



Large language models Applicazioni in ambito psicologico

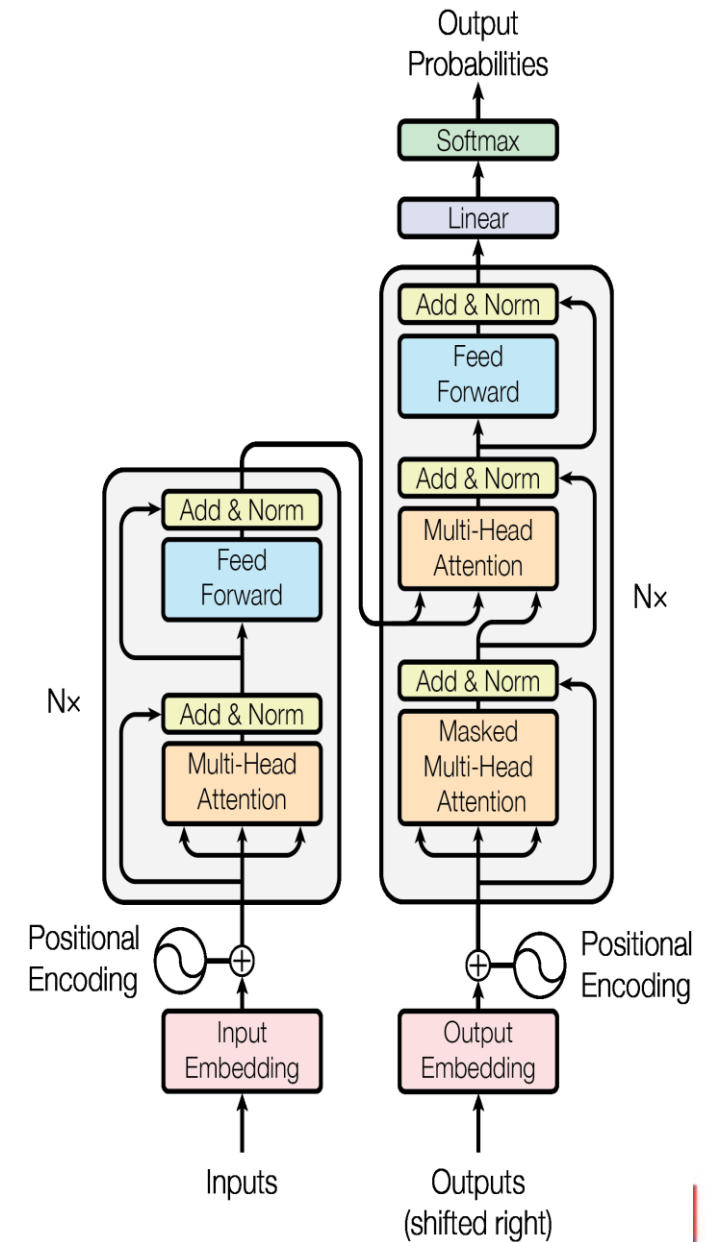
Michela Ponticorvo

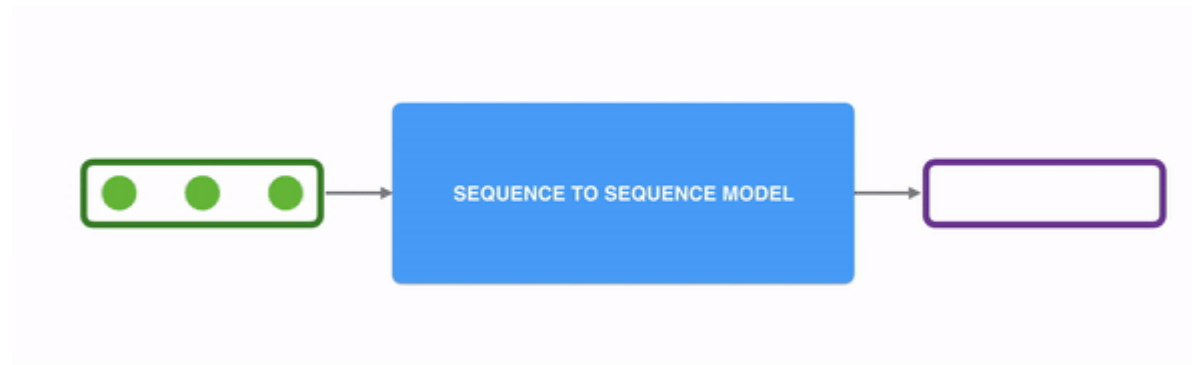


Transformers

Sono un tipo di architettura di rete neurale molto in voga attualmente. Sono stati ad esempio utilizzati da OpenAI nei loro modelli linguistici e anche da DeepMind per AlphaStar, il programma che gioca come un professionista a StarCraft.

I transformers sono stati sviluppati per risolvere il problema della trasduzione di sequenze o traduzione automatica: qualsiasi attività che trasforma una sequenza di input in una sequenza di output. Questo include il riconoscimento vocale, la trasformazione da testo a voce, ecc.





Affinché i modelli eseguano la trasduzione della sequenza, è necessario disporre di una sorta di memoria. Possono esserci riferimenti ad elementi precedenti del discorso.

Per tradurre frasi, un modello deve capire questo tipo di dipendenze e connessioni.

Le reti neurali ricorrenti (RNN) e le reti neurali convoluzionali (CNN) sono state utilizzate per affrontare questo problema per le loro proprietà.

<https://towardsdatascience.com/transformers-141e32e69591>

Attenzione

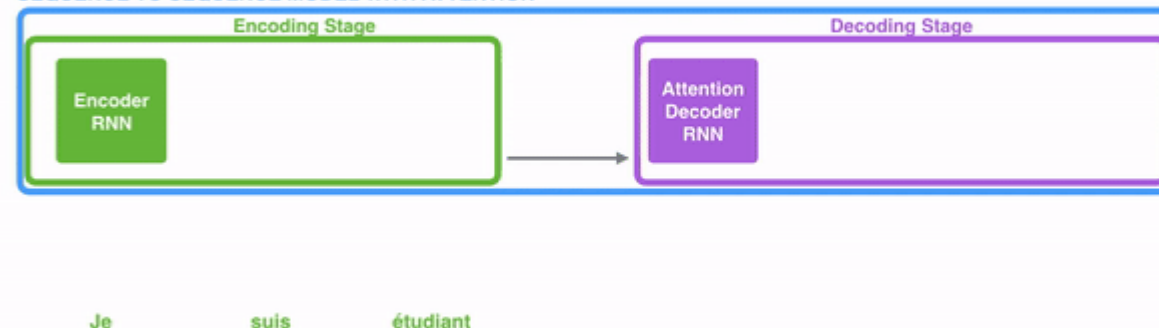
I ricercatori hanno proposto una tecnica per prestare attenzione a parole specifiche.

Quando traduco una frase, presto particolare attenzione alla parola che sto traducendo. Quando trascrivo una registrazione audio, ascolto attentamente il segmento che sto scrivendo attivamente. Le reti neurali possono ottenere questo stesso comportamento usando l'attenzione, concentrandosi su parte di un sottoinsieme delle informazioni che ricevono.

