Information Retrieval and Extraction - Term Project 2

Team 2：D05922018 程子玲 R05922038 黃郁庭 F03944049 原博文

Agree to share your report with your classmates? (YES)

1.Similarity method

1.1Methodology

首先在前處理的部分，我們先將斷詞後的小說中出現的人名(training set與test set中出現的人物名)做處理以確保名字不會被斷開，例如：'賈夫人'會被斷成'賈'與'夫人'兩個詞，因此需要先將其合併成'賈夫人'以提升後續處理的效果。接著在移除停用詞後，我們將此處理過的小說進行word embedding訓練出可以代表每個詞的vector，我們使用的model為gensim的word2vec。

我們將訓練出的word embedding在training set中分別針對每個relation去計算向量差，例如：在training set中'夫妻'關係中有'薛蟠'與'香菱'、'趙姨娘'與'賈政'、...、'文花'與'賈珍'共13組，我們在'夫妻'關係會計算這13組個別的向量差，即vec('薛蟠')-vec('香菱')、vec('趙姨娘')-vec('賈政')、...、vec('文花')-vec('賈珍')共13個向量差，同理在另外其他11個關係也用同樣做法保存個別relation的向量差。

接著在test階段，我們會對test set中每一對人物計算向量差，我們對其所預測之relation即為與此向量差之cosine similarity最大之training set中的向量差(即前一段所述)所屬之relation。

我們針對'居處'關係採取特別的判別方法：先將小說中詞性標為Nc (標出location的詞)的詞建為一個列表，若test data有一個entity出現在此列表中，則此test data會被預測為'居處'關係。

1.2Experiments

在訓練word embedding的部分，我們嘗試調整了幾種不同的參數：word vector的維度、iteration、window size、min_count、skip-gram或cbow。我們發現使用skip-gram或cbow的效果差不多，而其他設定在 word vector 的維度 = 100、iteration=300、window size=5、min_count=0的效果最好，所以下列皆是根據此設定所做的實驗與討論。

在移除停用詞的部分，我們嘗試了幾種不同的方法：直接使用網路找到的停用詞列表去移除小說中的詞(保留中文數字一~十)、移除小說中被標為比較沒有意義的詞性(Nh,Ps,DE,Pd,Pe,Dfa,Po,Dfb,Dl,Dj,Daa,Dh,Di,Dg,T4,T8,T3,T5,T,SHI,Dk,I)、將被標為上列比較沒有意義的詞性的詞建成一個停用詞列表去移除小說中的詞。這三種方法的accuracy依序為：39.29%、33.93%、31.25%。

基於前一段直接使用網路找到的停用詞列表去移除小說中的詞的方法，我們若在計算training data之向量差時也對所有12個relation均同時考慮反向向量差，則accuracy會從39.29%下降至32.14%，如果只對'母子'關係之training data考慮反向向量差，則accuracy會上升至41.96%。

基於上述直接使用網路找到的停用詞列表去移除小說中的詞的方法，我們另外發現若只將test set中的112 datas只分10類(relation：'母女','父子','父女','兄弟姊妹','夫妻','姑叔舅姨甥侄','遠親','主僕','師徒','居處'，不考慮前一段所述之反向向量差)，accuracy可以從39.29%進步至42.86%；目前試到最好是只分7類(relation：'父子','夫妻','姑叔舅姨甥侄','遠親','主僕','師徒','居處'，不考慮前一段所述之反向向量差)，accuracy可以進步至50.89%，若此時再考慮'父子'關係之training data考慮反向向量差，則accuracy會上升至51.79%。

基於上述直接使用網路找到的停用詞列表去移除小說中的詞的方法，若將training set 中找出的關係向量normalize後(將同一關係之所有向量差加總取平均)，accuracy可以從39.29%進步至42.86%。使用此normalize後的方法，若再只將test set中的112 datas減少至只分成7類(relation：'父女','夫妻','姑叔舅姨甥侄','遠親','主僕','師徒','居處'，不考慮反向向量差)，accuracy可以再進步至49.11%。

1.3Discussions

我們嘗試依據詞性篩選停用詞，但效果卻不比網路找到的停用詞列表好，很有可能是因為POS tagging不夠精確或是斷詞不夠乾淨所致，若是能嘗試其他POS tagging或斷詞方法，或是多嘗試不同詞性組合篩選停用詞，也許效果會更好。

