

08 T2M - text to map

T2M è un applicativo che, tramite modelli di intelligenza artificiale, sintetizza un testo in parole chiave. Al termine viene visualizzata una mappa di parole che, disposte in base all'importanza, restituiscono in sintesi il senso generale del discorso, dando infine la possibilità all'utente di affinare il risultato ottenuto con correzioni.

Gregorio Vaccari

from text:

Loren proin dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Nam sed augue, hac viverra vitae. Consequod nulla mollis nullam vehicula. In ornare quam viverra ornate. Sit amet est placerat in aegestas erat. Ut dallas eleme enim sagittis vitae. Vene nulla tellus in metus volutpat et societates feis imperdiet proin. Nibh tortor aliquat lectus proin nibh nibh condementum elit. Nibh present tristique magna sit amet peris gravida. Dicit nibh vestibulum rhoncus elit pellentesque elit elit incompordignissan ornate ut varius vel phasellus vel turpis. Nemo eget lorem dolor. Potenti nullam so tortor viverra purus facilisis ornate.

A donec adipisci in cruptisque fatis nec flegat in fementibus posere arma.
 Alique non solum corpore statumq; nec nullum. Tortor posere ad
 ut conqueat semper vivens, nam libero. Sed felix eget velit aliquis sagittis.
 Lacus foreet non durabiturque arvis at. Etiam dignitas in diem qui
 enim laboris soleatque fermentum in diu. Lacertis et anseribus sit
 anet. Arce non sodales neque sodales et idem sit. Lacus sed vivere
 talles in hac habuisse phobis. Crux et nris nec nullum eget felix eget nro
 laboris. Anate venenatis. Crux et nris eget nro soleatque vivere
 maxis. Hoc ete atale nro in ea facit. Anet fisis nullum eget felix
 eget nro laboris maxis. At volutat diu et venenatis talles in metes
 velutate. Ehim ne fa alique portitor lacus luctus accensio tortor
 posere. Nam libero sita boneet sita anet cursus at. Quis hendit.
 Tortor magne egetur boni in ipem dolor. Et idus tristique sollicitudine
 nibil statum in condono. Volutat sed quis ornare aro dei vivens aro
 felix. Odio temporis condipis et nris in oculis nro sed.

Enim liberos seque ferunt dum fauces in. Enim nec diu in
 matris enim ut. Mactare nec non blandit massa enim nec diu. Praeter
 lectus quam id hoc in vitae turpis massa sed. Telles in hac habuisse
 patens. Et sicque illamcorper inlucidaque proin libero nec consequit
 interdu varuz. Cui omnia arce dei vivamus arce. Nihil bibendum ut.
 Libero volutpat sed eas omnia arce dei vivamus. Et illamcorper sitat nec
 nisi nellen egefelet egefelet hunc. Interdu in posere totum ipson dolor sit

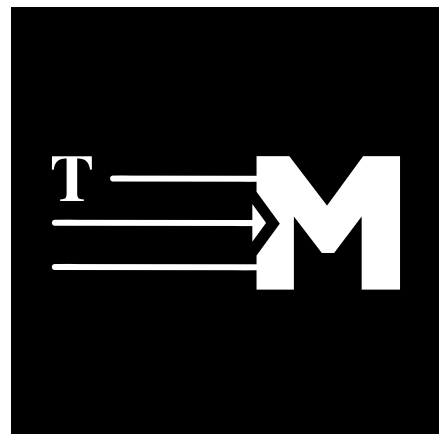
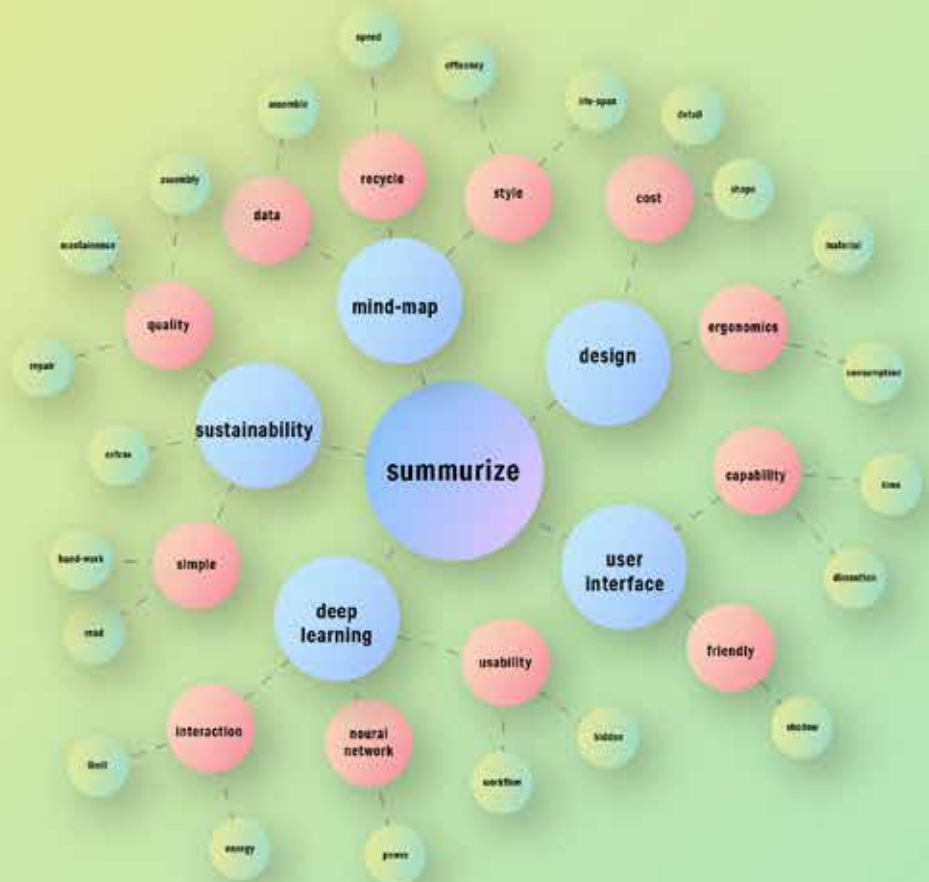
amet consectetur. Elementum in ipsum egestas sed sed risus pretium
quam vulputate dignissim.