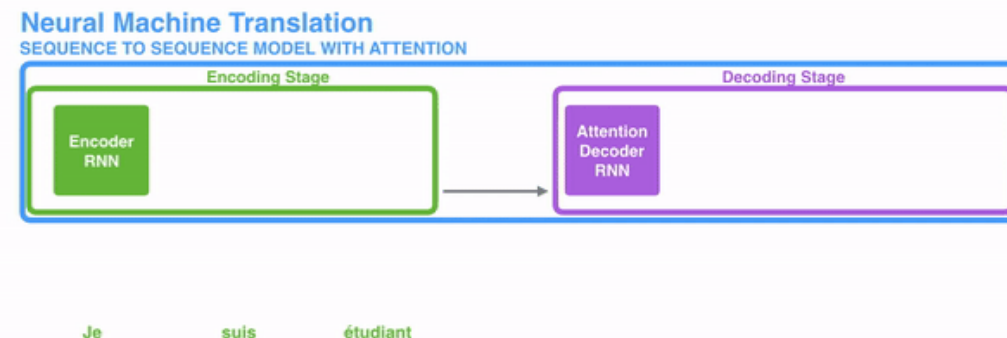
Ad esempio, una RNN (rete neurale ricorrente) può occuparsi dell'output di un'altra RNN. Ad ogni passo temporale, si concentra su diverse posizioni nell'altra RNN.

Invece di codificare solo l'intera frase in uno stato nascosto, ogni parola ha uno stato nascosto corrispondente che viene passato fino alla fase di decodifica. Quindi, gli stati nascosti vengono utilizzati in ogni fase dell'RNN da decodificare.

Neural Machine Translation
SEQUENCE TO SEQUENCE MODEL WITH ATTENTION



Attenzione



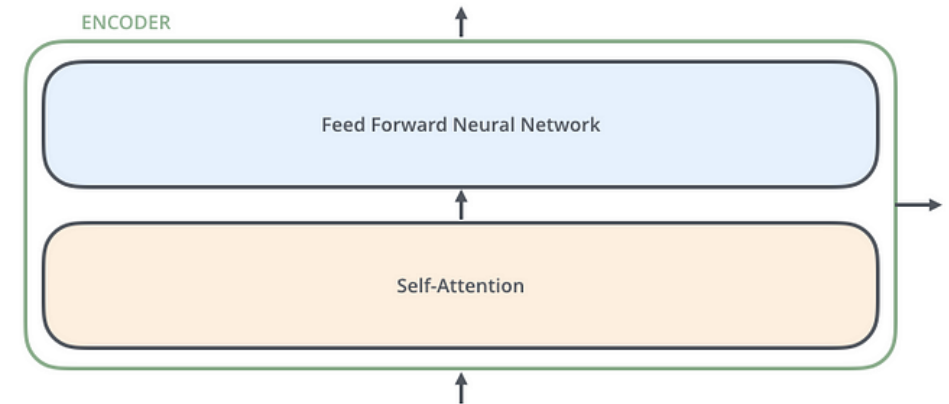
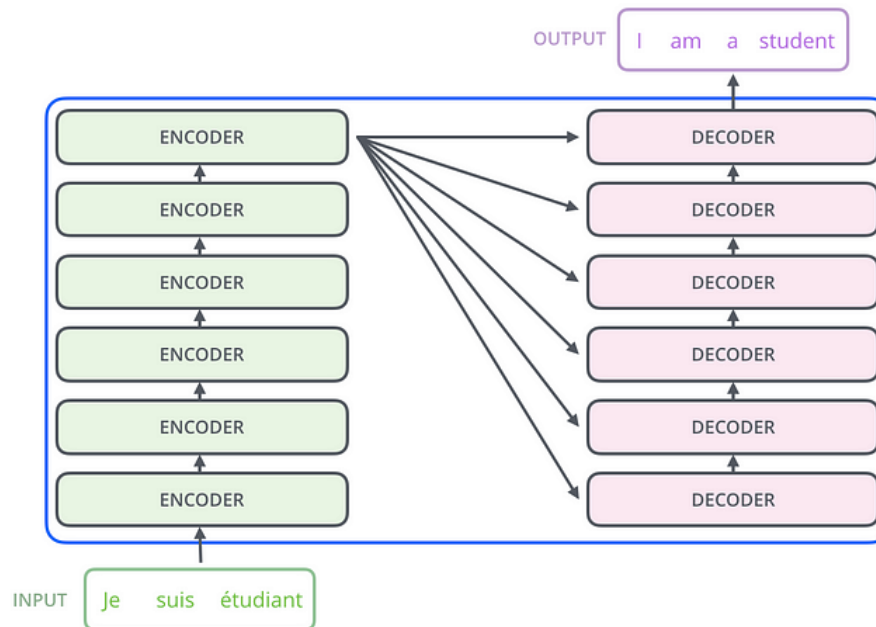
L'idea alla base è che potrebbero esserci informazioni rilevanti in ogni parola di una frase. Quindi, affinché la decodifica sia precisa, si deve tenere conto di ogni parola dell'input, usando l'attenzione.

Per attirare l'attenzione delle RNN nella trasduzione della sequenza, dividiamo la codifica e la decodifica in 2 fasi principali. Un gradino è rappresentato in verde e l'altro in viola. Il passaggio verde è chiamato fase di codifica e il passaggio viola è la fase di decodifica.

Lo step verde è incaricato di creare gli stati nascosti dall'input. Invece di passare ai decodificatori un solo stato nascosto, passiamo allo stadio di decodifica tutti gli stati nascosti generati da ogni “parola” della frase. Ogni stato nascosto viene utilizzato nella fase di decodifica, per capire a cosa la rete dovrebbe prestare attenzione.

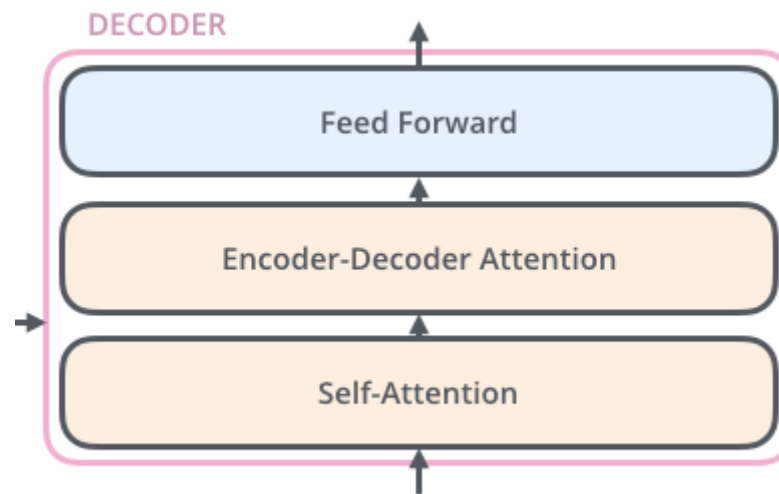
Transformers

I transformers affrontano questo problema usando encoders e decoders insieme a modelli di attenzione. L'attenzione aumenta la velocità di traduzione da una sequenza all'altra. Abbiamo più encoder e più decoder, molto simili tra loro. Gli encoder hanno la stessa architettura con uno strato di self-attention ed una rete feed-forward.



