibilità.

4. Lunghezza

Riassumendo: abbiamo costruito a partire dal testo originale diversi elenchi:

1. un elenco di tutte le parole, nell'ordine in cui compaiono ("parole")
2. un elenco ordinato delle stesse parole, meno i numeri ("indice")
3. un elenco ordinato delle forme diverse ("indice lemmi")
4. un elenco di queste forme con la frequenza con cui appaiono nel testo originale ("frequenze")

Sappiamo dire quanto è lunga la Costituzione, da quante parole diverse è composta, e che frequenza di distribuzione hanno. Se ora volessimo analizzarne la lunghezza, in modo da poter dire qual è la parola più lunga, o quante parole sono più lunghe di 11 lettere, o quante sono comprese tra 12 e 14?

Stavolta non c'è nulla di pronto, ma poco male. Faremo come prima: creeremo una nuova lista ('lunghezza') che in realtà è una lista di liste (parola, lunghezza). Poi:

- per ogni elemento X dell'indice:
  - contiamo la lunghezza della parola X (la chiamiamo L)
  - se l'indice delle lunghezze è vuoto, ci si aggiunge una lista fatta così (X, L)
  - se invece ci sono già degli elementi, si inserisce la lista (X,L) nella posizione corretta

La difficoltà relativa all'inserimento di un elemento al posto giusto l'avevamo già risolta sopra; in questo caso si tratta però di inserire ogni elemento (la parola insieme alla sua lunghezza) nella posizione giusta relativa alla *lunghezza*, non alla parola. Questo ci costringe a rivedere il nostro blocchetto 'inserisci' e a farne uno leggermente più complicato.

Una volta ottenuta la lista, possiamo giocarci in vari modi. Per esempio possiamo chiederci quante e quali sono le parole di lunghezza compresa tra 12 e 14:

Letteralmente: applichiamo una selezione di elementi (blocco arancione chiaro) alla lista 'lunghezza' usando la condizione logica 'e' (blocco verde scuro) tra due applicazioni dell'operatore aritmetico 'maggiore' (blocco verde chiaro).

Al risultato, applichiamo la funzione 'lunghezza' (blocco arancione scuro).

3. Significati

Di cosa parla la Costituzione? E in generale, come si fa a capire in maniera automatica, cioè usando un computer, di cosa parla un testo?

E' un problema molto, molto importante di cui vediamo in continuazione applicazioni. Quando scriviamo su Whatsapp o su Facebook, quando un motore di ricerca indicizza un sito web, riuscire a capire di cosa si parla significa poter categorizzare quel testo e, in ultima analisi, profilare il suo autore. Siccome le lingue naturali sono molto complesse, questa operazione è affidata sempre di più a tecniche di intelligenza artificiale, da quella tradizionale (che usa conoscenze di tipo linguistico) a quella più alla moda oggi (che si limita a cercare somiglianze con testi già categorizzati).

Noi qui però non vogliamo affatto usare strumenti meravigliosi ma già pronti: vogliamo *usare* questo problema per capire meglio come funziona un testo, per leggerlo in maniera diversa da come si fa di solito.

3.1 Temi

Come hanno fatto gli archeologi a capire cosa dicevano i testi scritti in lingue sconosciute? Sono partiti da alcune parole note e hanno provato per induzione a indovinare il senso complessivo. Se dovessimo insegnare ad un bambino, o ad un marziano, come si fa a capire di cosa parla un testo in una lingua che non conosce, come faremmo? Gli diremmo, probabilmente, che certe parole particolare sono indicatori di un tema più generale. Se compaiono i termini 'donna', 'famiglia', 'eguaglianza', 'matrimonio', 'maternità', 'coniugi', probabilmente si parla del tema *Diritti legati alla differenza di genere*. Questo approccio, molto ingenuo, si chiama "a parole chiave".

Probabilmente il procedimento più semplice da spiegare per estrarre gli argomenti da un testo è questo:

- si crea, per ogni tema, una lista di parole chiave

- si estraggono dal testo le parole (o le strutture più complesse, come i modi di dire, le espressioni idiomatiche)
- si contano le parole che appartengono alle diverse categorie

La prima difficoltà è rappresentata dalle espressioni complesse, come “presidente della repubblica”, che ha un significato a sé; non ha molto senso cercare separatamente "presidente" e "repubblica". Purtroppo non siamo in grado di individuarle, e dovremo accettare questa limitazione.

