Word Embedding: osservazioni su relazioni linguistiche apprese da una RNN

Jacopo D'Ignazi

September 11, 2020

Abstract

Nel mio progetto mi sono occupato del Word Embedding, svolto tramite una rete neurale ricorsiva CBOW ed Unsupervised Training. L'obbiettivo della ricerca è stato cercare di riprodurre le proprietà osservate dei vettori di parole [2], dunque studiare le relazioni algebriche tra i vettori appresi in rapporto al variare parametri del modello .

Il focus dello studio non sarà ottenere necessariamente una "ottima" rappresentazione vettoriale, quanto di ottenerne una "buona" ragionando sul metodo per raggiungerla e sui vettori in sè: si vedrà che le osservazioni svolte in diverse iterazioni del modello, hanno permesso sia di riprodurre consistentemente le proprietà sopracitate, che un ragionamento su come interpretare il comportamento della rete.

I. Introduzione

Il Word Embedding viene utilizzato per permettere alle reti neurali di lavorare su task riguardanti il linguaggio. Nell'ultimo decennio, si è notato come il trasformare le singole parole un un vettore denso, fornisse migliori risultati -rispetto ad una tecnica One-Hot-Encoding- nel momento in cui tali vettori venivano usati in task tipici del NLP. Questo risultato è da attribuirsi al fatto la metrica di uno spazio vettoriale può "codificare" relazioni tra le parole, in particolare di tipo semantico (riguardante le categorie di contenuto) e sintattico (riguardante le categorie grammaticali) [2].

Con questa ricerca ho cercato di capire in che modo, operativamente, una RNN riesca a codificare queste relazioni. Contemporaneamente, ho cercato di capire quale fosse una strategia efficace per "interrogare" l'algoritmo.

Nel paragrafo 2 si darà una spiegazione dell'architettura del programma e della RNN usata, quindi dei metodi di analisi e dei parametri coinvolti nel preprocessing e training. Nel paragrafo 3 si raccoglieranno osservazioni sulle iterazioni realizzate, e nel paragrafo 4 si ragiona su queste osservationi, riflette su quali altre analisi possono essere condotte e per quali motivi.

II. Metodo

Ho allenato il modello su due diversi dataset (separatamente), qualitativamente molto diversi tra loro. Il Lee Corpus, raccolta di 300 articoli di news sulla guerra in medioriente; ed "enwiki8", una raccolta dei primi 10^8 caratteri di Wikipedia inglese. Il primo è relativamente piccolo e fortemente contestuale, il secondo più grande e di natura più generica. Nel preprocessing ho fatto sì che si potesse scegliere se eliminare la punteggiatura, eliminare le parole più comuni, ridurre o meno le parole alla loro forma base ("was" diventa "be", "houses" diventa "house").

La RNN usata è del tipo CBOW descritta in [1]: un vocabolario di parole in forma one-hot-encoding viene esercitato a predire ogni parola delle parole prese nel contesto -parole intorno a quella di target, in una finestra di dimensione a scelta- mediante una rete fully-connected con un solo hidden layer. I parametri tra l'input layer e l'hidden layer saranno i valori del vettore embedded. La parola predetta dalla rete viene valutata dall'hidden layer all'output layer con soft-max ed i suoi valori sono aggiornati con Negative Sampling descritto in [3] per minimizzare la log-likelihood. Di quest'architettura, ho manipolato prevalemtemente il numero di neuroni nel layer interno (dunque la grandezza dello spazio vettoriale), la grandezza della finestra di parole, il numero di occorrenze minimo di una parola per essere vettorializzata, e la grandezza delle "batch" in cui suddividere il dataset.

Tutte le analisi dei vettori sono poi state condotte facendo uso della cosine similarity

$$similarity(\vec{x}, \vec{y}) = \cos(\theta_{xy}) = \frac{\vec{x} * \vec{y}}{\|\vec{x}\| \|\vec{y}\|} \qquad (1)$$

che fondamentalmente valuta la differenza di angolo solido tra due vettori; si considereranno dunque "simili" due vettori che "sono nella stessa direzione". Tramite questa ho poi calcolato le norme euclidee, la similitudine media delle più simili di una parola, e la similitudine media della i-esima parola più simile; da queste ho preso nota di come questi grafici cambiassero al variare dei parametri del modello.

La verifica delle relazioni semantiche è stata svolta con il paradigma suggerito da [2] ma non ho condotto test quantitativi. La "bontà" dell'embedding è stata valutata infine su un piccolo gruppo d'esempi di task tipici, confrontando la similitudine tra valore atteso e valore ottenuto nelle operazioni con la media similitudine tra le parole e i loro simili.

III. Osservazioni

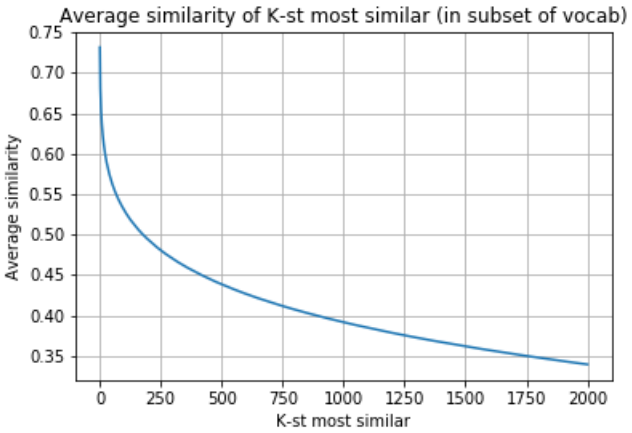
Preliminarmente, ho notato come i risultati dipendessero fortemente dal contenuto del dataset. Ad esempio, la parola più simile a "cave" nel primo dataset era "hideout" (ad intendere che diversi articoli parlassero appunto di attività terroristiche), mentre il dataset di wikipedia restituisce parole di contenuto geologico e/o archeologico. Per questo motivo ho scelto di continuare il training solo sul dataset di wikipedia.

