Imparare a quantificare guardando

Sandro Pezzelle CIMeC
Ionut Sorodoc EM LCT
Aurelie Herbelot CIMeC
Raffaella Bernardi CIMeC, DISI
Università degli Studi di Trento
{nome.cognome@unitn.it}

Abstract

English. In this paper, we focus on linguistic questions over images which may be answered with a quantifier (e.g. *How many dogs are black? Some/most/all of them*, etc.). We show that in order to learn to quantify, a multimodal model has to obtain a genuine understanding of linguistic and visual inputs and of their interaction. We propose a model that extracts a fuzzy representation of the set of the queried objects (e.g. *dogs*) and of the queried property in relation to that set (e.g. *black* with respect to *dogs*), outputting the appropriate quantifier for that relation.

Italiano. *In questo lavoro studiamo le domande del tipo "Quanti cani sono neri?", la cui risposta è un quantificatore (es. "alcuni"/"tutti"/"nessuno"). Mostriamo che al fine di imparare a quantificare, un modello multimodale deve ottenere una rappresentazione profonda della domanda linguistica, dell'immagine e della loro interazione. Proponiamo un modello che estrae una rappresentazione approssimativa dell'insieme degli oggetti e della proprietà sui quali verte la domanda.*

1 Introduzione

La linguistica computazionale ed i sistemi di visione artificiale stanno attraversando un momento particolarmente favorevole, ben rappresentato dallo sviluppo di modelli multimodali capaci di svolgere compiti che sembravano fuori portata fino a pochissimi anni fa, e che adesso si trovano integrati in molte applicazioni destinate all'utilizzo da parte degli utenti. Il compito di "Visual Question Answering" (VQA), finalizzato a rispondere a domande a partire da input visivi, e la generazione automatica di didascalie ("caption generation") sono solo alcuni dei compiti in cui maggiormente si è assistito a un rapido e inarrestabile miglioramento. Oltre a sfociare in dirette applicazioni nel mondo reale, dove i contenuti visivi giocano un ruolo cruciale, la performance di questi modelli di lingua e visione rappresenta anche un indicatore della misura in cui essi riescono a catturare il significato, fornendoci importanti intuizioni teoriche.

I recenti miglioramenti nel campo delle applicazioni VQA sono da attribuire principalmente alla nuova generazione di reti neurali artificiali di apprendimento profondo ("deep learning neural networks"), insieme alla crescente disponibilità di grandi dataset di immagini. Questi modelli hanno dimostrato che anche reti neurali (RN) molto semplici possono catturare interazioni complesse tra le proprietà di un dataset, spesso superando in performance modelli molto più complessi. Ad esempio, (Zhou et al., 2015) ha recentemente dimostrato che un semplice modello bag-of-words (BoW) può ottenere performance all'avanguardia in uno dei più importanti dataset di VQA (Antol et al., 2015). Gli autori dello studio, tuttavia, obiettano che la performance del modello è da attribuire più alla eccellente abilità della rete di memorizzare correlazioni che a una reale capacità di ragionamento e comprensione, e concludono sostenendo la necessità di passare al secondo obiettivo. Il loro studio dimostra anche quanto difficile sia valutare se un modello sia in grado di afferrare realmente il significato di immagini e parole. Nel presente lavoro proponiamo un compito che riteniamo essere utile per la comunità ai fini del raggiungimento di questo obiettivo.

Nel compito di VQA, le valutazioni dei modelli si basano su domande che riguardano le proprietà di oggetti specifici, come la posizione, il colore, ecc., mentre nel compito di generazione di dida-

scalie viene presa in considerazione l'intera immagine. In entrambi i casi, è ovviamente possibile imparare correlazioni tra le parole presenti nella domanda, le parole presenti nella risposta e l'immagine stessa (o parti di essa). Presentata a un modello RN insieme alla domanda *Cosa sta mangiando?*, ad esempio, l'immagine di una persona che mangia una torta attiverà sicuramente l'associazione tra il verbo *mangiare* e proprietà (sia linguistiche che visive) collegate al cibo, producendo la risposta corretta. Per testare un modello al di là di questo semplice meccanismo, emerge la necessità di proporre un compito in cui una particolare associazione tra domanda e immagine non dia come risultato sempre la stessa risposta.