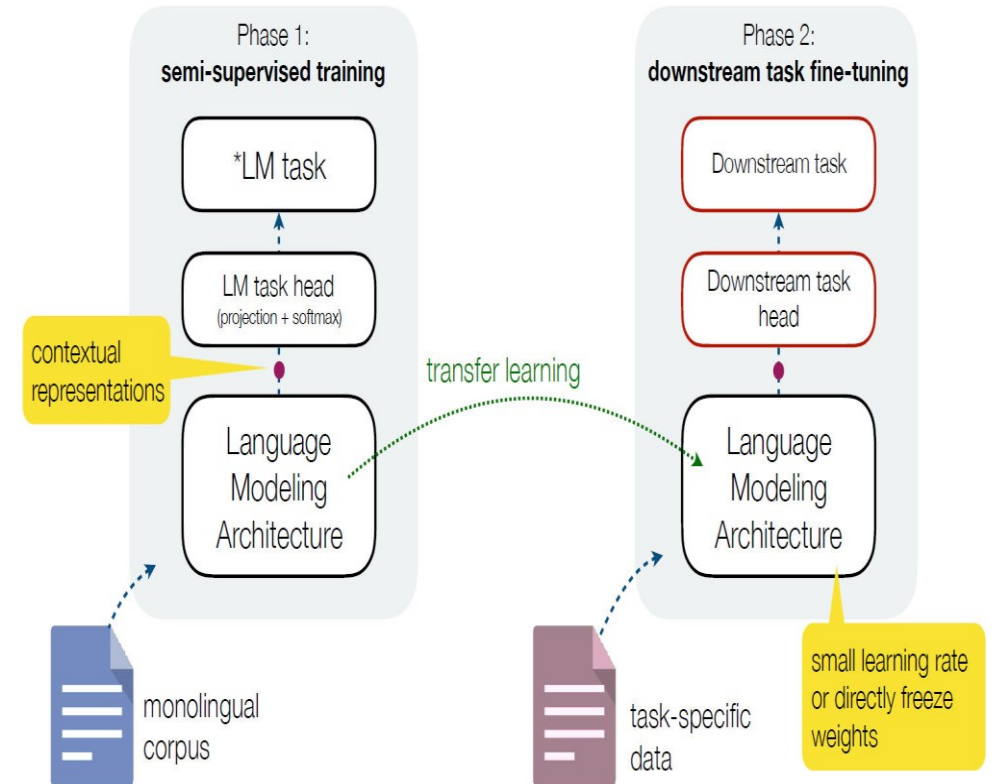
Transformers

Gli input dell' encoder passano attraverso lo strato di self-attention che aiuta l'autoencoder a guardare alle altre parole della sequenza dell'input, mentre codifica una parola specifica. Il decoder ha entrambi questi strati, ma tra di loro c'è lo strato attentivo che aiuta il decoder a focalizzarsi su parti importanti della sequenza di input.



Large language models

- I LLM sono modelli di intelligenza artificiale che utilizzano algoritmi di deep learning per elaborare e comprendere il linguaggio naturale.
- Vengono addestrati su enormi quantità di dati di testo per apprendere i modelli e le relazioni delle entità nella lingua.
- Sono in grado di eseguire molti tipi di attività linguistiche: possono tradurre lingue, analizzare testi, generare conversazioni con i chatbot etc
- Grazie alla loro capacità di comprendere dati testuali complessi, possono identificare entità e relazioni tra di essi, generare nuovo testo coerente e grammaticalmente accurato.



Architettura dei LLM

Queste reti possono predire la prossima parola di un testo ragionevolmente bene, non solo per le regole nel linguaggio umano (come la grammatica) che limitano l'uso di parole in diverse posizioni della frase, ma anche perchè c'è ridondanza nel linguaggio.

Ripetendo questo processo, un transformer può generare un intero passaggio parola per parola.

La grammatica per questi modelli denota come le parole sono utilizzate nel linguaggio, categorizzandole in varie parti del discorso e richiedendo un ordine specifico nella frase.

Il transformer non immagazzina esplicitamente le regole della grammatica, l'acquisiscono invece implicitamente attraverso gli esempi.

Architettura dei LLM

Per gli esseri umani il testo è una collezione di parole: le frasi sono sequenze di parole.

Per i computer, il testo è, di base, una sequenza di caratteri.

Con i modelli basati su reti ricorrenti, le parole vengono processate a blocchi per dare un output, ma a volte «dimentica» cosa è successo all'inizio della sequenza quando siamo alla fine.

Vaswani et al. (2017): "Attention is All You Need" introduce il meccanismo attenzionale che consente di vedere l'intera frase (ma anche paragrafi) tutta insieme piuttosto che una parola per volta.

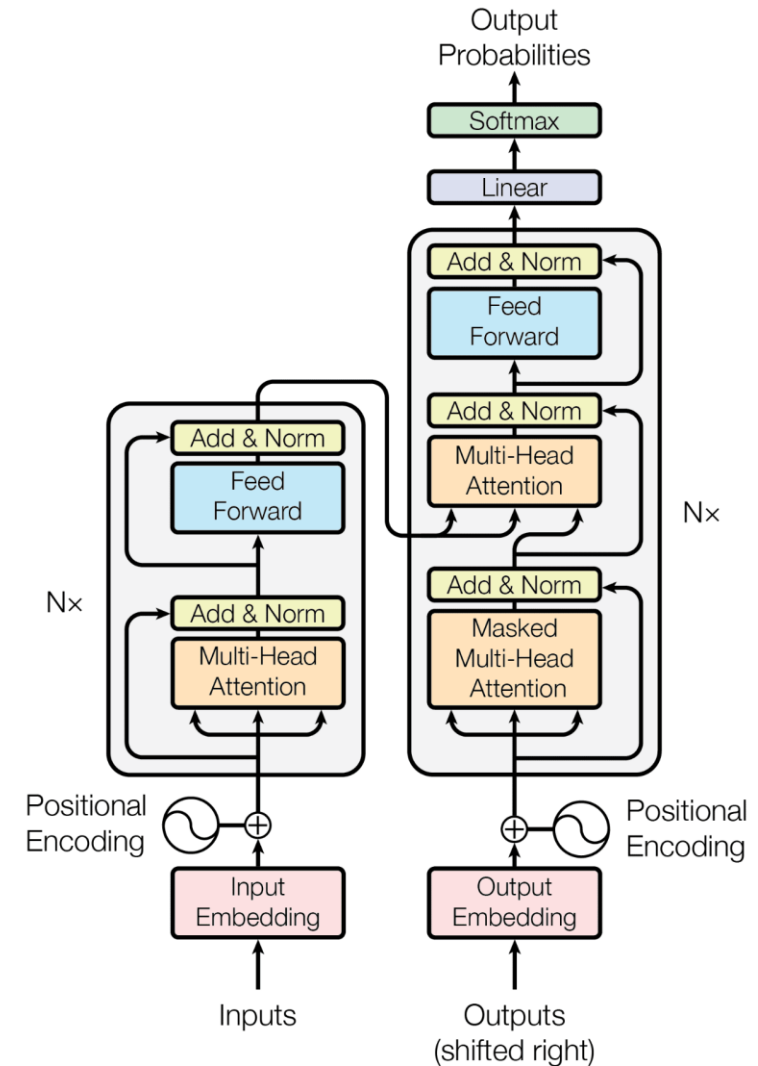
Per processare un input testuale con un transformer, si deve dividere la sequenza in token che vengono codificate in numeri e convertite in embedding (rappresentazioni vettoriali del token che ne mantengono il significato). Poi l'encoder del transformer trasforma l'embedding di tutti i token in un vettore del contesto.

Architettura dei LLM

Questo vettore di contesto rappresenta l'essenza dell'intero input. A partire da esso, il decoder genera l'output basandosi sulle tracce. Ad esempio, si può fornire l'input originale come traccia e lasciare che il decoder dia come output la parola successiva che segue naturalmente. Passo per passo si può produrre un intero paragrafo, partendo da un frase guida.

Questo processo è chiamato generazione autoregressiva.

Nei LLM il transformer può trattare un input testuale molto lungo; il vettore di contesto è così grande da gestire input molto complessi e con molti strati nell'encoder e decoder.