La seconda è legata al fatto che le parole chiave sono limitate ad una sola forma (per esempio, per i sostantivi il maschile singolare, per i verbi l'infinito), mentre le parole del testo non lo sono. Come fare a riconoscere le parole chiave?

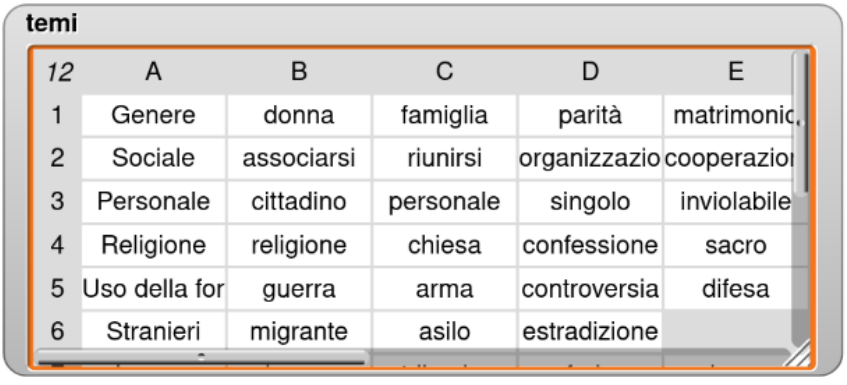
Ci scontriamo ancora una volta con questa caratteristica delle lingue naturali: *le parole hanno più forme*. Di nuovo, rimandiamo ad un paragrafo successivo il tentativo di soluzione. Qui suggeriamo una soluzione parziale e imperfetta: possiamo basarci su un calcolo di *somiglianza* tra le parole. Ci sono algoritmi che fanno proprio questo: prendono due sequenze dei lettere qualsiasi e ne calcolano la somiglianza, o da un punto fonetico (*soundex, metaphone*),[17] oppure dal punto di vista delle lettere che andrebbero cambiato in una parola per trasformarla nell'altra (*levenshtein*). In Snap! non esiste un blocco già pronto che applichi questi algoritmi, e costruirne uno per noi è troppo complesso. Però possiamo utilizzare una versione Javascript di un algoritmo che applica il metodo di Levenshtein – infatti Snap! è scritto in Javascript e può facilmente riusare codice sorgente scritto in quel linguaggio. Così potremmo confrontare ogni parola con una delle parole chiave cercandone non l'identità, ma la somiglianza.[18]

17 Questi algoritmi hanno una storia curiosa: sono stati inventati per correggere gli errori di inserimento dei nomi delle persone durante i censimenti

18 Negli esempi di codice sorgente trovate un blocchetto che fa esattamente questo. Attenzione: il codice Javascript usato non è particolarmente veloce.

Un primo elenco di temi e di parole chiave potrebbe essere questo:

- Genere
  - donna
  - famiglia
  - eguaglianza
  - matrimonio
  - maternità
  - coniugi
- Sociale
  - associare
  - riunire
  - organizzazione
  - cooperazione
  - collaborare
  - sociale
- Personale
  - cittadino
  - personale
  - singolo
  - inviolabile
- Religione
  - religione
  - chiesa
  - confessione
  - sacro
  - spirituale
  - fede
- Uso della forza
  - guerra
  - armi
  - controversie
  - difesa
  - militare
- Stranieri
  - migranti
  - asilo
  - estradizione
- Lavoro
  - lavoro
  - retribuzione
  - ferie
  - riposo
  - disoccupazione
- Diritto
  - diritto
  - libertà
  - potere
  - garantire
  - tutela
  - riconoscere
- Dovere
  - obbligo
  - tenere
  - vincoli
- Limiti
  - limite
  - anche se
  - in contrasto
  - responsabile
  - nell'ambito
  - salvo
- Umanesimo
  - scritto
  - storico
  - artistico
- Scienza
  - ricerca
  - tecnica
  - scientifico



Questa lista può essere preparata tramite un lavoro di gruppo. Ogni gruppo si prende cura di un tema e utilizzando un comune

vocabolario si appunta le parole che sembrano significative per quel tema.