Le domande riguardanti il "numero" sono state studiate solo marginalmente in letteratura, e la performance dei sistemi di VQA su queste domande si è dimostrata abbastanza scarsa (Ren et al., 2015; Antol et al., 2015). Inoltre, i lavori precedenti si sono concentrati quasi esclusivamente sulla modellizzazione di cardinali esatti, analizzando quindi solo parzialmente il fenomeno della quantificazione. Nel presente articolo investighiamo un nuovo tipo di domanda che richiede un certo grado di comprensione dei quantificatori generalizzati (*few, most, all*, ecc.). Il motivo per cui siamo interessati a queste domande è che, per rispondervi, non è sufficiente identificare l'area di un'immagine correlata alla domanda. La domanda *In che proporzione i cani sono neri?*, ad esempio, richiede qualcosa di più dell'identificazione delle proprietà *cane* e *nero* nell'immagine: la rete deve essere in grado di riflettere sulla relazione tra queste proprietà, e di generare uno dei quantificatori, che sono potenzialmente simili tra di loro. L'abilità di identificare i membri di un insieme e le loro proprietà condivise richiede un certo grado di ragionamento più profondo che, sosteniamo, non può essere ottenuto con un semplice meccanismo di memorizzazione.

In ciò che segue, consideriamo il compito di VQA come un problema di "fill in the blank" (riempire uno spazio vuoto con la parola corretta), e poniamo la domanda *In che proporzione i cani sono neri?* nella seguente forma: ___ *cani sono neri*. Le possibili risposte sono selezionate all'interno di un insieme di quantificatori linguistici, ovvero *no, few, some, most* e *all*. Per assegnare il quantificatore corretto, il modello deve essere in grado di porre l'attenzione sugli oggetti rilevanti (il restrittore del quantificatore) e di quantificare gli oggetti che, all'interno di questo dominio ristretto, possiedono la proprietà richiesta (la portata del quantificatore). Mostriamo che un semplice modello BoW non è sufficiente per compiere efficacemente questo compito, e proponiamo un modello RN alternativo e linguisticamente motivato, la cui performance risulta essere superiore al modello di (Zhou et al., 2015).

2 Dati

Abbiamo usato scenari contenenti ognuno sedici immagini estratte da ImageNet. ImageNet ci fornisce immagini annotate manualmente con un'etichetta identificativa dell'oggetto (e.s., *dog*, *wine*, *pizza*, ecc.) e diverse proprietà associate ad esso (e.s., *black*, *furry*, ecc.) Abbiamo selezionato tutte le immagini di ImageNet annotate con almeno una proprietà per un totale di 9600 immagini rappresentanti 203 tipi di oggetti e 24 proprietà. Abbiamo poi selezionato le immagini utili per la costruzione del nostro dataset, mantenendo solo gli oggetti che compaiono in un numero significativo di immagini e che vengono menzionati frequentemente nel corpus UkWaC.[1] Questo ci permette di ottenere rappresentazioni visive e linguistiche adeguate. Applicando questa selezione, abbiamo ottenuto 161 oggetti e 7124 immagini,[2] che sono state successivamente assemblate in scenari "plausibili". A tal fine abbiamo calcolato la probablità con cui due oggetti possano far parte di una stessa scena utilizzando la loro co-occorrenza nelle didascalie disponibili in MS-COCO (Lin et al., 2014) – ad esempio, un cane e un divano hanno più possibilità di co-occorrere in una stessa scena rispetto a un elefante e un divano. Abbiamo usato quindi la seguente misura:

$$PMI(o1, o2) = log \frac{f(o1, o2) * N}{f(o1) * f(o2)} \quad (1)$$

[1] Abbiamo scelto oggetti con almeno 16 immagini e che occorrono nel corpus almeno 150 volte.