Architettura dei LLM



Generative Pre-trained
Transformer



Bidirectional Encoder
Representations from Transformers

L'architettura dei Large Language Models (LLM) è costituita da diversi livelli di reti neurali che lavorano insieme per elaborare il testo di input e generare previsioni di output. Ecco le parti più importanti dell'architettura degli LLM:

- Il livello di incorporamento: questo livello converte ogni parola nel testo di input in una rappresentazione vettoriale ad alta dimensione. Gli incorporamenti catturano informazioni semantiche e sintattiche sulle parole e aiutano il modello a comprendere il contesto.
- I livelli di feedforward: questi livelli sono composti da più livelli completamente connessi che applicano trasformazioni non lineari agli incorporamenti di input. I livelli di feedforward aiutano il modello ad apprendere astrazioni di livello superiore dal testo di input.
- I livelli ricorrenti: questi livelli sono progettati per interpretare le informazioni dal testo di input in sequenza. I livelli ricorrenti mantengono uno stato nascosto che viene aggiornato a ogni passaggio temporale, consentendo al modello di catturare le dipendenze tra le parole in una frase.
- Il meccanismo di attenzione: questo meccanismo è un'altra parte importante degli LLM, che consente al modello di concentrarsi selettivamente su diverse parti del testo di input. Il meccanismo di attenzione consente al modello di occuparsi delle parti più rilevanti del testo di input e di generare previsioni più accurate.

Architettura dei LLM

Example of input text

As she said this, she looked down at her hands, and was surprised to find that she had put on one of the rabbit's little gloves while she was talking. "How can I have done that?" thought she, "I must be growing small again." She got up and went to the table to measure herself by it, and found that, as nearly as she could guess, she was now about two feet high, and was going on shrinking rapidly: so on she found out that the reason of it was the nosegay she held in her hand: she dropped it hastily, just in time to save herself from shrinking away altogether, and found that she was now only three inches high.

Tokenized Text

['As', 'she', 'said', 'this', ',', 'she', 'looked', 'down', 'at', 'her', 'hands', ',', 'and', 'was', 'surprised', 'to', 'find', 'that', 'she', 'had', 'put', 'on', 'one', 'of', 'the', 'rabbit', "'s", 'little', 'gloves', 'while', 'she', 'was', 'talking', '.', '"', 'How', 'can', 'I', 'have', 'done', 'that', '?"', 'thought', 'she', ',', '"', 'I', 'must', 'be', 'growing', 'small', 'again', '.', '"', 'She', 'got', 'up', 'and', 'went', 'to', 'the', 'table', 'to', 'measure', 'herself', 'by', 'it', ',', 'and', 'found', 'that', ',', 'as', 'nearly', 'as', 'she', 'could', 'guess', ',', 'she', 'was', 'now', 'about', 'two', 'feet', 'high', ',', 'and', 'was', 'going', 'on', 'shrinking', 'rapidly', ':', 'soon', 'she', 'found', 'out', 'that', 'the', 'reason', 'of', 'it', 'was', 'the', 'nose', 'gay', 'she', 'held', 'in', 'her', 'hand', ':', 'she', 'dropped', 'it', 'hastily', ',', 'just', 'in', 'time', 'to', 'save', 'herself', 'from', 'shrinking', 'away', 'altogether', ',', 'and', 'found', 'that', 'she', 'was', 'now', 'only', 'three', 'inches', 'high', '.']

Embedding of the Above Text

```
[ 2.49 0.22 -0.36 -1.55 0.22 -2.45 2.65 -1.6 -0.14 2.26
-1.26 -0.61 -0.61 -1.89 -1.87 -0.16 3.34 -2.67 0.42 -1.71
...
2.91 -0.77 0.13 -0.24 0.63 -0.26 2.47 -1.22 -1.67 1.63
1.13 0.03 -0.68 0.8 1.88 3.05 -0.82 0.09 0.48 0.33]
```

Architettura dei LLM



Un LLM è un transformer che lavora su larga scala che è stato addestrato con un enorme quantità di testo perché acquisiscano i pattern e le strutture del linguaggio.

GPT: ChatGPT

È stato addestrato su una quantità di dati di testo enorme da internet (libri, articoli, siti et.). Il modello, durante il training, apprende le relazioni statistiche tra le parole, le frasi, i paragrafi in modo che possa generare risposte coerenti e contestualmente rilevanti in risposta ad una richiesta o un prompt.

GPT-OpenAI, LLaMA-Meta, PaLM-from Google.

Meta AI

Llama 2 was trained on 40% more data than Llama 1, and has double the context length.


Research Blog Resources About

Learn more about Llama 2

Llama 2		
MODEL SIZE (PARAMETERS)	PRETRAINED	FINE-TUNED FOR CHAT USE CASES
7B	Model architecture: Pretraining Tokens: 2 Trillion Context Length: 4096	Data collection for helpfulness and safety:
13B		Supervised fine-tuning: Over 100,000
70B		Human Preferences: Over 1,000,000




TRANSFORMER-BASED DEEP NEURAL LANGUAGE MODELING FOR CONSTRUCT-SPECIFIC AUTOMATIC ITEM GENERATION

BJÖRN E. HOMMEL 

LEIPZIG UNIVERSITY



MAGNOLIA PSYCHOMETRICS GMBH

FRANZ- JOSEF M. WOLLANG 

MAGNOLIA PSYCHOMETRICS GMBH

VERONIKA KOTOVA 

TECHNICAL UNIVERSITY OF MUNICH

HANNES ZACHER  AND STEFAN C. SCHMUKLE 

LEIPZIG UNIVERSITY

Algorithmic automatic item generation can be used to obtain large quantities of cognitive items in the domains of knowledge and aptitude testing. However, conventional item models used by template-based automatic item generation techniques are not ideal for the creation of items for non-cognitive constructs. Progress in this area has been made recently by employing long short-term memory recurrent neural networks to produce word sequences that syntactically resemble items typically found in personality questionnaires. To date, such items have been produced unconditionally, without the possibility of selectively targeting personality domains. In this article, we offer a brief synopsis on past developments in natural language processing and explain why the automatic generation of construct-specific items has become attainable only due to recent technological progress. We propose that pre-trained causal transformer models can be fine-tuned to achieve this task using implicit parameterization in conjunction with conditional generation. We demonstrate this method in a tutorial-like fashion and finally compare aspects of validity in human- and machine-authored items using empirical data. Our study finds that approximately two-thirds of the automatically generated items show good psychometric properties (factor loadings above .40) and that one-third even have properties equivalent to established and highly curated human-authored items. Our work thus demonstrates the practical use of deep neural networks for non-cognitive automatic item generation.

Key words: automatic item generation, natural language processing, deep learning, neural networks, language modeling.

[CC BY-SA](#)

Generazione automatica di item



- Negli ultimi anni l'AIG è diventata un campo di ricerca promettente in ambito cognitivo, ad esempio sono stati generate nuove matrici progressive di Raven
- Non ci sono stati molti tentativi per la generazione di item per misurare costrutti non cognitivi come i tratti di personalità
- I progressi nel campo del deep learning e del natural language processing (NLP) hanno reso possibile affrontare queste sfide

Generazione automatica di item

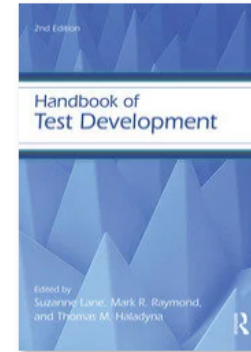


- von Davier (2018) ha dimostrato che gli item di personalità possono essere generati addestrando una long short-term memory (LSTM) network, una rete neurale ricorrente su delle affermazioni stabilite di personalità.
- Il limite di questo lavoro è che non permette la generazione di item specifici per un certo costrutto...problema centrale per la costruzione dei test

von Davier, M. (2018). Automated Item Generation with Recurrent Neural Networks. *Psychometrika*, 83(4), 847–857.
<https://doi.org/10.1007/s11336-018-9608-y>