有些關係可能因為方向性較明顯，例如：母子，所以在做向量相似度比對時，若能考慮到反向關係則能提升accuracy；然而有些關係可能因為方向性較不明顯，例如：姑叔舅姨甥侄、遠親等，若將這些關係的反向關係考慮進來則反而會使accuracy下降。

有些relation用這種相似度比對的方法效果不好，所以完全不考慮這些relation反而能增加performance，以'母女'關係為例，可能因為training data很少，再加上'母女'關係的有些entity在文章出現次數過少，而導致word embedding訓練效果不好。'遠親'關係的一些entity雖然在文章出現次數也極少，但可能因為在小說中context的線索較強，所以能訓練出較好的word embedding，因此比較不影響performance。

經過實驗可以發現透過normalized後的training data 的向量差預測test data能提升accuracy，很有可能是因為有些在文章中較少出現的詞計算得到的向量差會有相對較大的bias，若藉由normalize而降低誤差的影響則有機會提升performance。

2. Classifier method

2.1 Methodology

Baseline model

我們一開始從baseline的model做起，baseline的model主要是透過gensium word2Vec的package，斷詞部分參考助教提供使用中研院近代漢語標記語料庫的斷詞檔案，除去詞性的標籤之後，將紅樓夢中整份的文檔下去train word2Vec的model。

之後先看training data，將資料中各組關係依序在文本中找到兩個entities同時出現的句子，並對這些句子中的每一個詞向量做加總，再將一組關係所得到的累加向量，與該關係做fitting，然後進入classifier進行分類。

classifier部分，我們採取了Random Forest, SVM, MLP, ANN四種model進行比較，其中，Random Forest, SVM, MLP這三個model是應用sklearn package 中的方法。
ANN的方法則是使用兩層的neurons(1 個hidden layer)，將training data跟testing data的word Vec 加總放入model中，在放入training data的向量之後，建立synapse weight：synapse_0 += alpha * synapse_0_weight_update，在hidden layer中我們有嘗試放置20~25 個neurons，以及其他的參數調整，在experiment的section中會更加詳細的說明。

但在四個model中，即使在之後我們進行許多後續的處理，random forest的表現在決大部分的時候，表現都還是高於其他三個model。

在testing data的部分也是跟training data處理一樣，將資料中各組關係依序在文本中找到兩個entities同時出現的句子，並對這些句子中的每一個詞向量做加總。
最後再將每對組合的詞向量總和放進classifier進行分類。

做完baseline之後，我們依序做了以下的處理，以提高Accuracy：

1. 將word2Vec 做 Normalize
2. 如果在本來文本中有同時出現entity以及relation的單字，就把該句加權
3. 移除沒有意義的詞性tag
4. 針對Nc的詞性，以及紅樓夢小說人物特性進行處理
5. 將training data中每一個組合納為word2Vec model training的句子中

將word2Vec 做 Normalize

在第一步處理中，我們依舊使用找出每對關係中，兩個entities都有出現的句子，做word2Vec的加總，將每對關係的各句word2Vec加總完成之後，我們針對加總後的

vector(Entity Vector)進行Normalize，Nomalize的方法是取Entity Vector的平均值還有標準差：

$$Entity\ vector = \frac{Entity\ vector - Mean\ of\ Entity\ vector}{Standard\ deviation\ of\ Entity\ vector}$$

做完Normalize之後，accuracy在四個model的Accuracy都有提升，但是Random Forest的表現還是最高。

如果在本來文本中有同時出現entity以及relation的單字，就把該句加權

在觀察文本的特性之後，我們發現文本本身有些許敘述人物關係的句子：

當日寧國公與榮國公是一母同胞弟兄兩個。寧公居長，生了四個兒子。寧公死後，賈代化襲了官，也養了兩個兒子：長名賈敷，至八九歲上便死了，只剩了次子賈敬襲了官，如今一味好道，只愛燒丹煉汞，餘者一概不在心上。

幸而早年留下一子，名喚賈珍，因他父親一心想作神仙，把官倒讓他襲了。他父親又不肯回原籍來，只在都中城外和道士們胡羼。這位珍爺倒生了一個兒子，今年才十六歲，名叫賈蓉。

在這些句子裡，會參雜關係的單詞，而大多不會是training data中完整的關係名詞，所以我們將training data中的關係再分為單個字，希望藉由同時對兩個entities、以及關係單字都出現的句字的wordVec加總進行加權，增加feature的差異性。

