The 2019 Conference on Computational Linguistics and Speech Processing
ROCLING 2019, pp. 358-367

基於 Seq2Seq 模型的中文文法錯誤診斷系統

A Chinese Grammatical Error Diagnosis System Based on Seq2Seq Model

王鈞威　Jun-Wei Wang, 簡聖倫 Sheng-Lun Chien
陳義昆 Yi-Kun Chen, *吳世弘 Shih-Hung Wu
朝陽科技大學資訊工程系
Department of Computer Science and Information Engineering
Chaoyang University of Technology
s10427098@gm.cyut.edu.tw
s10727614@cyut.edu.tw
kun26712930@gmail.com
*shwu@cyut.edu.tw (contact author)

摘要

本文將以中文句子的錯誤診斷為實例，說明如何利用深度學習演算法序列對序列(Seq2Seq)模型[1]，使用其中的編碼器與解碼器架構，實作出能夠從學習者的句子當中生成出修改過後的句子，並且識別錯誤的類型。一個句子是由許多詞所組成，我們透過修正前與修正後的兩個句子配成一對讓演算法進行學習，盡可能的使模型識別原始與正確之間的關係，並將有錯誤或是不通順的句子加以修正與改正。此研究利用 Pytorch 所提供的範例更改為我們所想要的功能，以此理論作為基礎的中文文法錯誤診斷系統；此研究分為兩部分：首先利用 NLP-TEA2 至 NLP-TEA5 的 Shared Task 所提供的資料訓練模型。其次因應資料集數量不夠讓機器充分學習，所以我們利用 Ge 等人[2]所提出的方式來擴大訓練的資料集。過去 Chen [3]在 NLP-TEA3 的 Shared Task 使用條件隨機域[4](Conditional Random Field，CRF)得到當時最佳的準確度與精確度。所以我們主要針對 NLP-TEA3 當時所完成的任務結果來做比較，另外為了確保我們所使用的序列對序列的可行性與公平性，在此我們重新訓練 CRF 不做任何的調整與現在的序列對序列一樣做比較。

關鍵詞：文法錯誤診斷系統，深度學習，序列對序列模型，條件隨機域

一、緒論

根據網路調查中文是全世界第二多人學習的語言，現在使用中文的人口已經超過十三億，有越來越多的外國人學習中文。可是中文包含很多漢字符號五音加上沒有固定的文法規則等等因素，所以它被認為是全世界最難學習的語言之一，導致外國人要學習中文是非常困難的。

為開發新的文法錯誤診斷系統，需要使用大量的語料庫讓深度學習演算法調整出合適的模型，透過使用 NLP-TEA2 至 NLP-TEA5 Shared Task 所提供的所有訓練集，將訓練集裡面的原始的句子與修正後的句子倆倆進行配對便會產生 22,656 個句對，利用這些句對建立一個新的文法錯誤診斷系統與之前的 CRF 診斷系統進行比較，但這樣並不能讓模型充分的發揮出它的效果，而這邊我們會再使用 Ge 等人[2]提出的擴大訓練集的方式，因為一個句子裡面存在著不只一種的錯誤，然而模型也未必能夠一次性的完全修改正確，當然模型可有可能將原本正確的句子修改成錯誤，因此利用這幾種特性使得尚未修改完全的句子與修改錯誤的句子都成為了不同的錯誤，就能讓模型以多對一的方式訓練讓模型遇到不同種狀況的時候，能夠靈活的辨識出句子的錯誤，而繁體中文的 TOCFL 的資料集與簡體中文 HSK 的，是根據官方蒐集外國人寫作中文的句子，進行分析得出來的經常性錯誤，大致可以分為四類：冗字(Redundant word，簡稱 R)、缺字(Missing word，簡稱 M)、用字不當(word Selection error，簡稱 S)與詞序錯誤(Word ordering error，簡稱 W)。

二、方法

(一) 條件隨機域(Conditional Random Fields, CRF)