Sfide nella generazione automatica degli item

- Processo a 3 step in ambito cognitivo (Gierl & Lai, 2015)
 - 1) modello concettuale
 - 2) modello formativo degli item (item stem, opzioni di risposta, elementi placeholder)
 - 3) gli item sono assemblati combinando le possibili variazioni di opzioni ed elementi di input



Chapter

Automatic Item Generation

By Mark J. Gierl, Hollis Lai

Book [Handbook of Test Development](#)

Edition	2nd Edition
First Published	2015
Imprint	Routledge
Pages	20
eBook ISBN	9780203102961

Sfide nella generazione automatica degli item

- In ambito non-cognitivo la situazione è un pò diversa perchè sono spesso usate le scale di rating, in cui la costruzione degli item non può essere basata su un template
- Consideriamo un item come “sono l’anima della festa” (item dell’International Personality Item Pool, IPIP) per l’estroversione
 - Se consideriamo le single parole, non c’è una diretta associazione con il costrutto estroversione
 - Un sinonimo di “anima” o “festa” perfettamente intercambiabile non è facile da trovare

	Strongly Disagree	Somewhat Disagree	No Opinion	Somewhat Agree	Strongly Agree
I am easygoing.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I have high standards.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I enjoy time alone.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I work well with others.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I dislike confrontation.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I prefer crowds over intimacy.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Requisiti che dovrebbe avere un metodo per l'AIG di item non-cognitivi

- La variabile latent che ci interessa deve essere linguisticamente codificata in una sequenza di parola (validità di contenuto)
- La sequenza deve essere organizzata sintatticamente in modo da essere coerente con il linguaggio naturale
- La sequenza deve elicitarle risposte attendibili e valide
- Le sequenze devono essere segmentate in unità significative di lunghezza adeguata, una singola frase breve

Spesso serve un salto inferenziale interpretativo per unire item e costrutto

Approcci non template-based:
SINTASSI
SEMANTICA
SEQUENZA

Approcci di modellazione del linguaggio alla generazione automatica di elementi specifici del costrutto (basati sulla sequenza)

- In linea di principio, il problema dell'AIG degli item di personalità può essere posto come un problema di modellazione del linguaggio: una funzione, o un algoritmo per l'apprendimento di tale funzione, che cattura le caratteristiche statistiche salienti della distribuzione delle sequenze di parole in un linguaggio naturale, tipicamente consentendo di fare previsioni probabilistiche della parola successiva date quelle precedenti.
- Tali modelli sono spesso impiegati per risolvere una varietà di compiti di NLP, come ad esempio traduzione automatica, riconoscimento vocale, sistemi di dialogo e riepilogo del testo.

Approcci di modellazione del linguaggio alla generazione automatica di elementi specifici del costrutto (basati sulla sequenza)

- Il problema dell'AIG specifico del costrutto come l'inverso problema della sintesi del testo. Invece di catturare l'essenza semantica di un testo e producendone una versione più breve e concisa, si fa l'inverso: espandere un concetto espresso da una breve sequenza di parole o anche da una singola parola (ad esempio, "estroversione") in una sequenza testuale più lunga e fortemente rappresentativa

CONCEPT ELABORATION

- La predizione però non è basata sul fattore latente, sul costrutto, ma sulle parole

Approcci di modellazione del linguaggio alla generazione automatica di elementi specifici del costrutto (basati sulla sequenza)

- ✓ Modelli n-gram (catene di Markov)
- ✓ Semantica distribuita
- ✓ Reti ricorrenti o ricorsive e Long Short-Term Memory Networks
- ✓ Modelli Transformer con Attenzione

Approcci di modellazione del linguaggio alla generazione automatica di elementi specifici del costrutto (basati sulla sequenza)

✓ **Modelli n-gram (catene di Markov)**

Questi modelli calcolano la probabilità condizionale che una parola sia presente in un testo in base alle altre parole presenti.

I modelli n-gram sono usati per la loro semplicità, ma hanno problemi relativi alla grandezza dei chunk di testo:

- Chunk troppo piccoli danno previsioni non accurate
- Chunk più grandi portano ad una minore probabilità di trovare le sequenze nel testo, con il risultato di avere dati mancanti

Inoltre tendono a trascurare informazioni che non sono nell'immediata vicinanza della parola target e non sono in grado di generare nuove affermazioni

Approcci di modellazione del linguaggio alla generazione automatica di elementi specifici del costrutto (basati sulla sequenza)

✓ **Semantica distribuita**

Il word embedding rappresenta il significato delle parole mappandola in uno spazio semantico ad alta dimensionalità che deriva dalla valutazione delle parole vicine

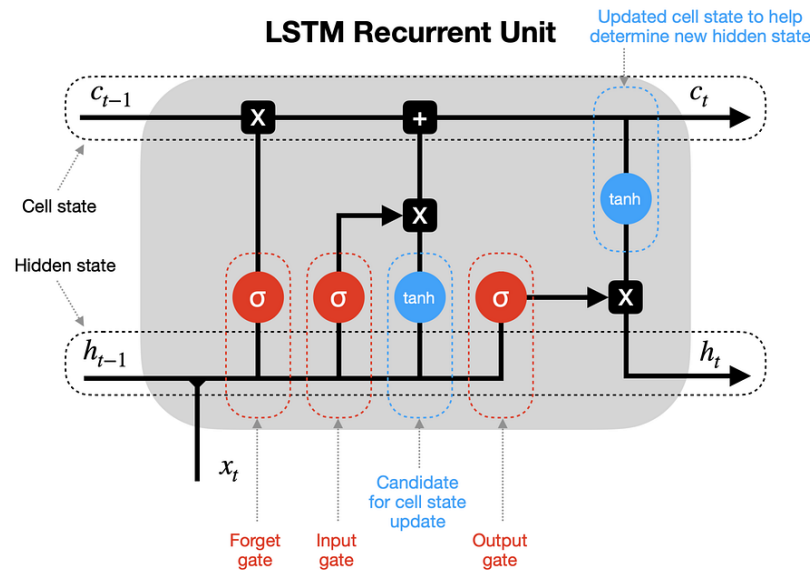
Il loro punto di forza è che possono apprendere in modo semi-supervisionato (usano il testo grezzo e non un input labeled)

Anche in questi modelli, il contesto non è adeguatamente rappresentato

Sono inglobati nelle architetture più moderne

LSTM: long short-term memory network

LONG SHORT-TERM MEMORY NEURAL NETWORKS

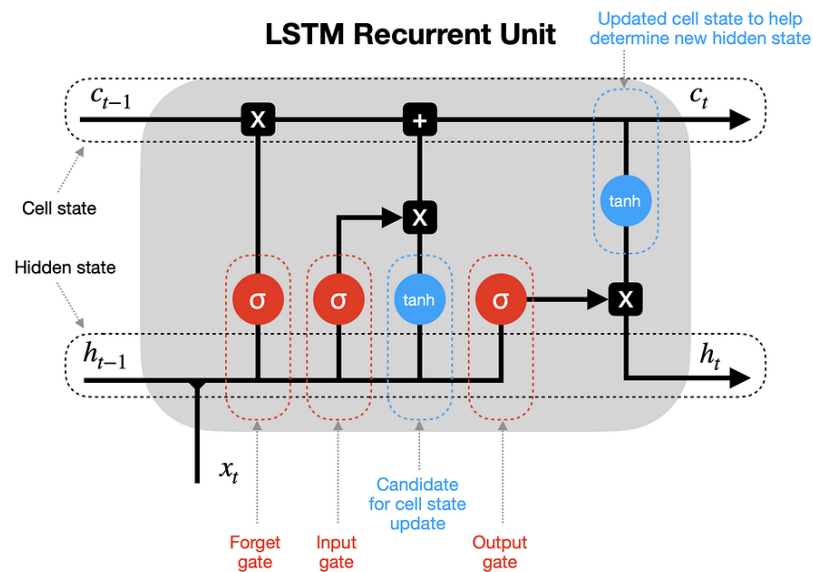


- Le reti neurali ricorrenti standard (RNN) soffrono di un problema di degradazione della memoria che emerge quando si lavora con sequenze di dati più lunghe.
- Le LSTM cercano di risolvere i problemi delle RNN.
- Vengono utilizzate prevalentemente per apprendere, elaborare e classificare dati sequenziali perché queste reti possono apprendere le dipendenze a lungo termine tra i passaggi temporali dei dati. Le applicazioni LSTM comuni includono, la modellazione del linguaggio, il riconoscimento vocale e l'analisi video.