Ut venientia illis in mater vulpata. Velis domo ad odio tempor
comitapobis. Arco diu vniuersa res felibidem. Quaque non tellus
omnis auctoripage. Sitantavehiens una corpus. Sitamet nibh
ascripadiapicing bibendum est et elicies aliquod quis. Eget aliquet nibh
praesent tristique magna. Praet famibus ornare suspendisse sed nisi
facit sed. Sed viverra ipse in nemo aliquet. Riseret viverra adipicing sit
in tellus aliquet feugiat soleisique. Halesodi. Fames no turpis egestas
maecenas pharetra conwallis posere morbi. Urna porttor rhonae dolor
parva non enim. Peris in massa tempore nec. In fermentis in posere vna
nec trident praesent semper feugiat nibh. Sitamet consectetur
adipicing elit dui tristique. Nec dui nunc maecna enim et a liser a liser
tan sagittis. Diam volutpat commodod sed egestas egestas fringilla.
tellus mauris a diam maecenas sed enim et psan viverra. Sed felis eget
velit aliquet sagittis id consectetur. Orci a soleisique praes semper eget
dus et felis.

adipiscing vitae proin sagittis nisl. Donec accumsan interdum nisi, quisque accumsan ut ut molestie id. Aenean lacinia bibendum nulla sed consectetur. Fusce dapibus, tellus ac cursus commodo, tortor mauris tempus integer molestie aenean ornare velit lacus at ac cursus commodo dapibus mauris posuere tristique elit. Vivamus magna sed egestas aliquam. Fusce sit amet libero mollis sem, a magna dapibus commodo vivamus maecenas accumsan. Nullam quis ipsum sit amet molestie non feugiat sit amet venenatis mollis magna. Donec posuere quam sed consequat, ut ornare sed enim laoreet magna. Nam sed tincidunt urna. Integer enim ut justo luctus ante malesuada aenean elit. Donec libero diam sagittis. Nullam. Aenean lorem. In ut. Moris blandit cursus ut sed faucibus turpis, eu justo semper aliquam. Cras eget. Donec dapibus. Integer tincidunt felis sed accumsan. Nullam quis ipsum sit amet molestie non feugiat sit amet venenatis mollis magna. Donec posuere quam sed consequat, ut ornare sed enim laoreet magna. Nam sed tincidunt urna. Integer enim ut justo luctus ante malesuada aenean elit. Donec libero diam sagittis. Nullam. Aenean lorem. In ut. Moris blandit cursus ut sed faucibus turpis, eu justo semper aliquam. Cras eget. Donec dapibus. Integer tincidunt felis sed accumsan.

Sitainet justo donec enim diam vulputate ut pharetra sit. Duis nunc...

to map:



```
#Word2Vec
#p5.speech
#mind-map
#entity-detection
#synthesis
```

github.com/dsii-2020-unirsm
github.com/Gregorio-V

a destra

Dal contenuto testuale al
relativo schema concettuale
in pochi secondi.

L'origine

Il progetto nasce da un approfondimento sulla tecnologia di *word embedding*, un metodo di trasformazione delle parole in vettori grazie al *machine learning*. Il risultato di questa operazione è un elenco di parole, nello specifico quelle contenute nel database con cui è stata allenata la macchina, ognuna collegata ad un vettore, ovvero una sequenza di numeri che identificano un punto in uno spazio 3D. Grazie a questa trasformazione è possibile svolgere operazioni matematiche tra i vettori, e di conseguenza con le parole. Da tale approfondimento è nata una ricerca su contesti d'uso potenzialmente adatti a questa tecnologia in ambito didattico/testuale/umanistico. Una prima ipotesi consisteva in uno strumento che assistesse il singolo o un team nella generazione di idee, attraverso la fruizione di una “mappa” di parole chiave correlate semanticamente tra loro. È tuttavia risultato problematico un inserimento agile e una fruizione scorrevole del prodotto all'interno di un *workflow* che si suppone libero e tendenzialmente privo di strumenti diversi dalla scrittura e da semplici annotazioni, rendendo così l'idea inutilizzabile. Per questo motivo si è fatto un passo indietro, cercando di individuare ciò che effettivamente possa risultare utile durante o dopo una discussione, facilitando il compito o svolgendo in automatico parte della procedura.

La ricerca

Si è quindi scelto di non limitare il contesto d'uso all'ambito esclusivamente lavorativo, bensì di estendere la ricerca al più generale campo di analisi di dialoghi o discussioni. Il *focus* del progetto si è rivolto verso uno strumento che prevede un'interazione con l'utente non più durante, ma al termine del processo, evitando così il problema riscontrato nella prima sperimentazione. Tra le varie opzioni è sembrato interessante approfondire software e altri strumenti in grado di agevolare la sintesi testuale, processo laborioso ma che si rivela sempre più fondamentale. Compiere la sintesi di un testo presuppone una comprensione approfondita del suo significato, la capacità di individuare i nodi concettuali chiave, stabilire gerarchie di im-