CRF 是條件機率分布模型P($Y|X$)，給定一個 X 序列的標籤，CRF 可利用這標籤經由訓練產出另一組序列輸出 Y，而 Y 是因任務的不同而也會有不同的標籤集合，由圖一所示，若是給定一個資料 X 則會計算 Y 所持有的標籤集合裡所有機率，最後返回一個機率值最大的標籤 Y 作為輸出，而根據模板的設定 X 也可以配合前幾個 X 所輸入的資料或是其他內外部特徵，都會是有助於模型是別出更為精確的輸出 Y。

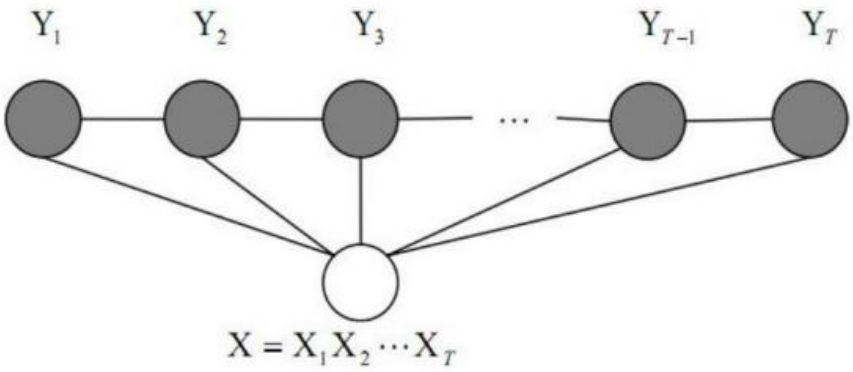

圖 一、條件隨機域

這系統 X 所代表輸入的句子與詞性，而 Y 就是跟 X 相對應的錯誤類型標籤，我們把標籤集定義為：{ O, R, M, S, W }，分別是以下幾種類型：沒有錯誤、冗詞、缺字、用字不當與詞序錯誤。若是將輸入 X 的序列放入到 CRF，此時 CRF 就會利用事先建立好的模板抓取所需要的特徵，而輸入的序列不只是單單放入詞這一種特徵加入 POS 等等的特徵對於 CRF 可以有更好的效果，而在測試時將輸入 X 的序列給 CRF 這時會產生多組可能的標籤組合而每一組組合都會產生符合 X 的機率，最後將機率最大的那一組作為 X 序列的輸出。

(二) 序列對序列(Sequence to Sequence, Seq2Seq)模型

在本次的中文文法錯誤診斷系統，本團隊使用的技術核心為 Seq2Seq 加入 Bahdanau[11]等人所提出來 Attention 專注機制以及雙向 GRU 架構的模型，而在 Seq2Seq 裡面 Encoder 就是負責將輸入序列消化、吸收成一個向量，我們通常把這個向量稱為 context vector，顧名思義，這個向量會把原序列的重要訊息包含起來送至 Decoder 當中，而 Decoder 則是根據 context vector 來生成文字，如圖二所示。

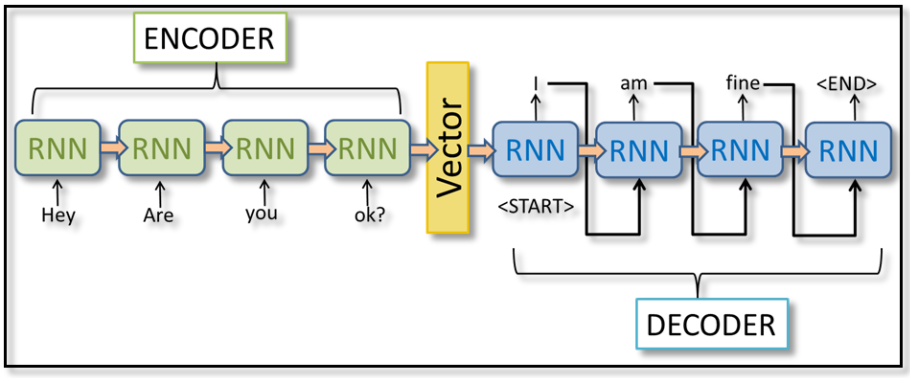

圖 二、Encoder 與 Decoder 示意圖

1. 編碼器(Encoder)