[2] Ognuno dei 161 oggetti è rappresentato da una media di 48 immagini uniche (sd=99), con una distribuzione che va da 13 (*pasta*) a 1104 (*dog*). Ogni oggetto è associato a un numero di proprietà che varia da un minimo di 2 (es. *lion*) ad un massimo di 18 (es. *dog*) con una media pari a 8 (sd=3.4). All'interno delle 7124 immagini uniche, la coppia oggetto-proprietà più frequente è *furry dog* con 769 occorrenze, mentre le meno frequenti (es. *pink salmon*) occorrono in una sola immagine (media=13.5, sd=37). Infine, la proprietà più frequente, *furry*, compare in 2936 immagini uniche, seguita da *brown* (2782) e *smooth* (2266). La meno frequente è *violet*, che occorre in 24 immagini. La frequenza media è 801 (sd=837).

in cui $o1$ e $o2$ sono due oggetti, $f(o1, o2)$ conta quante volte le etichette di $o1$ and $o2$ appaiono nelle didascalie delle stesse immagini, $f(o)$ conta quante volte o appare nella didascalia di MS-COCO in totale, e N è il numero delle parole in tutte le didascalie. Le etichette che non sono usate nelle didascalie ricevono un valore uniformemente distribuito in relazione a tutti gli altri oggetti. 10,000 scenari sono stati generati secondo la procedura seguente:

- scegliamo un'etichetta a caso dall'insieme di 161 oggetti (e.s., *dog*);
- scegliamo una proprietà p dall'insieme delle 24 proprietà (e.s., *black*);
- selezioniamo n_1 immagini che contengono oggetti con l'etichetta l e n_2 immagini che contengono oggetti con l'etichetta l e proprietà p, cosi che $0 \leq n_1 \leq 16$ and $0 \leq n_2 \leq n_1$;
- riempiamo le rimanenti celle dello scenario $(16 - n_1)$ con le immagini non etichettate con l ed usando la distanza PMI per scegliere oggetti che possano plausibilmente co-occorrere con l'oggetto target;
- usando la proporzione tra n_1 e n_2, calcoliamo quale quantificatore assegnare allo scenario assemblato, seguendo regole pre-definite: *no* e *all* sono assegnati quando, rispettivamente, nessun n_1 o tutti gli n_1 hanno la proprietà p. Per *most* e *few* usiamo le stime riportate in (Khemlani et al., 2009), e assegniamo il primo quando la proporzione ùguale o superiore al 70%, e il secondo quando la proporzione è al piú 17%. Tutte le proporzioni che cadono tra questi due valori sono assegnate a *some*. Ad esempio, se $n_1 = 6$ oggetti con l'etichetta l, assegniamo *no* a casi in cui $n_2 = 0$, *few* quando $n_2 = 1$ (1/6=0.1667), *some* quando $2 \leq n_2 \leq 4$, *most* quando $n_2 = 5$ $(5/6 = 0.833)$, e *all* quando $n_2 = 6$;
- da l e p generiamo la domanda (es. *How many dogs are black?*).

Come mostrato nell'esempio sopra descritto, nel generare gli scenari abbiamo posto come restrizione che il numero di immagini n_1 etichettate con il restrittore sia uguale o maggiore a 6. Sulla base delle proporzioni che abbiamo usato per definire i quantificatori, infatti, tale numero rappresenta il valore minimo per coprire tutti i 5 casi di quantificazione. Questo significa che i modelli non possono migliorare la loro accuratezza semplicemente imparando la correlazione tra valori bassi di n_1 e la non plausibilità di *few/most* negli scenari associati. Inoltre, gli scenari sono uniformemente distribuiti tra i 5 quantificatori. Di conseguenza, ogni quantificatore descrive circa 2000 scenari.[3]