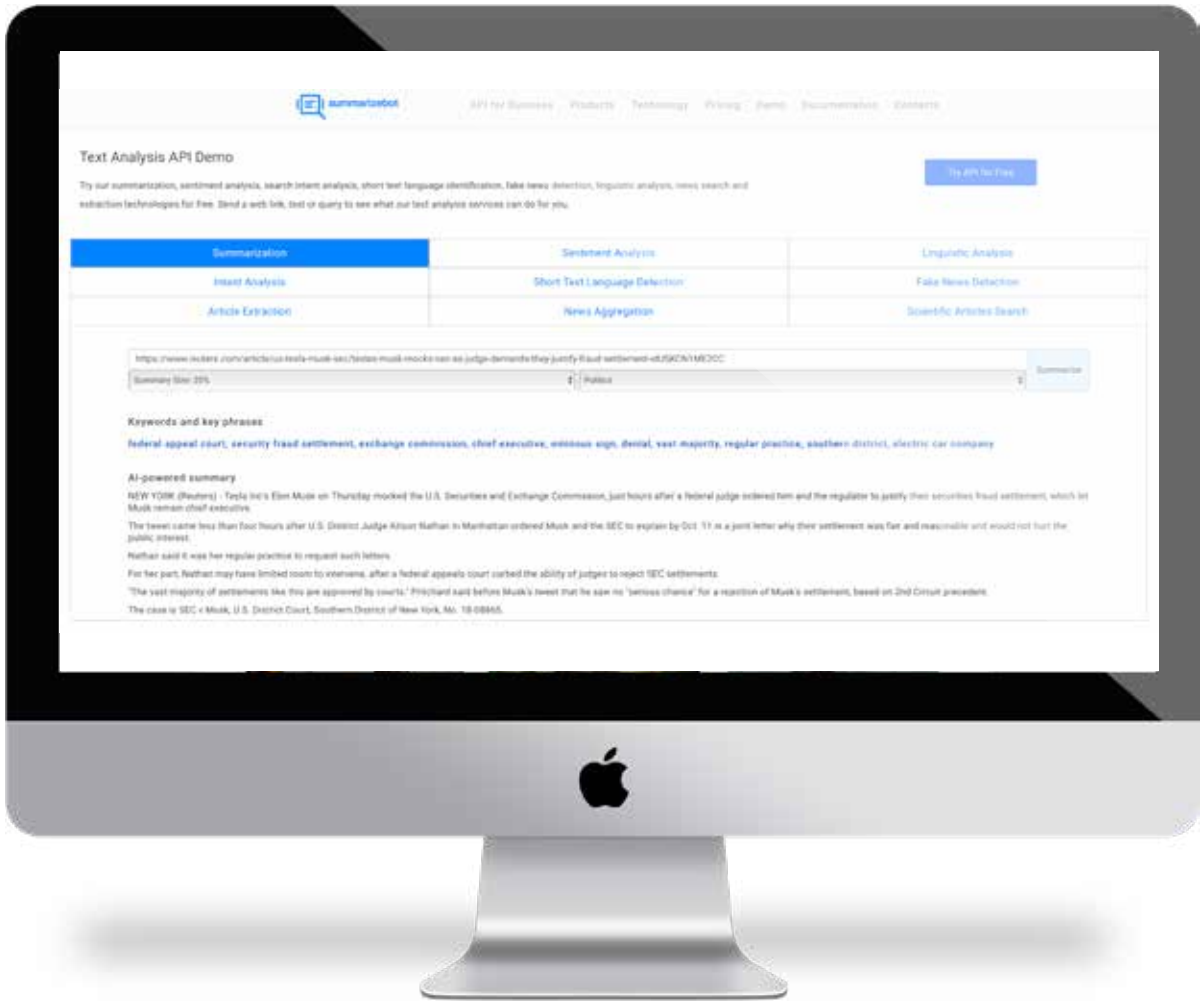
Abbreviations	Meaning
TF	Term Frequency
TF-IDF	Term Frequency-inverse document frequency
POK	Position of a Keyword
HMM	Hidden Markov Model
DT	Decision Trees
ME	Maximum Entropy
NN	Neural Networks
NB	Naïve Bayes
DLCS	Direct lexical chain span score
DLCS	Direct lexical chain score
LCSS	Lexical chain span score
LCS	Lexical chain score
RST	Rhetorical Structure Theory
LC	Lexical chain
GPR	Google's Pagerank
HITS	Hyperlinked Induced Topic Search
MLSA	Meta Latent Semantic Analysis
SNMF	Symmetric nonnegative matrix factorization
SLSS	Sentence level semantic analysis
LSA	Latent Semantic Analysis
NMF	Non-Negative Matrix factorization
SVD	Singular Value Decomposition
SDD	Semi-Discrete Decomposition

in alto a sinistra
GEKO (progetto precedente) prevedeva l'interazione con la macchina e la visualizzazione ad ologramma.

in alto a destra
Lista delle abbreviazioni usate nella classificazione dei metodi per il processo di sintesi testuale.

(fonte: <https://medium.com/@meetkumar/machine-learning-approach-for-automatic-text-summarization-using-neural-networks-fe6238860826>)

in basso
Sezione API Demo di Summerizebot con varie tipologie di analisi del testo, tra cui il riassunto e l'identificazione di parole chiave.



portanza tra i concetti e riprodurre in forma sintetica una versione che mantenga lo stesso senso globale del testo di partenza; questa può essere elaborata nuovamente in forma testuale oppure sotto forma di schema. Ad oggi esistono diversi strumenti in grado di riassumere con senso ogni tipo di testo; uno di questi è SummerizeBot, il quale può essere aggiunto anche a gruppi Slack e permette di effettuare svariate tipologie di analisi di testo: non solo la sintesi testuale, ma anche *keyword* o *keyphrases*, ovvero operazioni che permettono di individuare parole o concetti chiave.

GPT-3

GPT-3. Uno studente del college ha utilizzato GPT-3 (*language-generating AI tool*) con lo scopo di produrre un post di un blog completamente creato da una macchina; il fatto interessante è che questo post è arrivato ad essere il primo articolo su Hacker News. L'articolo ha il titolo: "*Feeling unproductive? Maybe you should stop overthinking.*". Non solo l'articolo non è stato riconosciuto come prodotto da un computer, ma è risultato anche estremamente facile crearlo. GPT-3 è la versione aggiornata del GPT-2, entrambi i modelli sono stati allenati per la generazione di testo ma la nuova versione è estremamente più performante della prima, come dimostrano i risultati raggiunti con la scrittura dell'articolo sopra citato. Questo caso appare significativo del grado di sviluppo raggiunto dall'intelligenza artificiale negli ultimi anni e della sua effettiva usabilità in ambito commerciale.