在此模型的編碼器，最主要的工作在於將每一個接收到的文字轉換成一個文字向量與隱藏狀態，並且將每一次的文字向量以及隱藏狀態存取起來，最後將整個句子的向量

及隱藏狀態串起並傳向給解碼器，解碼器將會使用這些向量和隱藏狀態來生成有對於先前的輸入有意義的文字輸出。

在此模型中的編碼器之核心是由 Cho 等人在 2014 年所發明的 multi-layered Gated Recurrent Unit—GRU[7]，此模型使用的是雙向變通的 GRU，這意味著此模型基本上有兩個獨立的 RNN：(1)一個正常方向的序列輸入(2)一個反向的序列輸入 在這兩個 RNN 的輸出端都會計算每個時間的向量和隱藏狀態以便抓取最佳解，使用雙向變通 GRU 將會使模型在編碼過去和未來上下文時有明顯的優勢。雙向變通 GRU 如圖三所示。

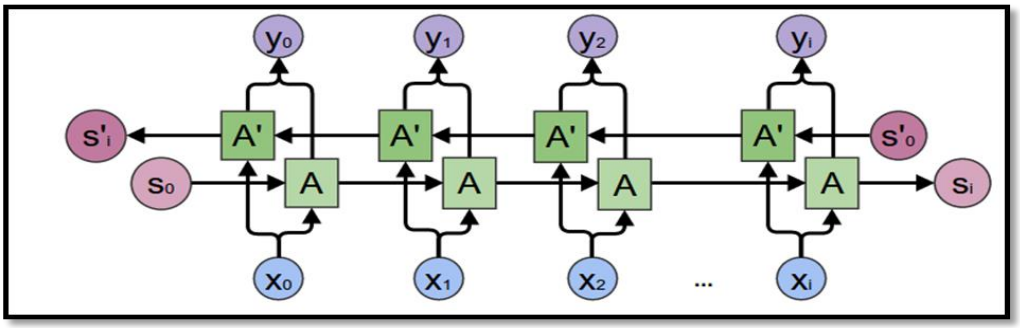

圖 三、雙向變通 GRU 示意圖

2 解碼器(Decoder)

在此模型的解碼器部分，解碼器會使用編碼器傳過來的文字向量以及隱藏狀態來做文字生成的工作，解碼器會依照收到的向量和隱藏狀態去計算並且逐字的生成單字直到解碼器最後生成 EOS_taken 表示句子的結尾，此時解碼器便會停止成文字，但是在一般 Sequence to Sequence 解碼器當中的一個常見問題是，如果我們只依賴於上下文的向量來編碼整個輸入文字的含義，那麼我們很可能會遺漏訊息或是完全讀取錯誤訊息，尤其在處理一段長的輸入序列時更是如此，這極大地限制了解碼器的能力。

為了解決這個問題，Bahdanau[6]等人，創建了一種“Attention 機制”，允許解碼器關注於輸入文字的某些部分，而不是整個輸入的句子，進而提升解碼器再生成文字時能更加的正確率，而在進行錯誤句子與正確句子的分析時，Attention 會將兩者的句子互相差異的地方視為一個需要修改的重點，讓這些有差異的錯誤再生成時不會再出現。在 PyTorch 所提供的 Chatbot 中將 Attention 機制加入此模型中，提高了解碼器的效能。

(三) 內部擴增訓練資料

訓練資料過於少，這是一個在實作深度學習時經常會遇到的棘手問題，特別在處理自然語言處理上的深度學習，更會因為訓練資料量非常的小而使機器訓練的效果不太理想，在經過許多實驗後，決定採用生成的方法擴大訓練資料，Ge 等人提出將第一次 Seq2Seq 所產生的錯誤句子當作訓練資料，然後重新訓練機器並且持續循環到產生出最好的正確句子。利用此方法來擴大機器自己的訓練資料進而提升機器的文字生成正確率，由此方法解決訓練資料過少的問題，本團隊將此概念運用在中文文法錯誤診斷系統上，