但可能由於詞語的變化以及斷詞還是沒有對應的很好，因此在Accuracy的表現上，沒有很明顯的提升。

移除沒有意義的詞性tag

在中研院的漢語標記中，我們可以明顯發現許多詞性代表的詞是沒有意義的，像是：的、者等詞性，但由於詞性太多，所以我們有跑迴圈嘗試不同的詞性疊加的組合，怎樣可以達到比較高的準確度，但是後來發現，其實去除太多詞性tag反而會很顯著的降低Accuracy，因此在後續的處理中，我們只拿掉了一種詞性類別：T4，我們猜測會造成這樣的原因應該是文本的資料量太少。

針對Nc的詞性，以及紅樓夢小說人物特性進行處理

此部分比較技巧一些，我們觀察到詞性與關係上的關聯，如果詞性為Nc，標示的詞為地方，因此套到判斷關係上，如果任一個Entity的詞性為Nc，很明顯的一定是居處。另外，針對遠親的方面我們發現，在文本的架構中，遠親出場的場合多在某一回喪禮的段落中，因此我們特別針對那個段落，抓出裡面的人名（利用詞性為Nb），如果這些人名在整份文本中出現次數小於三次，則列入遠親的候選名單中。

在最後testing data的每組組合檢查，如果任一個entity出現於遠親的候選名單中，就把關係定義成遠親。

將training data中每一個組合納為word2Vec model training的句子中

由於考量到文本與training data數量比較少，所以我們想到能夠把training data中每個關係組合一起加到文本的最後，放進word2Vec的model裡做training，經過一些實驗之後，在我們的結果中發現以以下的方式加入，會是提升Accuracy最高的方式：

[Entity 1], [Entity 2], [Relation], [Entity 1], [Entity 2]

在把training data一起加入文本training word2Vec model之後，Accuracy 蠻大幅度的提升，我們認為這個方法應該還有其他的變形，能夠提升更多Accuracy，只是礙於時間有限，還不及嘗試更多種組合方式。

2.2 Experiments

在第一部分的實驗，各model的參數如下：
Random Forest Model：
n_estimators = 100

SVM Model：

random_state=0

MLP Model：

solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(15,), random_state=1

ANN Model：
hidden neurons=25, alpha=0.01, epochs = 100000, dropout_percent=0.0005

	Random Forest	SVM	MLP	ANN
Baseline model	0.43243243	0.351351	0.342342	0.369369
Word2Vec + Vec Normalize	0.486486	0.441441	0.351351	0.441441
+ relation sentence weighted sum	0.468468	0.450450	0.387387	0.378378
+ 移除沒有意義的tag	0.432432	0.423423	0.297297	0.333333

+ Nc preprocessing 遠親preprocessing	0.522523	0.504505	0.432432	0.414414
+ Extended training set into sentence	0.549550	0.513514	0.540541	0.432432

第二部分的實驗為 ANN Parameters 的參數調整：
此部分單純是希望提升我們自己寫的ANN model的Accuracy，因此是取做完所有處理後，單獨就ANN的model參數進行實驗。總共我們就三個參數進行調整：iteraion(epoch)、一個layer中neuron的數量、Alpha值。

epochs = 10000

	neuron數量 = 15	neuron數量 = 20
alpha = 0.1	0.459459	0.450450
alpha = 0.001	0.414414	0.504505

epochs = 50000

	neuron數量 = 15	neuron數量 = 20
alpha = 0.1	0.468468	0.459459
alpha = 0.001	0.468468	0.477477

2.3 Discussions

就以上的Methodology以及實驗結果我們可以發現，在處理的過程中，將word2Vec 做Normalize、針對Nc的詞性，以及紅樓夢小說人物特性進行處理、將training data中每一個組合納為word2Vec model training的句子中這三種方法是可以有效提高Accuracy的，而如果在本來文本中有同時出現entity以及relation的單字，就把該句加權、移除沒有意義的詞性tag在實驗中雖然會造成Accuracy的下降，但是我們有嘗試如果再最後最佳Accuracy時回頭把這兩者移除，還是會造成Accuracy的下降，因此我們認為應該是各項處理會互相影響彼此的最終結果。
在Classifier的部分，由於希望不要加入太多根據文本設置的特定規則，透過一些基本的方法提升準確度，否則應該還有許多類似篩選Nc詞性這種方式可以使用提高準確度。在我們的實驗中，Random Froest加上所有我們的處理程序會達到最高的Accuracy：0.549550。