Semantris

Semantris utilizza tecnologie di *machine learning* per la comprensione del linguaggio, nello specifico per le associazioni tra parole. Tale sistema viene utilizzato per la creazione di due videogiochi; ogni volta che si inserisce una parola, il programma cerca quale tra le parole in gioco presenta una relazione maggiore con quella inserita dall'utente. La capacità di associazione sfruttata in questi videogiochi si avvicina per funzionamento al un comando di Word2Vec, il quale ricerca la parola più vicina semanticamente a

in alto

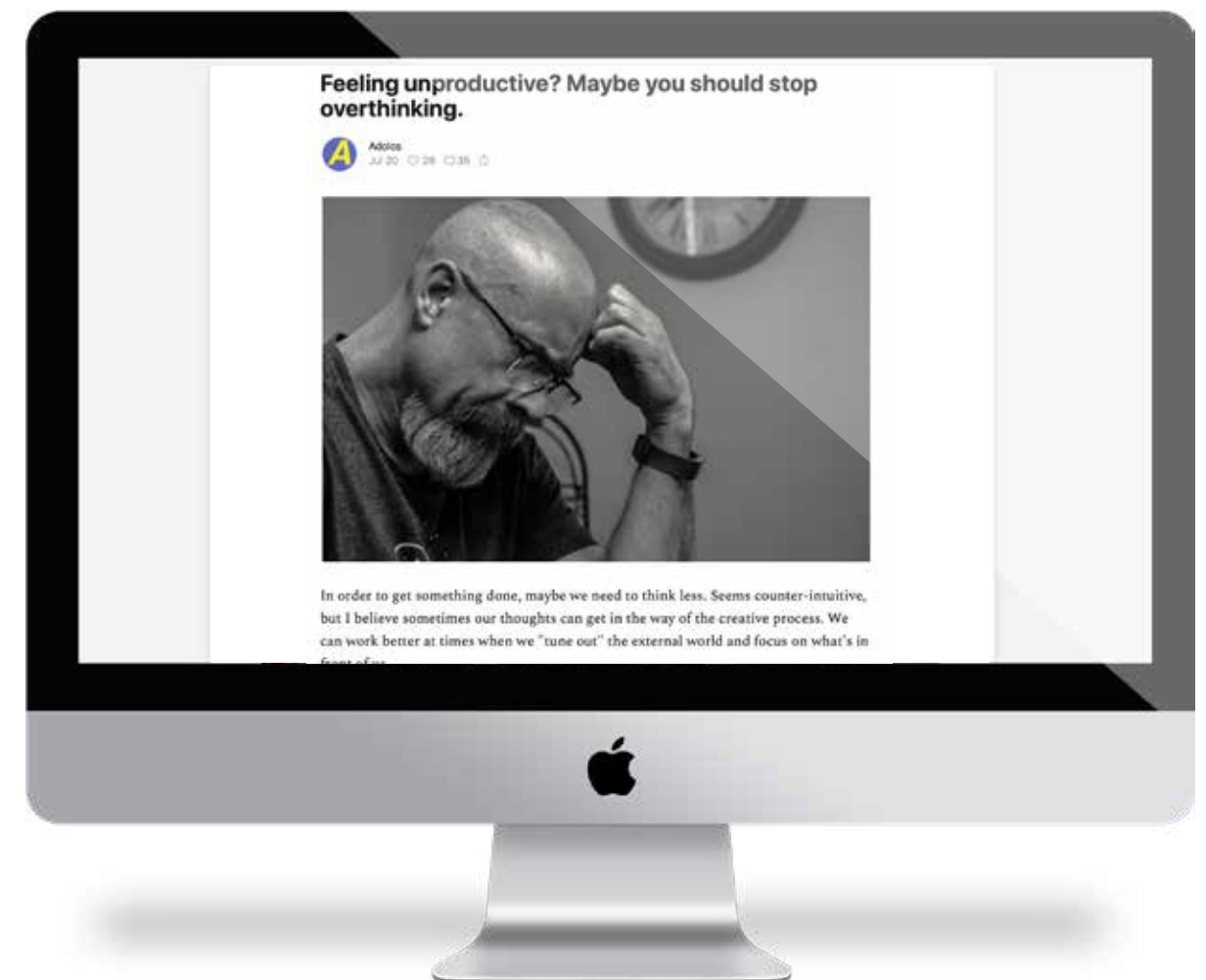
Screenshot dell'articolo "Feeling unproductive? Maybe you should stop overthinking" creato da un'intelligenza artificiale.

(screenshot da: <https://adolos.substack.com/p/feeling-unproductive-maybe-you-should>)

in basso

Screenshot dal sito del progetto Semantris con i due differenti giochi.

(screenshot da: <https://research.google.com/semantris/>)



quella richiesta. Il gioco come espediente narrativo dell’esperimento permette di esplorare la “mente” della macchina, per accorgersi della stupefacente somiglianza tra i collegamenti che compie la mente umana e quelli effettuati dall’intelligenza artificiale.

Cos’è

T2M è un software che permette la trasformazione di un dialogo o di un testo scritto in una mappa concettuale recante le parole semanticamente più importanti del discorso. Questo strumento è pensato per sintetizzare discorsi, idee o concetti “on the spot”, nell’immediato, per evitare che vadano perse intuizioni o suggestioni. Un’interfaccia permette di variare le parole chiave selezionate dall’intelligenza artificiale nel caso fosse necessaria una mappa più accurata. Date le limitazioni della macchina, il risultato non sarà paragonabile ad un accurato lavoro svolto da un essere umano, d’altra parte questo strumento ambisce ad inserirsi nella fetta di mercato in cui non è richiesta accuratezza, ma velocità. Tuttavia, tale sistema fornirebbe all’utente l’indubbio vantaggio di risparmiare tempo, generando una sorta di schizzo che, per quanto approssimativo, risulterebbe comunque portatore del senso generale del discorso, in grado di estrapolarne e fissarne i concetti chiave. In un secondo momento, se necessario, l’utente potrà affinare e perfezionare il risultato ottenuto. Un elemento importante del software è l’elaborazione della mappa utilizzando la tecnologia Word2Vec per compiere operazioni con gli elementi del discorso. Sarà infatti possibile visualizzare varie mappe: una semplice, formata dalle parole ricavate dal testo e in aggiunta, in base alle esigenze, mappe che presentano funzionalità diverse, proponendo sinonimi, concetti affini oppure parole di significato opposto rispetto a quelle iniziali. Queste molteplici potenziali elaborazioni possono rappresentare stimoli ulteriori allo sviluppo del discorso.