Una volta costruita la lista possiamo realizzare una funzione che cerca, tra tutte le parole della Costituzione, quelle che appaiono nelle colonne da B in poi, e ne conta la frequenza:

Questa funzione si può applicare ripetutamente, sommando le frequenze di ogni parola chiave, per creare un report in cui per ogni tema vengano riportate le frequenze complessive.

per ottenere alla fine questo risultato, o uno simile:

indice_temi

13	A	B
1	Tema	Frequenza
2	Genere	17
3	Sociale	17
4	Personale	18
5	Religione	6
6	Uso della for	6
7	Stranieri	3
8	Lavoro	21
9	Diritto	57
10	Dovere	3
11	Limiti	11
12	Umanesimo	2
13	Scienza	4

Cosa si evince da questa tabella? Che, se abbiamo scelto bene in nostri indicatori, nella Costituzione italiana si parla molto più di diritti che di doveri, molto di lavoro, sociale e genere e meno di stranieri, più di scienza che di umanesimo.

A questo punto – altrimenti tutto il lavoro resterebbe un bell'esercizio di coding e basta - non resta che domandarsi: *perché*? Chi erano gli stranieri nel 1949? Qual era la situazione della donna nella società italiana dell'epoca? Che ruolo avevano le discipline scientifiche nella scuola riformata da Gentile?

3.2 Co-occorrenze

Una modo diverso di affrontare gli aspetti semantici è quello della ricerca delle *coppie di parole*. Si tratta di utilizzare la Costituzione (o meglio, una versione appositamente preparata) come un archivio a cui porre una domanda del tipo: "Quante volte la parola A appare insieme alla parola B?"

Non è semplice curiosità statistica: scegliere delle coppie di parole di cui andiamo a cercare la presenza ricorrente permette di scoprire se insieme costituiscono un tema. Si tratta di un'operazione classica dell'informatica umanistica. Ad esempio: quante volte, nella Divina Commedia, compaiono insieme i verbi della visione e i sostantivi della divinità? quante volte nell'opera

di D'Annunzio appaiono i pronomi personali e gli avverbi di tempo?[19]

Per *co-occorrenza* si può intendere la frequenza con cui due parole si trovano all'interno di una stessa frase; oppure la vicinanza tra due parole in termini di distanza di caratteri.

La prima versione richiede che si sia capaci di separare un testo in frasi. Che cos'è una frase? La grammatica parla di periodi e di proposizioni; ma noi per creare un programma che sappia dividere in frasi un testo abbiamo bisogno di un separatore che divida una frase da un'altra. Nel caso del periodo è piuttosto semplice: ci sono i vari tipi di punti (fermo, esclamativo, interrogativo). Nel caso della proposizione è più complesso perché la virgola, che pure divide le proposizioni, è usata anche per altri scopi, come gli elenchi, gli incisi, e altre forme; esistono anche altri simboli (punto e virgola, due punti, trattino, parentesi) che possono avere una funzione simile.

Noi però potremmo però sfruttare il fatto che la Costituzione è divisa in Articoli. Gli Articoli possono essere composti da un solo periodo, o più di uno (come il primo Articolo). Ogni periodo è di solito composta da una sola proposizione. Per i nostri scopi ristretti, potrebbe andare bene anche soltanto individuare gli Articoli in cui si parla insieme di A e di B.

Cominciamo con un'ipotesi, con un sondaggio tra i ragazzi: diritti e doveri sono temi connessi, nella Costituzione, oppure affidati ad Articoli diversi? In altre parole, secondo la Costituzione italiana i diritti hanno una controparte di dovere, oppure sono completamente indipendenti, tanto da meritare una trattazione separata?

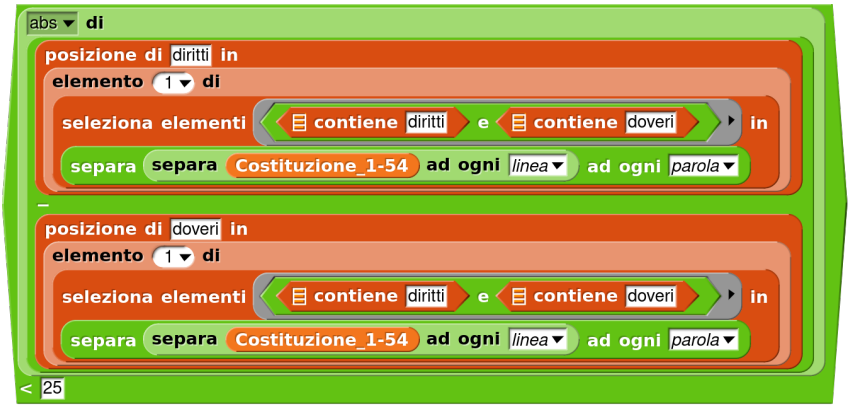
In questo caso sfruttiamo una funzione molto comoda di Snap!, che abbiamo visto più volte, che filtra una lista restituendo solo gli elementi che soddisfano una condizione. Possiamo perciò prendere tutti gli Articoli in cui è contenuta la parole "doveri" e la parole "diritti":

---

19 La più bella rappresentazione dei risultati di questo tipo di analisi che mi sia mai capitato di vedere è quella applicata ai nomi dei personaggi dell'Iliade: http://moebio.com/iliad/

Verifichiamo che ce n'è uno solo. In effetti, la parola "doveri" compare una sola volta nella Costituzione.