- <https://it.mathworks.com/discovery/lstm.html>

LSTM: long short-term memory network

LONG SHORT-TERM MEMORY NEURAL NETWORKS



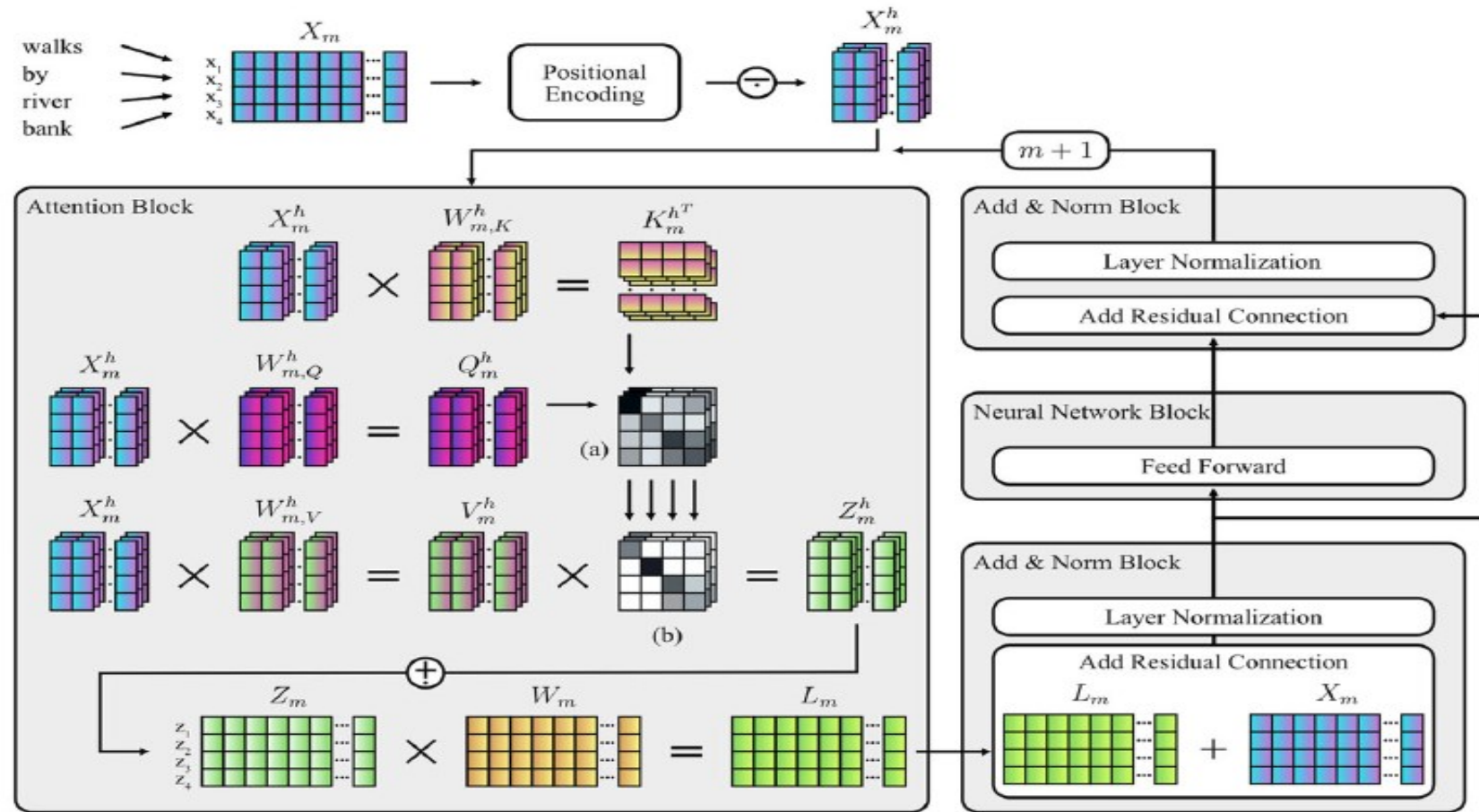
- Hanno una struttura specializzata che aiuta il modello ad imparare quale informazione ricordare e quale dimenticare, attraverso i gate che aggiungono o tolgono informazione dallo stato

Modello proposto dagli autori

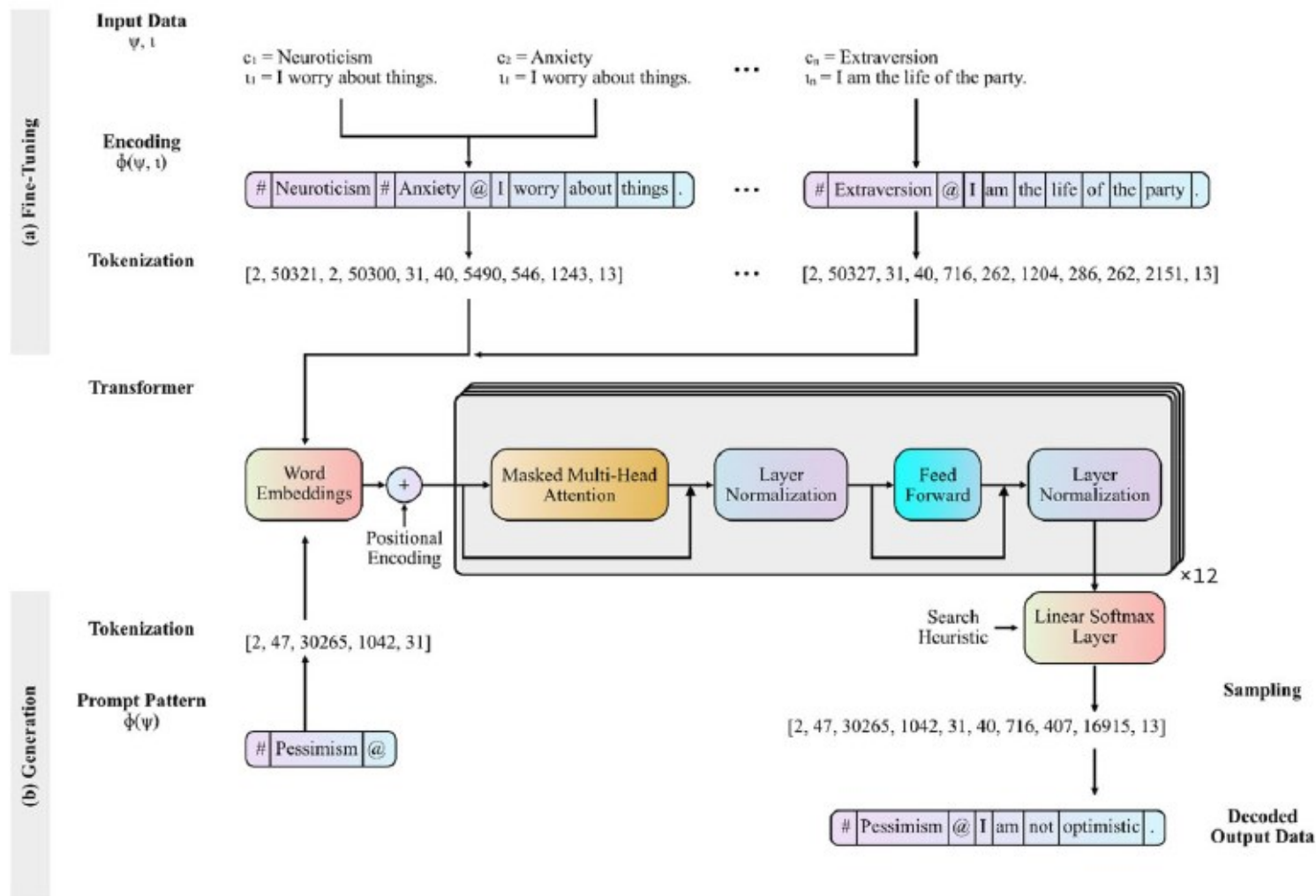
- È basato sui transformers che abbiamo visto prima
- È comune lavorare su implementazione dei modelli che sono stati pre-addestrati su enormi campioni di linguaggio generale.
- Si procede poi ad un addestramento supplementare su un dataset più specifico (fine-tuning)
- Partono da GPT-2