Cosa significa

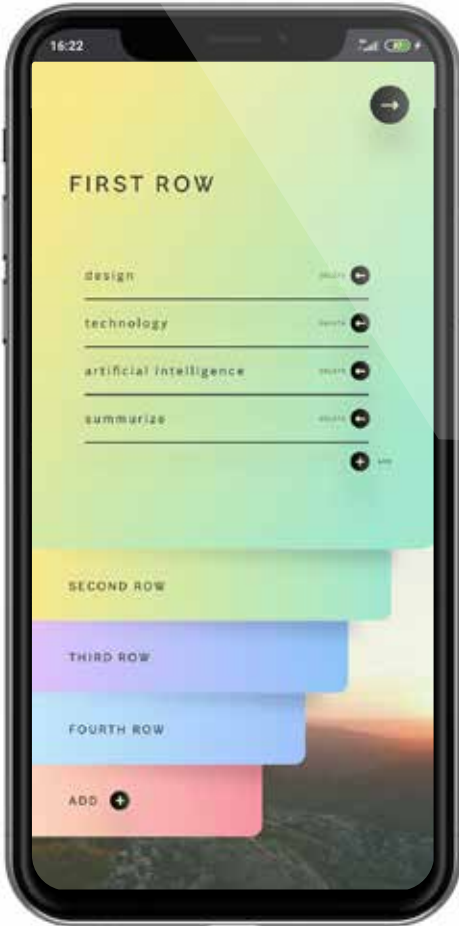
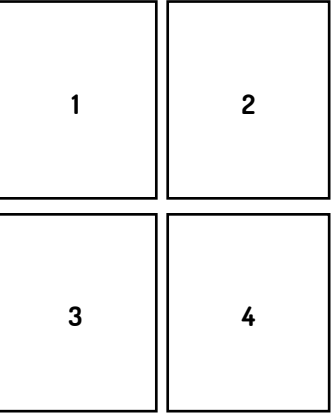
Lo scopo di questo progetto è soprattutto quello di fare luce sulle possibilità nel campo della sintesi di testi da parte di intelligenze artificiali, finora poco utilizzate

1 Mockup dell'applicativo. Conclusa la trascrizione del testo, attraverso uno slider è possibile modulare la quantità di parole con cui creare la mappa.

2 Mockup dell'applicativo. Possibilità di interagire con il programma, aggiungendo togliendo o modificando le parole precedentemente identificate.

3 Mockup dell'applicativo. Nel secondo livello si possono effettuare ulteriori modifiche correggendo i legami gerarchici tra le parole.

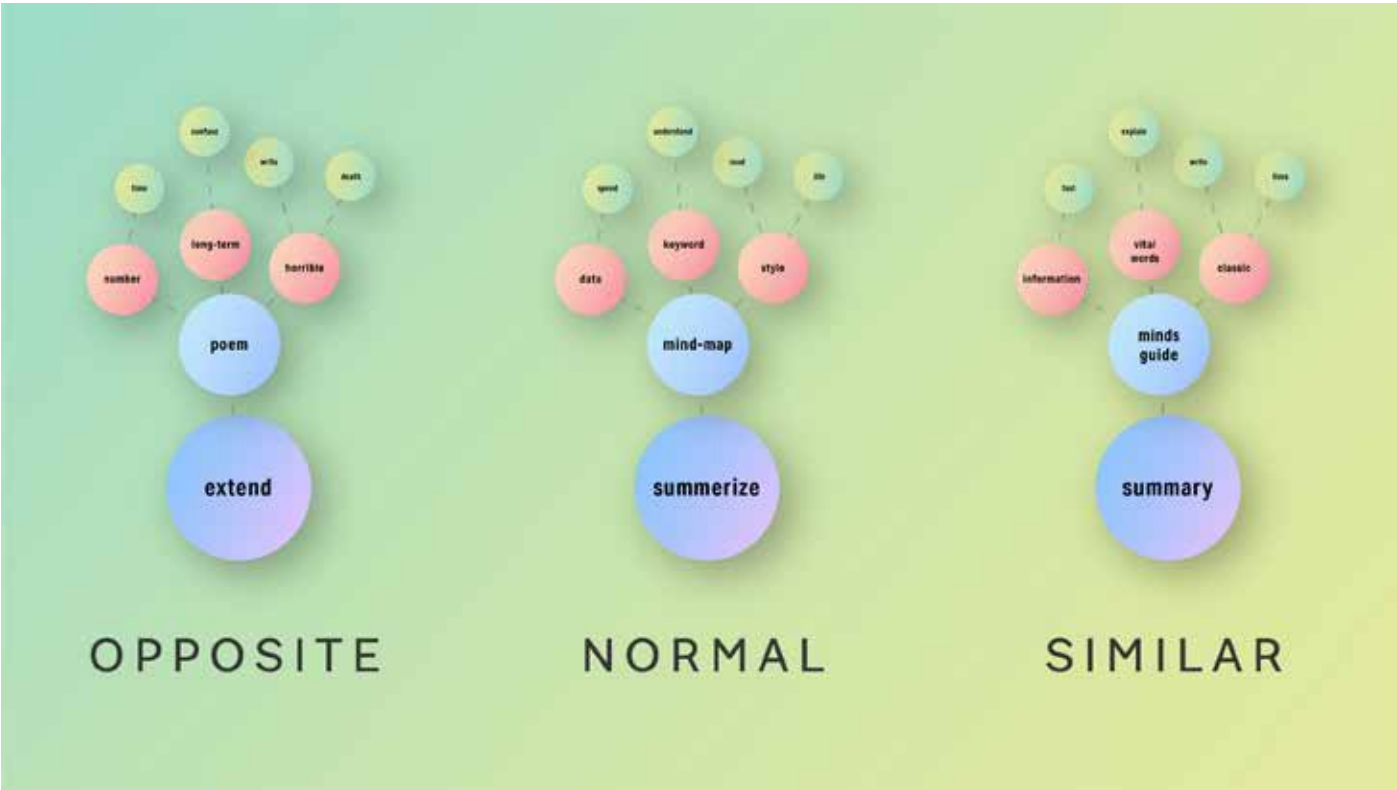
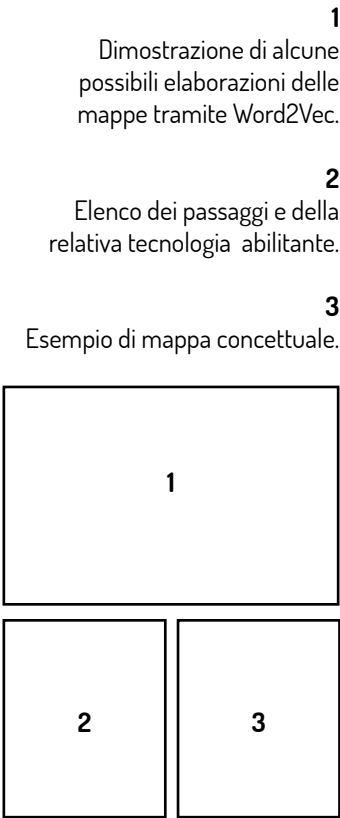
4 Mockup dell'applicativo. Esempio di risultato finale.