Se ci interessa anche sapere se la distanza tra le due parole è minore di un limite prefissato da noi (per esempio: 25) possiamo fare la differenza tra la posizione della prima parola ('diritti') all'interno della selezione e la posizione della seconda parola ('doveri'). Siccome non sappiamo quale delle due parole compare per prima, dobbiamo prendere il valore assoluto ('abs') di questa differenza e confrontarlo con il limite (25).



Un momento: stiamo cercando le sole parole 'diritti' e 'doveri', al plurale, e non anche 'diritto' e 'dovere' al singolare. Questo è sicuramente un limite grosso. Se vogliamo, possiamo costruire una funzione che non si limita a verificare se un elemento è identico ad un altro, ma se è uguale alla *prima parte* dell'altro. In questo modo, possiamo ricercare la sequenza di lettere "diritt" che va bene sia per "diritto" che per "diritti", e lo stesso per "dover".

Perché non cerchiamo semplicemente di vedere se una parola è contenuta in un'altra? Perché noi vogliamo trovare "diritto" a partire da "diritt", ma non "addirittura".

Questa nuova versione risponde correttamente che gli Articoli in cui si parla contemporaneamente di diritti e doveri sono quattro: il 2, il 30, il 48 e il 52.

La maniera di funzionare di "quasi contiene" è abbastanza semplice: prende la lunghezza della parola (N) e confronta le prime N lettere di ogni elemento della lista con la parola. Se uno di questi confronti ha successo, la funzione restituisce il valore 'vero'.

Naturalmente la co-occorrenza all'interno di uno stesso Articolo di 'diritti' e 'doveri' è solo una delle possibili ricerche che si possono fare con i ragazzi. Si può cercare 'donna' e 'famiglia', 'stranieri' e 'lavoro', 'libertà' e 'limite', eccetera.

4. Leggibilità

Se le competenze del gruppo lo consentono, se ne ha il tempo e l'interesse, si può continuare a interrogare la Costituzione in tanti altri modi, appena un po' più complessi dei precedenti. In quest'ultimo paragrafo affronteremo il problema della leggibilità della Costituzione; di passaggio, cercheremo di risolvere anche un problema a cui abbiamo accennato più volte: quello del riconoscimento automatico delle forme flesse dei lemmi.

Un testo è leggibile in funzione delle competenze del lettore e della sua padronanza dell'argomento; ma anche di alcune caratteristiche linguistiche del testo stesso. Esistono da tempo degli indici di leggibilità, specifici per le diverse lingue, che utilizzano alcuni parametri come:

- la lunghezza media delle parole (in lettere o in sillabe)
- la lunghezza media delle frasi (in parole)
- la lunghezza del testo (in frasi)

L'indice Flesch (1948), adattato da Roberto Vacca per l'italiano, propone questa formula:

$$206 - (0{,}6 \times S) - P$$

in cui S è il numero di sillabe sul totale delle parole e P è il numero medio di parole per frase.

L'indice GULPEASE (1988)[20] propone invece questa formula:

$$89 - (Lp / 10) + (3 \times Fr)$$

in cui Lp è uguale a:

$$(100 \text{ X totale lettere}) / \text{totale parole}$$

mentre Fr è uguale a:

$$(100 \text{ X totale frasi}) / \text{totale parole}$$

L'indice Gunning tiene esplicitamente conto della parole complesse (in termini di lunghezza in sillabe) oltre alla complessità delle frasi. Il punteggio – che corrisponde al numero di anni di scuola che è necessario per leggere quel testo – viene calcolato con questa formula:

---

20 Pietro Lucisano e Maria Emanuela Piemontese, *GULPEASE: una formula per la predizione della difficoltà dei testi in lingua italiana*, in «Scuola e città», 3, 31, marzo 1988, La Nuova Italia

0,4 x [(parole/frasi) + 100 (parole complesse/parole)]

Possiamo ottenere l'indice di leggibilità della Costituzione utilizzando un servizio online apposito.[21]

Ma perché l'attività abbia un senso didattico vero, in linea con l'impostazione generale di questo testo, invece di limitarci a rifare quello che altri hanno già fatto possiamo cercare arrivare un po' alla volta ad una *nostra* valutazione automatica della leggibilità della Costituzione, sulla base delle esperienze dei ragazzi e delle ipotesi che emergono dalla discussione. Ricominciamo dall'inizio.