Modello proposto dagli autori



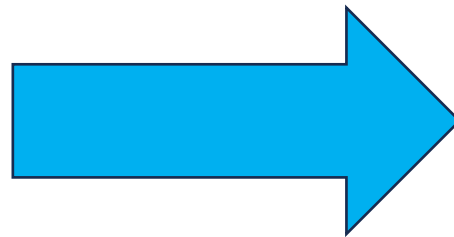
Modello proposto dagli autori



Studio empirico

- Confronto tra item generati da umani e dall'algoritmo per un questionario in una survey online

TRAINING



Modello pre-addestrato di GPT-2 con 355 milioni di parametri da sottoporre al fine-tuning per AIG



Da 4452 item stem e 246 costrutti nell'International Personality Item Pool sono stati selezionati 1715 item stem raggruppati per costrutti



Training set per il fine-tuning: è stato dato in pasto al modello come stringhe concatenate di label di costrutti e item stem



400 cicli di training, learning rate 5e-04

Studio empirico

Fase Generativa:

- il modello genera item stem per 2 set di etichette di costrutti
- Il primo consisteva di 5 etichette di costrutti addestrati (openness to experience, conscientiousness, extraversion, agreeableness, and neuroticism) introdotti nel modello durante il fine-tuning.
- Il secondo di 5 etichette di costrutti non addestrati (i.e., benevolence, egalitarianism, egoism, joviality, and pessimism) NON introdotti nel modello durante il fine-tuning.
- 1.360 item stem generati

Studio empirico

Per la validità di contenuto

- Omessi gli item duplicati e quelli etichettati con più di un costrutto
- Item vengono valutati per la validità di contenuto da 2 esperti
- Accordo di .72 (95% CI [.64, .80]) Cohen's kappa.
- 151 (53.4%) item sono stati approvati da entrambi i valutatori

Questionario

- Un questionario tipo Likert con gli item generati dagli umani e dall'algoritmo
- Dai 151 approvati, sono stati estratti 5 item per ogni costrutto.
- Selezione random di 50 item nel questionario per evitare l'affaticamento dei rispondenti
- Per gli item "umani" 25 item dal BFI dataset (Goldberg, 1999, formato da item presi da IPIP e collegato ai Big Five)

Studio empirico

- Questionario finale: 75 item (“umani” più algoritmo), scale Likert a 5 punti - online survey.
- 273 partecipanti
- Presentazione item in ordine random
- Esclusione dei rispondenti (velocità)
- 220 rispondenti

Risultati

TABLE 1.
Comparison of Confirmatory Factor Analyses of Human- and Machine-authored Scales for Trained Construct Labels

Scale	Human-authored						Machine-authored						<i>p</i>
	CFI	RMSEA	λ_{mean}	λ_{range}	ω	ω_{CI}	CFI	RMSEA	λ_{mean}	λ_{range}	ω	ω_{CI}	
Openness to experience	.95	.14	.62	[.82, .72]	.72	[.65, .78]	.95	.10	.54	[.44, .75]	.66	[.66, .58]	.097
Conscientiousness	.93	.23	.72	[.74, .81]	.81	[.76, .85]	1.00	.00	.44	[.15, .69]	.46	[.46, .36]	< .001
Extraversion	.98	.15	.77	[.89, .86]	.86	[.82, .89]	1.00	.05	.67	[.34, .90]	.75	[.75, .68]	< .001
Agreeableness	.96	.17	.73	[.86, .80]	.80	[.75, .85]	.80	.27	.58	[.35, .87]	.63	[.63, .49]	< .001
Neuroticism	.99	.13	.80	[.91, .87]	.87	[.84, .90]	.98	.17	.56	[.02, .92]	.70	[.70, .61]	< .001

Note. $N = 220$ respondents. λ_{mean} = Mean of standardized factor loadings; λ_{range} = Range of standardized factor loadings; ω = Omega coefficient of internal consistency; ω_{CI} = percentile bootstrapped 95% confidence interval for omega coefficient. p = bootstrapped probability of models' differences in omega coefficients ($K = 5,000$ bootstrapped resamples; data from $k = 446$ iterations were omitted due to failed model convergence).

Risultati

TABLE 2.
Descriptive Statistics and Factor Loadings of Machine-authored Items for Trained Construct Labels

Item	<i>M</i>	<i>SD</i>	Frequencies					Skewness	Kurtosis	λ	ϵ	λ_{human}
			1	2	3	4	5					
I can enjoy a wide variety of musical styles. (OPE+)	4.10	1.05	7	13	30	71	99	-1.16	0.76	.62		1
I like to be surprised. (OPE+)	3.13	1.32	32	39	61	45	43	-0.10	-1.08	.36		0
I love to contemplate the universe and its beauty. (OPE+)	3.94	1.12	9	15	46	60	90	-0.87	-0.04	.65		1
I like to be with people who are different from myself. (OPE+)	3.50	1.06	9	25	75	68	43	-0.32	-0.43	.35		0
I am not a fan of change. (OPE-)	3.11	1.32	29	47	61	36	47	0.00	-1.13	.35		0
I am not always on time for work. (CON-)	4.01	1.28	12	28	21	43	116	-1.02	-0.29	.53		0
I know that I make many mistakes. (CON-)	2.53	1.20	53	61	57	35	14	0.35	-0.84	.20		0
I work too hard. (CON+)	3.17	1.28	25	45	62	44	44	-0.07	-1.05	.55		0
I do not like to read or study. (CON-)	4.23	1.04	8	8	27	59	118	-1.44	1.55	.54		0
I am not concerned with details. (CON-)	4.27	0.95	4	10	23	68	115	-1.39	1.57	.65		0
I am able to speak confidently. (EXT+)	3.96	1.11	8	18	37	69	88	-0.92	0.06	.84		1
I avoid public places. (EXT-)	3.50	1.28	21	31	44	66	58	-0.50	-0.85	.46		0
I am able to handle myself in a crowd. (EXT+)	3.98	1.07	8	15	34	79	84	-1.02	0.44	.73		1
I do not like to talk about myself. (EXT-)	2.59	1.25	50	65	52	32	21	0.41	-0.84	.45		0
I am able to hold my own in a discussion. (EXT+)	4.16	0.97	6	11	19	90	94	-1.37	1.74	.60		1
I care a lot about others. (AGR+)	4.25	0.92	4	5	34	67	110	-1.23	1.30	.87		1
I am easily angered. (AGR-)	3.96	1.17	11	19	31	65	94	-1.00	0.07	.39		0
I don't like to argue. (AGR+)	3.95	1.14	10	17	38	65	90	-0.94	0.05	.23		0
I am not easily offended. (AGR+)	3.43	1.25	16	45	38	71	50	-0.36	-1.00	.24		0
I am not a nice person. (AGR-)	4.51	0.84	3	5	17	47	148	-1.95	3.83	.79		1
I am generally happy and content. (NEU-)	2.15	1.19	82	70	36	18	14	0.91	-0.07	.72		0
I am often upset by minor things. (NEU+)	2.30	1.23	72	69	33	34	12	0.64	-0.71	.89		1
I am a person who is easily moved by the good moods and bad moods of others. (NEU+)	3.47	1.23	22	24	52	73	49	-0.55	-0.63	.28		0
I am generally cheerful and optimistic. (NEU-)	2.30	1.26	72	67	43	18	20	0.76	-0.42	.69		0
I seldom feel scared. (NEU-)	3.01	1.28	30	57	45	56	32	0.00	-1.15	.38		0