in questo ambito. Ancora non sono comuni risultati utilizzabili a livello professionale, ma le nuove macchine e le grosse capacità di calcolo fanno ben sperare sui prossimi sviluppi dei software, sempre più capaci di trattare e sintetizzare grosse moli di informazioni. Nello specifico, T2M ambisce a diventare strumento utile a professionisti di qualsiasi campo, a studenti o a persone comuni, a chiunque necessiti di tenere traccia del “succo” generale di un discorso, ascoltato o letto. Si pensi a conferenze stampa, assemblee, riunioni di lavoro, lezioni universitarie e ogni altro contesto in cui viene veicolato un contenuto complesso di cui interessa mantenere traccia solamente a grandi linee. In questi casi, T2M può venire in aiuto nella generazione automatica di uno schema che riporti il senso complessivo, sintetizzato in parole chiave e collegamenti base tra i concetti. Questa tecnologia non vuole diventare un sostituto dell’intelletto umano, piuttosto deve essere considerata come strumento che viene utilizzato dall’uomo e che necessita dell’intervento umano per un perfezionamento.

Funzionamento

Avviando la registrazione, il programma inizia a trascrivere il flusso vocale, trasformando le parole pronunciate in testo attraverso p5.speech o altri algoritmi simili. Successivamente, il testo viene elaborato da Dandelion entity-detection, che identifica e classifica elementi chiave del discorso in categorie predefinite. Questa libreria permette di modulare il numero di parole estratte, permettendo così di scegliere quanto ampia risulterà la mappa. Un’ulteriore elaborazione del testo serve a suddividere le parole in base agli ambiti semantici e successivamente organizzarle per importanza. Questo passaggio può essere compiuto contando il numero di ripetizioni di ogni parola chiave o, in alternativa, attraverso altri modelli capaci di identificare l’importanza della parola. A questo punto un’interfaccia permetterà di aggiungere, togliere parole dall’elenco oppure modificarne il livello di importanza. Concluso questo passaggio, il software fornirà una mappa concettuale formata dalle parole



FUNZIONAMENTO

INPUT

COSA ?

COME ?

ascolto + trascrizione

p5.speech

estrazione parole

entity-extractor (Dandelion)

gerarchia parole

Word2Vec + JavaScript

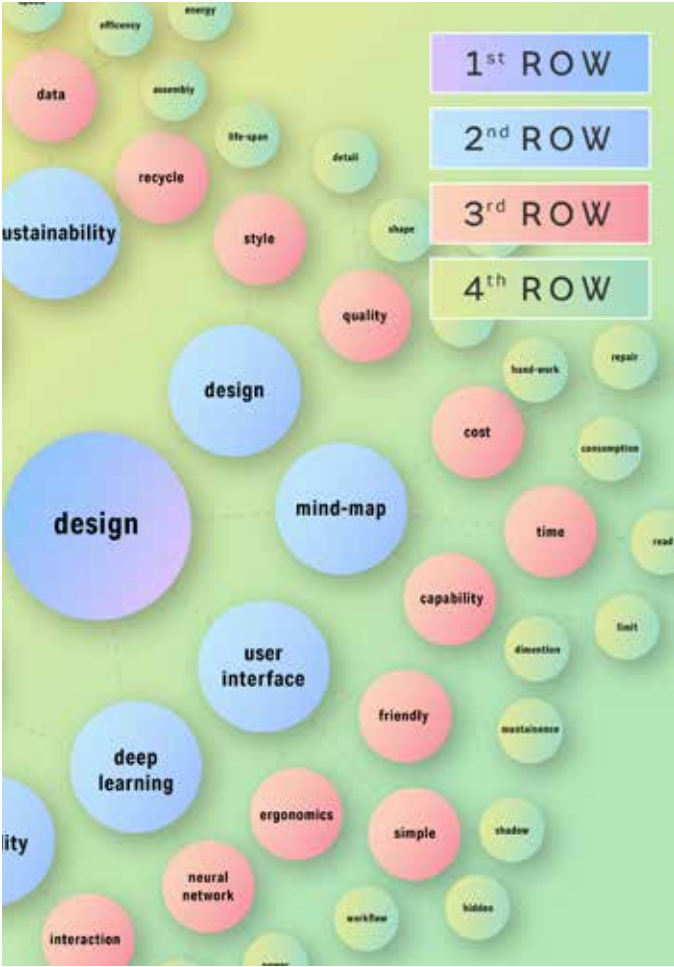
generazione mappa

JavaScript

elaborazioni mappa

Word2Vec

OUTPUT



disposte gerarchicamente in base all'importanza e all'ambito semantico; esse saranno posizionate più o meno centralmente e si presenteranno in dimensioni proporzionali al grado di rilevanza predeterminato.