Non si tratta qui solo di calcolare un indice, ma di *individuare* ed evidenziare le parole difficili da leggere, o le frasi troppo complesse sulla base delle competenze linguistiche dei ragazzi che partecipano all'attività, in modo che sia possibile intervenire su quelle, discutendone i significati o le possibili ambiguità, se ce ne sono.

Potremmo cercare di individuare gli elementi che (a nostro avviso) potrebbero rendere il testo più facile o più difficile. Si può fare un sondaggio in classe; probabilmente emergeranno alcuni indicatori. Ad esempio, un testo scritto è troppo difficile se:

1. ha molte parole difficili
2. ha molte frasi complesse

in cui resta da definire (con la ricerca e la discussione) cosa significa "molte", cosa significa "difficile" e cosa significa "complesso"; e soprattutto, quale peso relativo abbiano i due indicatori.

Il risultato del lavoro potrebbe essere qualcosa del tipo:

```
La Costituzione Italiana contiene 2801 parole.
Tra le parole, 41 sono troppo lunghe; 56 sono rare.
Tra le frasi, 12 sono troppo lunghe; 5 troppo complessa.
Queste sono le prime 10 parole difficili (lunghe o rare):
        1. …
```

21 Come questo: http://digilander.libero.it/RobertoRicci/variabilialeatorie/esperimenti/leggibilita.htm

```
        2. …
        3. …
  Queste sono le prime 10 frasi difficili (lunghe o
complesse):
        1. …
        2. …
        3. …
```

4.1 Difficoltà lessicale

Il primo livello di difficoltà di un testo è quello lessicale. Qui difficoltà di lettura può voler dire difficoltà di decodifica visiva, difficoltà a far corrispondere i simboli ai suoni e difficoltà a reperire la parola nel proprio "dizionario" mentale. Quali sono le parole difficili? Ad esempio:

- Parole lunghe
  - Parole composte o derivate da greco e latino (es. "metabolizzare")
  - Parole derivate (es. avverbi in -mente)
- Parole rare
  - Parole con ortografia "strana"
  - Parole che appartengono ad un lessico giuridico
  - Parole straniere

Cosa significa "parole troppo lunghe"? Su questo è utile aprire una discussione. Prendiamo le parole più lunghe della media? Oppure della lunghezza media per categoria (cioè: sostativi più lunghi della lunghezza media dei sostantivi)?

La lunghezza della parola non necessariamente è indice di difficoltà; le parole rare sono più probabilmente difficili da leggere.

Possiamo invece appoggiarci all'ortografia e possiamo considerare difficili le parole con troppe consonanti consecutive.

In italiano sono possibili da 1 a 3 consonanti consecutive. Se creiamo una funzione che estrae tutte le sequenze di consonanti lunghe 3 da una parola, possiamo usarla come misura di questo tipo di difficoltà di lettura.

Abbiamo ipotizzato che siano difficili da leggere per il lettore medio le parole appartenenti ad un vocabolario ristretto, tecnico. Di parole di questo tipo nella Costituzione ce ne sono probabilmente parecchie, soprattutto nella seconda parte, mentre nella prima - quella generale – non dovremmo trovarne molte. Possiamo farne un elenco e andarle a contare.

Per individuare le parole rare in maniera più precisa potremmo invece definirle per *esclusione*, partendo dal lessico delle parole più comuni, quelle che più o meno tutti i parlanti Italiano conoscono. Se una parola non è in quella lista, allora è rara – quindi difficile da leggere. Utilizzando i risultati di analisi statistiche di testi, sono stati realizzate diverse versioni di "vocabolario fondamentale"; quella più conosciuta oggi è quello stilato da De Mauro una prima volta nel 1980[22] e una seconda insieme a Chiari nel 2016, un anno prima della morte del grande linguista.[23] E' composta da circa 2.000 termini, che tutti i parlanti dell'Italiano conoscono, e che si stima costituire circa il 90% di ogni testo; poi ci sono insiemi sempre più grandi ("di alto uso", "di alta disponibilità") che tutti insieme compongono il vocabolario di base di circa 7.700 lemmi.