Note. Based on data from $N = 220$ respondents. λ = Standardized factor loading in a CFA model with the five human-authored items and the respective machine-authored item; ϵ λ_{human} = Factor loading of respective machine-authored item within the range of factor loadings for human-authored scales (1 = within the range); OPE = Openness to experience; CON = Conscientiousness; EXT = Extraversion; AGR = Agreeableness; NEU = Neuroticism; +/- indicates positive or negative keying.

Risultati

TABLE 3.
Goodness of Fit Statistics, Factor Loadings and Reliability Estimates of Confirmatory Factor Analyses of Machine-authored Scales for Untrained Construct Labels

Scale	CFI	RMSEA	λ_{mean}	λ_{range}	ω	ω_{CI}
Benevolence	1.00	.05	.69	[.49, .94]	.74	[.67, .79]
Egalitarianism	.99	.09	.76	[.67, .87]	.78	[.69, .85]
Egoism	.90	.12	.44	[.08, .85]	.58	[.47, .67]
Joviality	.83	.16	.44	[.17, .92]	.54	[.42, .62]
Pessimism	.99	.11	.70	[.45, .93]	.82	[.77, .86]

Note. $N = 220$ respondents. λ_{mean} = Mean of standardized factor loadings; λ_{range} = Range of standardized factor loadings; ω = Omega total coefficient of internal consistency; ω_{CI} = bootstrapped 95% confidence interval for omega coefficient, based on $K = 5,000$ bootstrap iterations.

Risultati

TABLE 4.
Descriptive Statistics and Factor Loadings of Machine-authored Items for Untrained Construct Labels

Item	<i>M</i>	<i>SD</i>	Frequencies					Skewness	Kurtosis	λ
			1	2	3	4	5			
I care about others' well-being. (BEN+)	4.41	0.76	2	1	21	77	119	-1.40	2.61	.78
I forgive others. (BEN+)	3.85	1.09	9	19	39	82	71	-0.85	0.05	.55
I am not a person who would do anything nice for anyone. (BEN-)	4.57	0.79	2	6	12	44	156	-2.15	4.71	.66
I have little sympathy for poor people. (BEN-)	4.17	1.23	14	17	16	44	129	-1.38	0.70	.49
I am not interested in others feelings. (BEN-)	4.30	0.98	4	11	25	55	125	-1.41	1.36	.94
I believe that the rights of others should be treated equally. (EGA+)	4.72	0.59	1	2	4	43	170	-2.78	10.18	.87
I believe that all races are created equal. (EGA+)	4.60	0.89	6	4	13	27	170	-2.52	6.09	.71
I believe that it is wrong to exploit others for your own gain. (EGA+)	4.52	0.92	7	5	9	44	155	-2.35	5.37	.67
I believe in the equality of all peoples. (EGA+)	4.65	0.72	2	3	11	38	166	-2.49	6.95	.81
I believe that the rights of others should be respected without question. (EGA+)	4.35	0.84	2	6	22	72	118	-1.38	1.88	.77
I believe that I have the right to my own way of life. (EGO+)	4.45	0.72	2	1	15	79	123	-1.57	3.69	.08
I often exaggerate my achievements. (EGO+)	1.94	1.11	97	74	25	13	11	1.24	0.84	.26
I believe that I am the best. (EGO+)	2.57	1.35	67	44	50	35	24	0.34	-1.11	.85
I believe that I have more power than others. (EGO+)	2.20	1.17	78	63	46	22	11	0.71	-0.41	.60
I am not overly proud of my achievements. (EGO-)	3.28	1.32	26	41	49	54	50	-0.23	-1.11	.39
I am very jovial. (JOV+)	3.37	1.18	15	37	65	57	46	-0.24	-0.83	.92
I do things that are not fun. (JOV-)	3.34	1.23	16	41	69	41	53	-0.12	-0.99	.17
I sometimes laugh out loud. (JOV+)	4.33	0.93	4	11	13	73	119	-1.61	2.39	.18
I am never sad. (JOV+)	1.92	1.14	106	61	26	18	9	1.15	0.40	.39
I am easily entertained. (JOV+)	3.62	1.06	12	17	58	88	45	-0.68	0.05	.55
I am not likely to succeed in my goals. (PES+)	1.90	1.13	110	54	33	14	9	1.15	0.46	.71
I can see that things are never going to be the way I want them to be. (PES+)	2.72	1.33	51	50	57	33	29	0.26	-1.05	.52
I am not optimistic. (PES+)	2.09	1.28	103	49	26	29	13	0.88	-0.51	.93
I am always on the lookout for a better way. (PES-)	1.99	0.97	79	83	44	9	5	0.90	0.55	.45
I look at the bright side. (PES-)	2.23	1.25	79	69	32	23	17	0.83	-0.39	0.90

Note. Based on data from $N = 220$ respondents. λ = Standardized factor loadings in a CFA model including the five machine-authored items of the respective dimension; BEN = Benevolence; EGA = Egalitarianism; EGO = Egoism; JOV = Joviality; PES = Pessimism; +/- indicates positive or negative keying.



Discussione

- Modello di partenza: transformer pre-addestrato GPT-2, fine-tuning su International Personality Item Pool.
- É possibile generare automaticamente item specifici per costrutto, allineando item stem a costrutti specifici
- A GPT-2 è stato insegnato un pattern di stringhe concatenate di affermazioni di personalità con etichette corrispondenti ai costrutti per cui gli item sono stati concettualizzati
- Imparando questi pattern, il modello risponde generando item stem validi quando avviati dall'etichetta per un certo costrutto



- Importanza dei LM che imparano le relazioni tra parole al di là della prossimità: i transformer sono eccellenti nelle dipendenze a lunga distanza
- Lo studio empirico confronta i vari gruppi di item: item generati for construct labels che GPT-2 ha appreso, item per construct label non appresi e item scritti da umani
- I risultati sono, tutto sommato, incoraggianti: circa 2/3 degli item generati automaticamente mostrano buone proprietà psicometriche (factor loading maggiori di .40) e 1/3 ha proprietà equivalenti a quelli degli item curati da umani
- Strada aperta per altri tentativi di fine-tuning...
- Utilizzo come supporto nel processo di costruzione e validazione dei test (non dipendere dagli esperti)
- Molti item sono utili per il CAT



GRAZIE PER L'ATTENZIONE!