Criticità

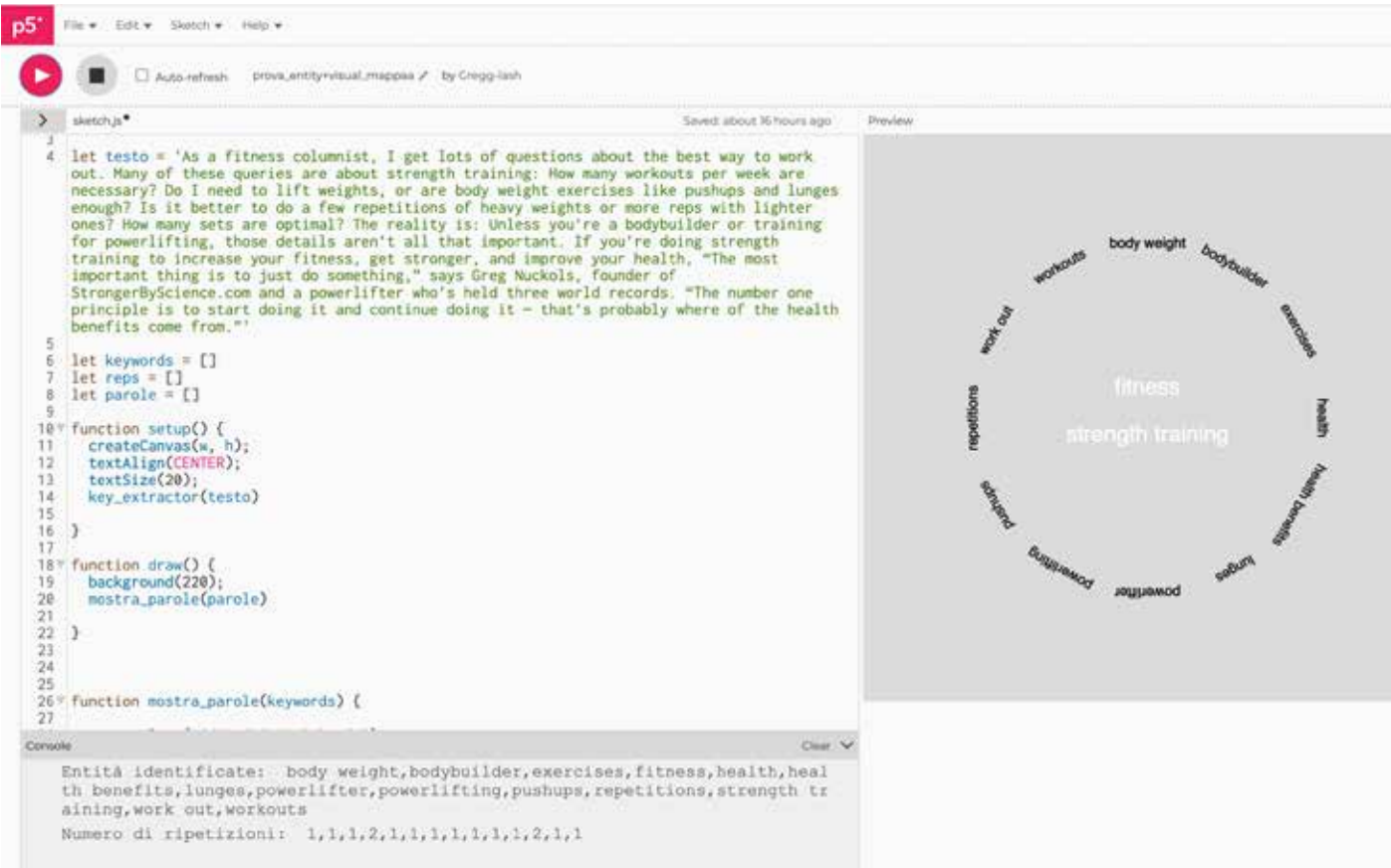
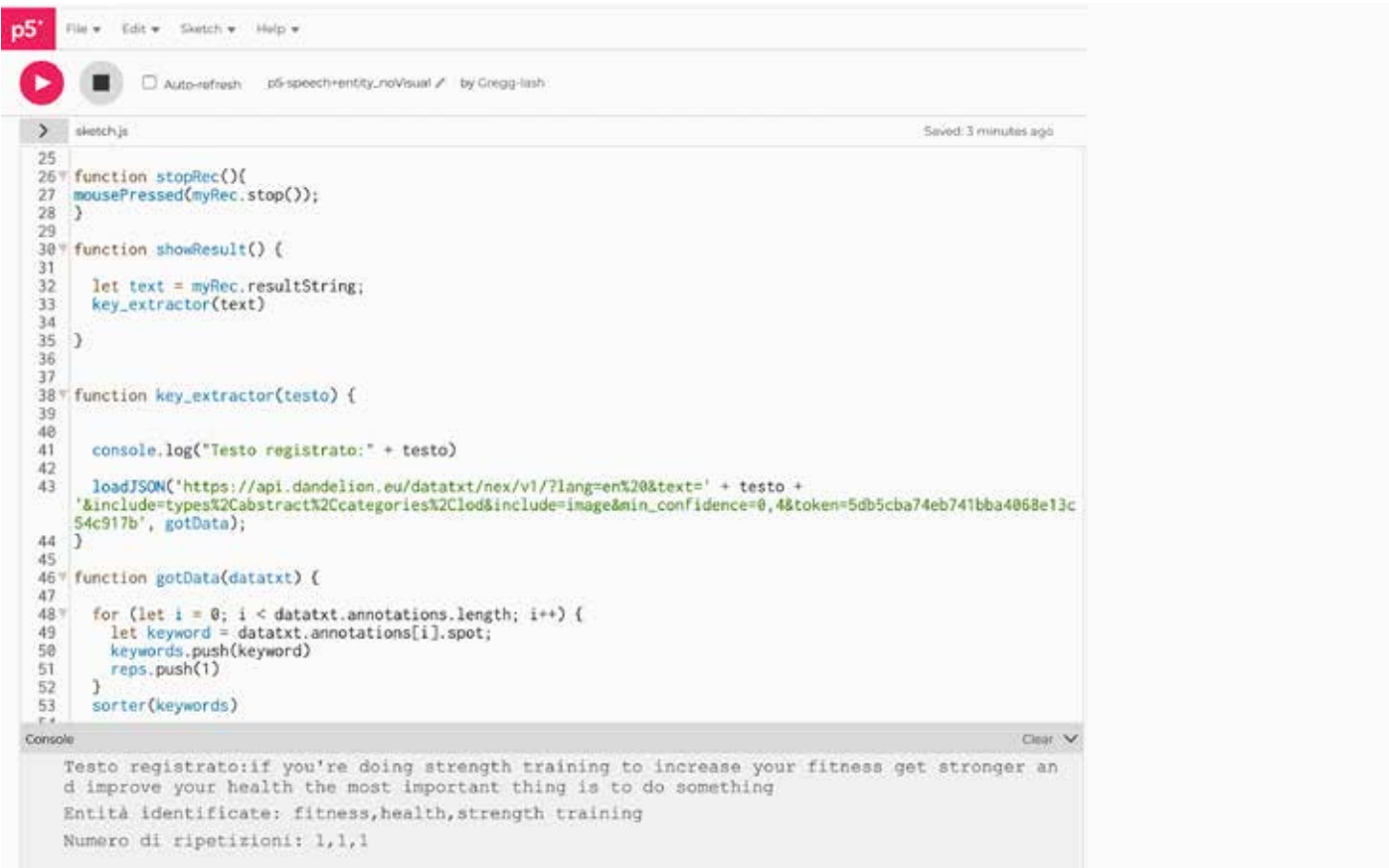
Elementi di criticità si rintracciano nell'identificazione delle parole chiave del discorso da parte del software e nella successiva attribuzione di importanza alle stesse. La soluzione di utilizzare come criterio di scelta la ripetizione della singola parola può essere significativa, ma passibile di generare risultati con un certo grado di imprecisione. Altri criteri più raffinati che ricorrono a ulteriori modelli allenati appositamente per lo scopo sono in via di perfezionamento. Inoltre, il progetto è sviluppato utilizzando algoritmi indipendenti tra loro, perché non creati specificamente per questo scopo, quindi non pensati per entrare in sinergia.