Le analisi sono poi state interamente condotte "esplorando" lo spazio dei word-vectors, quindi cercando di collegare relazioni geometriche a relazioni linguistiche. Prendendo in analisi la norma, ho notato come questa fosse legata alla quantità di training (dunque alla frequenza) delle parole. La similitudine tra parole non dipende dalla norma, ma per "buoni embedding" ho individuato un andamento tipico velocemente decrescente.

Le disposizioni dei vettori variano sensibilmente al variare della grandezza scelta del word-vector e dal numero di parole nel vocabolario: intuitivamente, le parole devono avere "abbastanza spazio da potersi allontanare, ma non troppo da disperdersi". Un maggior tempo di training sembra inoltre enfatizzare certe categorie (riconoscimento dei plurali), pur penalizzandone altre (riconoscimento di genere). Ho pensato questo possa esser dovuto al fatto che è relativamente probabile che in una frase (considerando una window ridotta) si parli sempre coniugando al plurale o al singolare, piuttosto che sempre al maschile o al femminile.

Più significativi per comprendere invece "come le parole sono disposte" sono i grafici sulla similitudine media: dalla media di questi valori si intuisce quando le parole tendono a raggrupparsi in certe regioni di spazio, mentre dalla deviazione standard si intuisce quanto questi gruppi stessi siano dispersi nello spazio. Buoni risultati corrispondono ad una media di circa ~ 0.65 ed una deviazion standard di circa ~ 0.1. Dalla similitudine media con i-sima parola più simile, si intuisce invece quanto i vettori siano clusterizzati: una discesa veloce al variare di k significa "che quella parola è molto simile a poche e poco simile a molte", intuitivamente ragionevole per il linguaggio naturale.

Sulla base di queste considerazioni, sono riuscito a calibrare i parametri del modello in modo da ottenere



buoni risultati su esempi di task tipici, seguendo il metodo di somme e differenze tra vettori ([2]) per "aggiungere o togliere" categorie ad una parola. In alcuni casi tuttavia, le operazioni restituivano una delle parole stesse delle operazioni (vedasi testo in allegato), il che significa che per quanto il modello le distinguesse, l'offset coinvolto fosse più piccolo dell'intorno "vuoto" della parola in questione.

IV. Conclusioni e sviluppi futuri

Il modello, sostanzialmente, avvicina tra loro parole che è più probabile siano pronunciate insieme in una frase. Quel che gli permette di distinguere categorie è tuttavia come queste parole siano "concatenate" tra loro: anche se nel dataset non sono mai capitate nello stesso contesto, è possibile che entrambe siano molto vicine ad una stessa terza parola ed il modello le spingerà "nella stessa direzione".

Per questa ragione, è fondamentale avere una quantità sufficiente di parole nel vocabolario, dunque un adeguato dataset ed una adeguata finestra di parole. Un criterio utile per migliorare le performance può essere: *maggiore è la varietà di combinazioni di parole, maggiore sarà la capacità del modello di distinguere categorie più astratte*.
Per questo motivo, inoltre, il modello riesce meno nei task in cui la categorizzazione è ambigua [1].

Un modo per migliorare le performance in questi casi potrebbe essere proporre al modello una fase iniziale di training più guidata, magari su un dataset limitato di parole comuni. Potrebbe essere interamente knowledge based o supervised, e fornire quindi al modello delle categorie più "delineate" a cui poi avvicinare le altre parole

[1] ovverosia tra parole che possono già esser disposte in più categorie semantiche, ad esempio "houses-house+bird" può restituire "nests"

Per quanto riguarda il programma in sè da me realizzato, sebbene sia riuscito a riprodurre dei comportamenti tipici in termini qualitativi, non ho ancora avuto modo di verificarli quantitativamente. Uno sviluppo necessario è quindi quello di aggiungere diverse tipologie di task su cui testare il modello, ed usarne i risultati per calibrare bene i parametri ed eventualmente riconsiderare le interpretazioni fatte finora. In particolare, sarebbe utile avere task molto diversi, perché permetterebbero più specificità nel comprendere il modello

Uno sviluppo interessante in chiave più interpretativa, può essere invece vedere effettivamente come le relazioni tra parole evolvano durante il training. O anche, sarebbe curioso realizzare un Clustering (basandosi sulla distanza angolare e non la norma euclidea), per chiedere effettivamente al modello quali categorie principali abbia appreso. Con queste si può anche pensare di realizzare una "base vettoriale" e verificare che possa fungere in qualche modo da "base semantica".

Per concludere, trovo molto meritevole di approfondimento, il fatto in sè che un algoritmo totalmente unsupervised possa apprendere così efficacemente non solo categorie, ma anche relazioni tra queste.

References

[1] Milkov et al,2013 Efficient Estimation of Word Representations in Vector Space *Google Inc.*

[2] Milkov et al,2013 Linguistic Regularities in Continuous SpaceWord Representations *Google Inc.*

[3] Milkov et al,2017 EnrichingWord Vectors with Subword Information *Facebook AI Research*