[ML4] Aspect-based & opinion mining: estrazione di caratteristiche e opinioni dalle recensioni online con tecniche di ML

di Leonardo Albanese

Introduzione

Lo scopo del progetto è quello di testare l'approccio descritto nel paper "*Fine-grained Opinion mining with Recurrent Neural Networks and Word Embeddings*" (Pengfei Liu, Shafiq Joty and Helen Meng) utilizzato nella challenge "*SemEval-2014 Task 4: Aspect Based Sentiment analysis*" (2014). Utilizzando i modelli già addestrati dagli autori del paper (su domini relativi a laptop e ristoranti), si è proposto il testing su domini differenti per valutare la capacità dell'algoritmo, in termini di precision e recall, di essere adattabile a diversi scenari di applicazione.
L'algoritmo utilizza un approccio basato su Recurrent Neural Networks (RNN) e l'utilizzo di word embeddings provenienti da fonti esterne (Google, Amazon e Senna) per l'estrazione di aspetti da reviews online.

Testing dell'applicazione

Per testare l'applicazione si è deciso di utilizzare un container Docker su un'immagine che contiene le risorse software necessarie all'esecuzione dell'applicazione, ovvero Maven, Python, GCC e le librerie richieste.
Per eseguire il codice è stato utilizzato un singolo pc portatile, questo ha portato a non poter eseguire completamente l'algoritmo poiché richiede alte capacità computazionali per essere eseguito, in particolare gli strumenti che fanno parte del back-end delle librerie utilizzate (come Theano).
Il test è stato eseguito sui dataset offerti dalla sfida in formato .xml riguardo laptop e ristoranti.

Formato xml:

```
<sentences>
    <sentence id='index'>
        <text>...</text>
        <aspectTerms>...</aspectTerms>
    </sentence>
```

…
</sentences>

Testing su domini differenti

I test sono eseguiti in domini diversi da quelli proposti dal paper. Nel mio caso ho preso due dataset di reviews da Amazon e Yelp: il primo comprende un set di oltre 300k recensioni relativi a telefoni cellulari e accessori, il secondo comprende circa 500k recensioni di locali di vario genere.

Da entrambi i dataset sono state estratte 10.000 recensioni in maniera casuale e parsate in un xml compatibile con la struttura descritta dalla sfida. Inizialmente si è pensato di suddividere a loro volta questi subset dei dataset originali in training:test con un rapporto di 80:20, tuttavia nei file di training offerti dalla sfida è presente un ulteriore campo xml figlio del campo <sentence> che comprende degli aspectTerms, ovvero dei termini che sintetizzano il significato e l'opinione del revisore. Nell'esempio: "*the hard disk is very noisy"*, hard disk è un aspect term e l'opinione è negativa, data dall'espressione *very noisy*.

Per questo motivo si è deciso di utilizzare come modelli di training gli stessi utilizzati dagli autori del codice, provvisti di aspect term annotati a mano da uno studente ed un esperto linguista. La possibilità di utilizzare questi modelli di addestramento è data dalla relativa vicinanza semantica che accomuna i termini utilizzati frequentemente nelle review di laptop e cellulari e di ristoranti e locali.

Valutazioni

L'algoritmo (eseguito fino a dove possibile, sugli embeddings di Senna e con RNN Elman type) restituisce alcuni aspect terms all'interno delle reviews di cellulari e locali, poiché tende a classificare come tali soltanto aspetti semanticamente molto simili ai domini dei dati di training.

```
t O O
needed O O
a O O
UNKNOWN O O
of O O
bluetooth O B-TERM
headphones O O
for O O
running O O
and O O
since O O
the O O
```

```
the O O
UNKNOWN O B-TERM
battery O I-TERM
life O I-TERM
and O O
bluetooth O B-TERM
UNKNOWN O I-TERM
of O O
the O O
UNKNOWN O B-TERM
hd O I-TERM
```

```
you O O
cannot O O
carry O O
your O O
phone O I-TERM
in O O
your O O
front O I-TERM
pocket O I-TERM
while O O
```

Come si può osservare, gli aspetti trovati (bluetooth, battery life, hd) sono termini che si trovano a cavallo tra i due domini, tuttavia alcune classificazioni relative al dominio dei

cellulari, come "phone", nel terzo esempio, sono stati individuati correttamente. Tuttavia ci sono casi (frequenti) di totalmente errata o totalmente mancante classificazione di aspetti (esempio successivo).

```
option O B-TERM
either O B-TERM
but O B-TERM
expected O B-TERM    i O O
for O B-TERM         am O O
behind O B-TERM      very O O
the O B-TERM         satisfied O O
UNKNOWN O B-TERM     with O O
headphones O B-TERM  this O O
signal O B-TERM      product O O
can O B-TERM         now O O
```

Per quanto riguarda le polarità degli aspetti e delle categorie, non è stato possibile estrarle.

Risultati

Per il calcolo di precision e recall per valutare l'estrazione degli aspetti sono state usate le formule definite dalla sfida SemEval. Ovvero, per ogni sentence:

$$P = \frac{|S \cap G|}{|S|}, R = \frac{|S \cap G|}{|G|}$$

Dove S rappresenta l'insieme degli aspetti estratti e G l'insieme gold di aspetti corretti.

Sono state estratte casualmente 5 reviews per il dominio dei cellulari e 5 per il dominio dei locali (sono state considerate solo reviews che avessero almeno un elemento taggato come aspetto), confrontate con gli aspect terms annotati a mano (gold set).

Di seguito i risultati per gli embeddings di Senna per differenti dimensioni dell'hidden layer della RNN Elman type:

CELLULARI E ACCESSORI	50	100	150	200
Precision	0,796	0,882	0,806	0,932
Recall	0,394	0,703	0,473	0,582

LOCALI E RISTORANTI	50	100	150	200
Precision	0,950	0,900	0,651	0,807

Recall	0,640	0,698	0,713	0,890

Conclusioni

Non avendo potuto terminare la computazione, il modello non è stato addestrato correttamente, infatti compie numerose scelte sbagliate, selezionando come aspect terms anche termini che non sono dei sostantivi (anche il POS-tagging risulta assente).

Nel dominio dei cellulari e degli accessori il riconoscimento degli aspect terms riesce a funzionare, in particolare rispetto ai primi che presentano un elenco di sostantivi utilizzati nelle reviews comuni ai laptop. Tuttavia un'alta percentuale di termini risulta taggata come 'UNKNOWN' ovvero non presente più di una volta all'interno del training set e ciò rende ancora più difficile la classificazione.

Nel dominio dei locali si è cercato di verificare la classificazione su locali differenti dai ristoranti, sui quali il tagging funziona con un'elevata precision e recall. Il sistema funziona adeguatamente su locali che trattano di materia gastronomica (quindi bar, locali notturni che servono cibo, etc) ma riesce a classificare anche alcune review relative ad hotel e bed and breakfast (per esempio riesce a classificare termini come 'room', anche se non strettamente legati al dominio dei ristoranti).

Il codice, presente su GitHub[1], fornisce un Dockerfile che permette di creare un ambiente con tutte le risorse necessarie per poterlo anche eseguire in cloud e permettere così di effettuare approfondimenti, anche su domini differenti.

[1] https://github.com/Leonardo610/progettoMLSII