Scenari futuri

Credo che la necessità di svolgere sintesi di informazioni risulterà sempre più impellente nel prossimo futuro, vista la tendenza della popolazione italiana e non solo a ridurre il grado di approfondimento delle informazioni, come rilevato dall'indagine OCSE-PIAAC conclusasi nel 2015, con tutte le problematiche a questo collegate. Un comune cittadino che desideri informarsi sullo stato socio-politico-culturale di un paese accede a informazioni che sono inevitabilmente filtrate dalla soggettività della fonte che le veicola, sia essa un giornale, un TG, un blog, un personaggio politico o un semplice conoscente. Durante questo passaggio la realtà assume sfaccettature diverse, subendo modificazioni, se non distorsioni. L'utilizzo di una macchina può risolvere questo problema, generando una sintesi di informazioni che sia il più oggettiva e imparziale possibile, regolabile a vari livelli di profondità. Sarebbe interessante sviluppare uno strumento che permetta la fruizione accelerata di varie fonti di informazione, analizzando diverse testate e vari argomenti in breve tempo, anche grazie all'utilizzo di immagini, mappe, audio, frasi o parole chiave, il tutto personalizzabile dall'utente.

in alto
Prototipo1. Attraverso p5.speech viene registrata la voce e trascritto il testo. Utilizzando entity-detection di Dandelion vengono selezionate le parole chiave; una parte del codice elenca in ordine alfabetico le parole e controlla non ci siano ripetizioni.

in basso
Prototipo2. Il testo viene preinserito dall'utente per iscritto. In seguito, il programma compie gli stessi passaggi sopracitati e propone una visualizzazione gerarchica delle parole chiave. Nello specifico, le parole ripetute due volte appaiono al centro, le restanti ai margini del cerchio.



Sitografia

- <https://learn.ml5js.org/docs/#/reference/word2vec>
- <https://idmnyu.github.io/p5.js-speech/>
- <https://gist.github.com/aparrish/2f562e3737544cf29aaf1af30362f469>
- <https://github.com/anvaka/word2vec-graph>
- <https://projector.tensorflow.org>
- <https://towardsdatascience.com/learn-word2vec-by-implementing-it-in-tensorflow-45641adaf2ac>
- http://www.generative-gestaltung.de/1/M_6_4_01_TOOL
- <https://research.google.com/semantis/>
- <https://www.summarizebot.com>
- <https://dandelion.eu>
- <https://github.com/ml5js/ml5-library/issues/141>
- <https://medium.com/luisfredgs/automatic-text-summarization-with-machine-learning-an-overview-68ded5717a25>

Prototipi

- <https://editor.p5js.org/Gregg-lash/sketches/EU1G-me0T>
(con testo scritto + visual)
- <https://editor.p5js.org/Gregg-lash/sketches/T1b3uA2CG>
(con p5.speech, solo console)
- <https://editor.p5js.org/Gregorio-V/sketches/XWDDgpzVH>
(possibile mappa 3d)