Rappresentazioni visive Per ogni immagine in ciascuno scenario abbiamo estratto una rappresentazione visiva usando una tecnica che si basa su reti neurali convoluzionali (CNN) (Simonyan and Zisserman, 2014). In particolare, abbiamo usato il modello VGG-19 preaddestrato sui dati di ImageNet ILSVRC (Russakovsky et al., 2015) e il pacchetto MatConvNet (Vedaldi and Lenc, 2015) per l'estrazione delle features. Ogni immagine è rappresentata da un vetttore di 4096 dimensioni estratto dal settimo layer totalmente connesso ("fully-connected layer") e successivamente ridotto a un vettore di 400 dimensioni usando la tecnica SVD. Questi vettori di 400 dimensioni rappresentano l'input visuale dei nostri modelli.

Rappresentazioni linguistiche La domanda è espressa mediante due parole: $parola_1$, il restrittore del quantificatore, e $parola_2$, la sua portata. Ogni parola è rappresentata da un vettore di 400 dimensioni costruito usando l'architettura CBOW del pacchetto word2vec (Mikolov et al., 2013) e i migliori parametri di (Baroni et al., 2014). Il corpus usato per costruire lo spazio semantico, contentente circa 2.8 miliardi di tokens, è una concatenzione di UKWaC, un ampio estratto in inglese di Wikipedia 2009 e il British National Corpus (BNC).

3 Modelli

iBOWIMG è il modello di domanda e risposta su immagini (VQA) di (Zhou et al., 2015) adattato al nostro compito, cioè l'apprendimento dei quantificatori. Le domande sono rappresentate da un vettore che mette insieme tutte le parole del vocabolario indicando con "1" quelle presenti ("one-hot bag-of-words") ed elaborando un vettore basato su proprietà salienti delle parole ("word feature embedding"). La rappresentazione linguistica è

[3] Sia i codici che il dataset usati nel presente lavoro saranno resi disponibili per studi successivi.

concatenata con quella visiva. Il vettore è quindi usato come input da un classificatore (*softmax*) che sceglie la risposta considerando tutto il vocabolario. L'ultimo passaggio puó esser visto come un modello di regressione logistica multi-classe. Per adattarlo al nostro scopo, abbiamo modificato il modello originario in due modi. Abbiamo convertito lo scenario in una singola immagine, concatenando i vettori delle sedici immagini ottenendo un vettore di 6400 dimensioni. Inoltre, la risposta deve essere scelta tra cinque casi (i cinque quantificatori), per cui il vocabolario in output consiste di cinque nodi.

Rete neurale dei quantificatori (RNQ) Questo modello sfrutta i vantaggi delle reti neurali e riesce ad imaparare anche dai propri errori tramite *backpropagation*. Inoltre, disponendo di celle in cui archiviare le rappresentazioni vettoriali delle singole entità, riesce ad ottenere un'astrazione dello scenario alla quale contribuiscono tutte gli oggetti dello scenario. I passi della rete nurale sono i seguenti: (1) i vettori visuali e linguistici sono transformati in uno spazio di trecento dimensioni ($V1$). (2) I vettori visuali delle sedici immagini sono archiviati nelle celle della "memoria": per ogni cella calcoliamo la somiglianza tra ciascun vettore visuale e il vettore linguistico rappresentante il nome della domanda (e.s. *cane*); a tal scopo utilizziamo la norma euclidea. Il risultato è un vettore "di somiglianza" di sedici dimensioni ($S1$). (3) Calcoliamo quindi i vettori pesati di ogni entità moltiplicando le celle della memoria $V1$ con i valori corrispondenti di somiglianza in $S1$. Questo ci dà la rappresentazione di quanta "caninità" ci sia in ognuno degli oggetti. (4) `Sunto 1` è calcolato sommando le celle della memoria con i vettori pesati e rappresenta quanta "caninità" ci sia in un certo scenario. (5) Calcoliamo il prodotto scalare tra i vettori pesati ($W1$) e il vettore linguistico della proprietà (e.s. *nero*), e normalizziamo i valori con la norma euclidea. Il risultato è un vettore di somiglianza di sedici dimensioni ($S2$). Un secondo vettore pesato $W2$ è ottenuto moltiplicanto $W1$ e $S2$. Questo ci da la quantità di "caninità nera" in ogni oggetto. (6) `Sunto 2` è ottenuto sommando i nuovi vettori pesati che sono archiviati nelle celle della memoria e rappresenta quanta "caninità nera" ci sia in un certo scenario. (7) `Sunto 1` e `Sunto 2` sono concatenati in un singolo vettore di seicento dimensioni che viene poi trasformato linearmente in un vettore di cinque dimensioni. (8) Applichiamo quindi un classificatore (*softmax*) che prende come input il vettore di cinque dimensioni e restituisce per ogni quantificatore la probabilità che esso sia la risposta giusta. La rete neurale impara la proporzione tra il nome e la proprietà che caratterizza la relazione espressa dal quantificatore.