Se partiamo da questa lista di parole, possiamo contare quante parole del nostro "indice_lemmi" appaiono anche nel vocabolario di base, per verificare quello che diceva il professor De Mauro, e che abbiamo citato all'inizio dell'attività: che la Costituzione è

22 Tullio De Mauro, *Guida all'uso delle parole*, Editori Riuniti, Roma 1980 - 1997

23 Il vocabolario divisto è stato pubblicato da Internazionale nell'articolo citato più in alto, ma in PDF e in un formato pochissimo adatto all'uso automatico.

stata scritta usando parole comuni per essere leggibile da parte del maggior numero di persone.

Dobbiamo prima di tutto importare il vocabolario di base in Snap!.[24] Lo facciamo come prima, trascinando il file nvdb.csv nell'area del disegno.

Poi usiamo questo blocchetto magico:

Scopriamo così che sono solo 377, quasi un terzo. Ce ne aspettavamo di più, per un testo che si vuole comprensibile a tutti.

Perché così poche? Perché nella lista che abbiamo usato ci sono i lemmi, cioè i vocaboli nella forma *standard* in cui si trova in un vocabolario: i verbi all'infinito, i sostantivi al singolare maschile. Ma il nostro indice delle parole della Costituzione contiene invece le parole così come le abbiamo trovate: i verbi coniugati nei modi, tempi e persone, gli aggettivi concordati coi sostativi. Quindi molte parole non vengono riconosciute semplicemente *perché sono forme flesse*.

Ma come facciamo a riconoscere le forme flesse? Come facciamo a riconoscere che "lavorando" e "lavoravo" sono forme dello stesso lemma?

In sostanza: dovremmo essere capaci di estrarre le radici delle parole. Date le parole:

- ***lavor***o
- ***lavor***ando
- ***lavor***ò

bisogna poter estrarre:

- ***lavor***

24 Il file è scaricabile dal sito web del libro, in un formato già preparato per l'attività.

per categorizzare tutte e tre le parole sotto lo stesso lemma. Cosa che ci permetterebbe anche di raffinare il lavoro sul riconoscimento dei temi e sulle co-occorrenze.

Potremmo elencare *tutte* le forme di un nome (cioè singolare e plurale) e metterlo nella nostra lista. Possiamo, per ogni aggettivo, inserire la forma maschile e femminile, singolare e plurale. E di un verbo, possiamo inserire *tutte* le forme coniugate? E' vero che nella Costituzione non ci sono tante variazioni di tempi, modi e persone. Nella grande maggioranza dei casi, il modo è indicativo; il tempo è presente; la persona è la terza singolare o plurale. Un lavoro un po' lungo, ma che si può affrontare con la strategia di Cesare: divide et impera. Si divide la classe in gruppi e si assegna ad ogni gruppo una lista di lemmi: ogni gruppo dovrà restituire un nuovo elenco con tutte le varianti ragionevoli. Ad esempio:

- diritto: diritto, diritti
- riconoscere: riconoscere, riconosce, riconoscono
- libero: libero, libera, liberi, libere

eccetera.

Per fortuna abbiamo un'altra strada, che va in direzione opposta. Prendiamo dal vocabolario di base il lemma nella forma standard (ovvero: sostantivi e aggettivi al maschile singolare, verbi all'infinito presente), estraiamo la radice[25] e la andiamo a ricercare all'interno di ogni parola del testo. Se la radice è lunga N lettere, e se le prime N lettere della parole corrispondono alla radice, allora abbiamo (molto probabilmente) riconosciuto la parola per lo meno come una derivazione della radice.

Ad esempio:

lemma	**radice**	**lunghezza**	**parole riconoscibili**
adottare	*adott**	5	*adott*ato, *adott*a, *adott*ando, *adott*erà
adozione	*adozion**	7	*adozion*e, *adozion*i

25 Trascuriamo qui la vocale tematica e in generale la differenza in termini linguistici tra radice e tema: and-a-re, allegr-a-mente.

Il problema, l'avrete già indovinato, è che la desinenza da staccare per ottenere la radice è di dimensioni diverse a seconda della *categoria* grammaticale, e noi non sappiamo la categoria delle 898 parole che compongono la Costituzione.

Ma se usiamo il vocabolario di base, abbiamo la fortuna che all'interno di esso è riportata anche la categoria del lemma: verbo, sostantivo, avverbio, aggettivo, eccetera. Il file originale[26] si presenta così:

a s.f. e m.inv.
a prep.
abbagliante p.pres., agg., s.m.
abbaiare v.intr. e tr.
abbandonare v.tr.
abbandonato p.pass., agg., s.m.

ma con un po' di astuzia possiamo buttare le informazioni che non ci servono, riportare su una stessa riga tutte le categorie grammaticali possibili per ogni lemma e ottenere una versione un po' più comoda da usare:[27]

a	preposizione	sostantivo
abbagliante	aggettivo	sostantivo
abbaiare	verbo	
abbandonare	verbo	
abbandonato	aggettivo	sostantivo
abbandono	sostantivo	

Possiamo allora disegnare un blocchetto che in base alla categoria del lemma sia in grado di estrarne la radice, secondo una tabella di questo tipo:

26 Non quello originale (in PDF), ma quello elaborato dal benemerito Alberto Pettarini, che ha rilasciato il suo codice e i risultati su Github: https://github.com/pettarin/nvdb

27 Anche questa versione è disponibile sul sito web del testo.

categoria	numero di lettere da togliere	esempio
sostantivo	1	*adozion*-e
verbo	3	*adott*-are
aggettivo	1	*democratic*-o
avverbio modo	6	*democratic*-amente

Alcune righe della nostra tabella sono un po' traballanti (come l'ultima relativa agli avverbi), e richiederebbero un raffinamento; ma nei confini di un'attività didattica ci va bene così.

Questa tabella può essere interrogata da un blocchetto che, data una categoria grammaticale, restituisce il numero di lettere da togliere per ottenere la radice:

Il nostro algoritmo è in due passi:

1. prima creiamo un nuovo vocabolario che contenga però le radici, oltre ai vocaboli
2. poi confrontiamo le parole con le radici.

Un esempio del nostro *nuovissimo* vocabolario di base:

Vocabolo	Radice
adozione	adozion
adottare	adott
democratico	democratic
...	...

Ora possiamo costruire un blocchetto che, data una parola del vocabolario di base, restituisce la radice

Con questo nuovo blocchetto possiamo cercare le parole "assicura", "assicurarne", "assicurati", "assicuri" (che sono tutte presenti nel nostro indice di parole, dal numero 65 al 69) e scoprire che sono tutte varianti dell'unico lemma "assicurare".

E finalmente possiamo dire che negli Articoli da 1 a 54 della Costituzione il numero di parole appartenenti o riconducibili al Nuovo Vocabolario di Base è 811 su 897. Un bel risultato, no?

Questo algoritmo è lontano dall'essere perfetto, perché non ha modo di riconoscere le forme verbali che derivano da un tema diverso da quello principale, cosa che accade con molti verbi antichi. Per esempio, "andare" ha due temi: and- e vad-, ma noi siamo in grado di riconoscere solo le forme che derivano dal primo tema (quello dell'infinito). Per superare questa difficoltà, possiamo aggiungere a mano questi (pochi) verbi al vocabolario_radici.

4.2 Difficoltà sintattica

Il secondo livello di difficoltà è quello delle frasi. E' un tipo di difficoltà diverso, legato ad una competenza linguistica più globale e astratta. Cosa rende difficile una frase, al di là delle parole che la contengono, o meglio a parità di difficoltà lessicale? Potremmo cominciare con alcuni esempi:

1. "Il gatto era sopra il tavolo. Il topo entrò dalla tana. Il gatto saltò dal tavolo e inseguì il topo"

2. "Benché fosse sopra il tavolo, quando il topo entrò dalla tana, il gatto saltò giù e lo inseguì lì"

Il contenuto – la storia raccontata – è lo stesso. Anche il numero di lettere e di parole è praticamente lo stesso. Quale è più difficile? La teoria ci dice che sono difficili da leggere:

- Frasi molto lunghe
- Frasi con subordinate una dentro l'altra, o con subordinate meno comuni (concessive, ipotetiche)
- Frasi che usano riferimenti anaforici (pronomi relativi, pronomi indefiniti, ...)
- …

Come fare a definire una versione quantitativa, cioè traducibile in un algoritmo, di questa maggiore difficoltà?

Torniamo all'esempio: nel primo caso abbiamo due punti fermi; nel secondo nessun punto, ma due virgole.

Nel primo caso abbiamo molte ripetizioni: gatto, topo e tavolo sono ripetuti due volte. Nel secondo invece abbiamo una sola occorrenza di gatto, topo e tavolo, ma in più abbiamo delle parole-connettori: benché, quando, giù, lo, lì.