4 Risultati

Il modello a rete neurale `RNQ` ottiene un'accuratezza del 38.9% contro il 30.8% del modello `iBOWIMG`. Il modello `RNQ` va molto bene con *no* e *all* e va abbastanza bene con *few* e *most*. Ma la sua accuratezza diminuisce per *some* (18.16%), incidendo negativamente sul risultato complessivo. Una diversa analisi emerge invece dai risultati dell'altro modello, che va abbastanza bene nel predire *no* e *all*, ma va molto male negli altri casi. Per esso, *few*, *most* e *some* sono ugualmente difficili da imparare. Sono da rimarcare i suoi errori in particolare nell'apprendere *few* e *most* (si veda la Tabella 1). Questi quantificatori richiedono una comprensione più precisa delle proporzioni tra nome e proprietà di quanta non ne richiedano *no* e *all*. Pensiamo, quindi, che ciò dimostri i limiti di un modello che non impara ad astrarre ed elaborare le proporzioni. Il modello `RNQ` compie errori comprensibili, confondendo ad esempio *few* con *no* ma non con *some*, *most* e *all*. Ciò dimostra che questo modello ha imparato a quantificare e non semplicemente a tener traccia delle correlazioni nel dataset. Al contrario, l'altro modello confonde *few* con *no* e con *all* in numero uguale.

In generale, il modello `RNQ` risulta essere piut-

RNQ					
	some	all	no	few	most
some	73	88	57	89	95
all	29	**211**	20	19	125
no	32	28	**240**	70	32
few	46	53	104	**129**	68
most	49	148	31	38	126
iBOWIMG					
	some	all	no	few	most
some	89	77	50	108	78
all	45	**163**	63	46	87
no	30	69	**199**	59	52
few	82	81	100	85	52
most	75	110	63	64	80

Tabella 1: Matrice di confusione.