Forse possiamo usare tutti questi elementi: le occorrenze, la frequenza della punteggiatura, la presenza di parole grammaticali (pronomi, avverbi).

Potremmo cioè contare il numero di virgole nel periodo: se sono più di N per numero di parole, allora il periodo è complesso o comunque difficile da leggere.

E potremmo contare le parole che fungono da indicatori lessicali di complessità. Può partire qui un'attività di lettura-ricerca della Costituzione alla ricerca delle subordinate. Quali sono le parole che si possono considerare inizio di una subordinata?

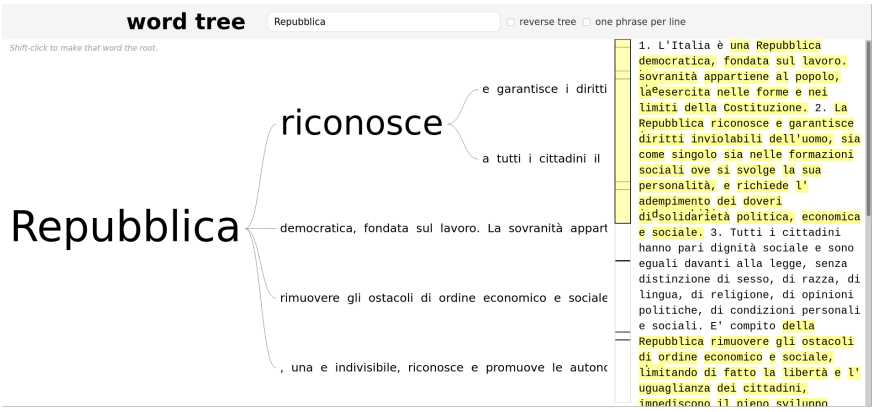
Lo svolgimento di questa attività è lasciato come esercizio – significativo, però - alla classe.

5. Rappresentare i dati

Una volta ottenuti ed elaborati i nostri dati, dobbiamo trovare un modo per comunicarli in maniera efficace e coinvolgente.

Che significa *efficace* in questo caso? Può significare che la risposta alla domanda iniziale (quanto è complessa da leggere la Costituzione? di cosa parla?) emerge meglio dal confronto tra la visualizzazione relativa alla Costituzione Italiana e quella, poniamo, di un Decreto della Presidenza del Consiglio dei Ministri.

Un bell'esempio di strumento di rappresentazione di un testo è la tecnica dei Word Tree.[28] Quello che vedete qui sotto è un esempio tratto da un'applicazione web[29] che permette non solo di visualizzare, ma anche di analizzare il testo e evidenziare il ruolo delle singole parole nelle diverse occorrenze.



Ma è molto più divertente e coinvolgente far creare ai ragazzi il loro strumento di rappresentazione, anche perché così potranno riflettere sulle scelte comunicative, sui codici visuali, sulle preferenze soggettive e quelle condivise.

La cosa più semplice, se restiamo nell'ambito del foglio di calcolo, è usare i grafici. Ad esempio, potremmo fare un

28 La tecnica è stata inventata nel 2007 da Martin Wattenberg e Fernanda Viégas per lavorare sulle concordanze in maniera visuale: http://hint.fm/projects/wordtree/

29 https://www.jasondavies.com/wordtree/

istogramma sulla lunghezza minima, media e massima delle delle frasi della Costituzione e confrontarlo con quello di un altro testo.

Per rappresentare la frequenza dei lemmi (che, ricordiamo, sono quasi 900) probabilmente gli istogrammi non sono una grande idea. Si può invece pensare ad una Word Cloud che mostri solo le parole al di sopra di una certa frequenza. Esistono naturalmente degli strumenti online[30] per creare una Word Cloud a partire dal caricamento di un testo.

Ma è possibile usare linguaggio di programmazione, o un ambiente di coding come Snap!, per realizzarla. Un piccolo suggerimento: in Snap! gli sprite possono scrivere delle parole con dimensioni diverse. Oppure si può usare la capacità degli sprite di essere *clonati*, cioè duplicati. Questa funzionalità si può abbinare al fatto che tramite il modulo aggiuntivo "Text Costume" uno sprite può assumere come costume una lettera o una parola di una certa dimensione.

30 Ad esempio, https://www.wordclouds.com/

Sempre nell'area del coding, possiamo pensare a scrivere una rappresentazione del testo della Costituzione in cui le dimensione di ogni parola, o il suo colore, sono indici della sua frequenza, o della difficoltà di lettura:

I codici per questa attività, come per le altre, sono disponibili sul sito web del libro.