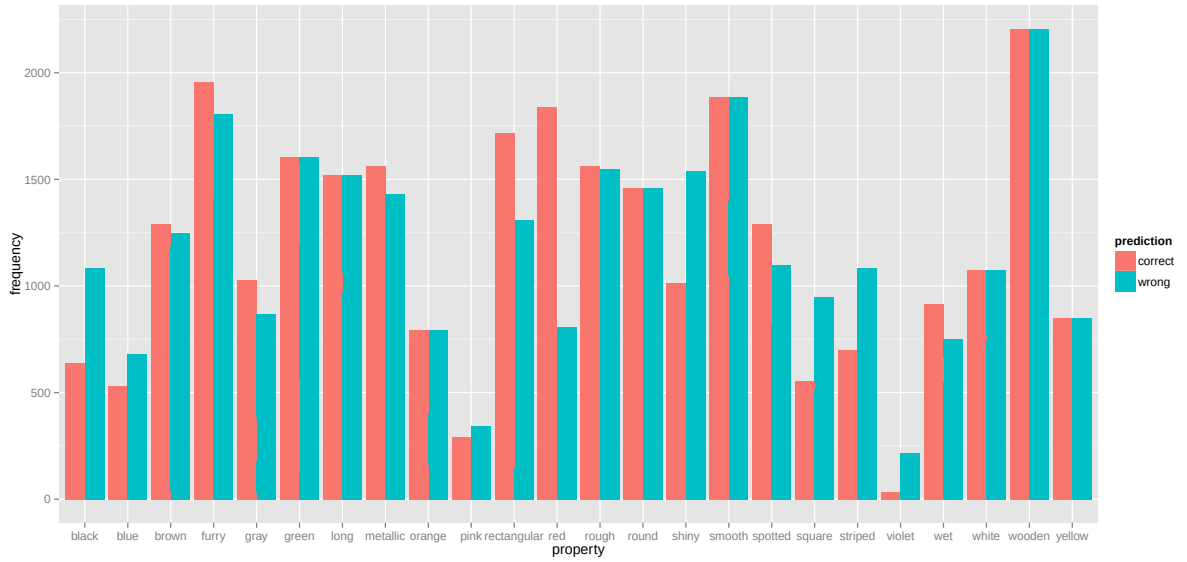

Figura 1: Risposte corrette e risposte sbagliate del modello in relazione alla frequenza della combinazione nome-proprietà.

tosto immune a possibili effetti collegati alla frequenza con cui ciascuna delle 24 proprietà occorre nel dataset. Si potrebbe pensare che combinazioni molto frequenti vengano imparate meglio di combinazioni poco frequenti, ma questo si rivela essere vero solo per pochi casi, su cui spiccano *red* e *rectangular*. Nella maggior parte dei casi, il modello si comporta in modo simile, in termini di riposte giuste/sbagliate, di fronte a combinazioni poco o molto frequenti. Nel caso di alcuni colori (*black*, *blue* e *violet*), notiamo un'accuratezza più alta per combinazioni poco frequenti rispetto a casi molto frequenti (vedi Figura 1). Possiamo dire che la performance del modello RNQ non dipende da effetti di frequenza delle combinazioni presenti nel dataset. Di nuovo, quindi, non si tratta di imparare correlazioni presenti nel dataset, ma per svolgere il compito di quantificazione è necessaria una comprensione più profonda dello scenario visivo e della domanda.

Ringraziamenti

Il secondo autore ringrazia l'Erasmus Mundus European Masters Program in Language and Communication Technologies (EM LCT). Gli altri autori sono stati finanziati dal progetto COMPOSES (ERC 2011 Starting Independent Research Grant n. 283554). Ringraziamo NVIDIA Corporation per la donazione delle GPU usati per questa ricerca.

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual question answering. In *International Conference on Computer Vision (ICCV)*.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *ACL (1)*, pages 238–247.

Sangeet Khemlani, Sarah-Jane Leslie, and Sam Glucksberg. 2009. Generics, prevalence, and default inferences. In *Proceedings of the 31st annual conference of the Cognitive Science Society*, pages 443–448. Cognitive Science Society Austin, TX.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Microsoft COCO: Common Objects in Context*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

M. Ren, R. Kiros, and R. Zemel. 2015. Image question answering: A visual semantic em- bedding model and a new dataset. In *In International Conference on Machine Learningt Deep Learning Workshop*.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al.

2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Andrea Vedaldi and Karel Lenc. 2015. *MatConvNet – Convolutional Neural Networks for MATLAB*. Proceeding of the ACM Int. Conf. on Multimedia.

B. Zhou, Y. Tian, S. Suhkbaatar, A. Szlam, and R. Fergus. 2015. Simple baseline for visual question answering. Technical report, arXiv:1512.02167, 2015.