將中文訓練資料擴大。

(四) 編輯距離(Edit Distance)

在此次的語法改錯中，我們基於語法錯誤類型中使用了 Edit_distance[10]，由於官網所提供的答案有四種錯誤類型的形式，而我們現階段只能計算三種錯誤類型，所以此次不考慮其中的一種錯誤是否詞序錯誤(W)，因本身詞序錯誤在我們所獲得的資料集裡並不是很多，也因本身想要找出兩句之間字詞調換的位置是困難的，利用此方法將 Edit distance 中的刪除視為冗字、插入為缺字與替換並非詞序錯誤是將原本字詞替換成正確的字詞，使用套件的方式是將官網所提供的句子跟模型產生的句子做字串比對，透過套件的 opcode 會回傳句子的錯誤類型及位置。

三、實驗與結果

(一) 實驗設定

此次的實驗分為四個部分，實驗一是使用 NLP-TEA2 與 NLP-TEA3 的訓練集，實驗二是利用 Ge 的方法擴大訓練集使原本的訓練集擴大兩倍來增加模型對於不同錯誤的修改。實驗一與實驗二分別是模擬當時的 NLP-TEA2 的 Shared Task 利用有限的資料並完成當時的任務。實驗三將 NLP-TEA2 至 NLP-TEA5 所有的資料集，而實驗四同樣是將實驗二的擴大資料集的方式擴大 NLP-TEA2 至 NLP-TEA5 的所有的資料集並放大兩倍，實驗三與四是利用能夠得到的所有資料都套用到模型裡面，盡可能讓模型多看到一些文字。

表 一、訓練集大小

	NLP-TEA2	NLP-TEA3	NLP-TEA4	NLP-TEA5
Redundant	434	10,010	5,852	208
Missing	622	15,701	7,010	298
Disorder (word ordering)	306	3,071	1,995	87
Selection	849	20,846	11,591	474

而評估方式上主要利用模型所生成出來的句子與原始句子作比對，透過比對將所需要的四個類型(冗詞、缺字、詞序錯誤與用字不當)辨識出來並使用 FPR、Accuracy、precision、recall 與 F1-score 再配合表二混淆矩陣來評估。

- False Positive Rate (FPR) = FP / (FP+TN)

- Accuracy = (TP+TN) / (TP+FP+TN+FN)
- Precision = TP / (TP+FP)
- Recall = TP / (TP+FN)
- F1 = 2*Precision*Recall / (Precision+Recall)

表 二、混淆矩陣

混淆矩陣		系統結果	
		Positive	Negative
模型生成	Positive	TP	FN
	Negative	FP	TN

除了使用上述四項評估標準還會再分為 Detection Level、Identification Level 兩個等級再繼續細分下去，Detection Level 會是一個二分類的問題，看此句子正確或是不正確，Identification Level 檢測是否為該錯誤類別，模型預測出來的必須與原本給定的錯誤類型相同。

(二)實驗一

在使用 Pytorch 所提供的 Chatbot 模型裡我們將隱藏層更改為 500 層與 750 層訓練都訓練 50000 回讓兩個模型來比較看哪個能夠有比較好的效果，而測試集的部分是使用 NLP-TEA3 所提共 TOCFL 與 HSK 兩種的測試資料，並且使用官方所釋出的測試工具以達到一個公平的測試結果。

而我們可以根據表三所示，從 RUN1 至 RUN3 是 Chen 參加 NLP-TEA3 所得到的分數。可以看出在各種的分數上我們還是無法與當時最好的成績來比較，但是雖然成果不好但在 FPR(False Positive Rate)的這個部分還是有不錯的成績，因 FPR 是必須越小越好也就表示模型的誤判程度是比原先來的低的。

表 三、實驗一 NLP-TEA3 TOCFL 測試結果

Runs	FPR	Detection Level				Identification Level			
		Acc.	Pre	Rec	F1	Acc.	Pre	Rec	F1
RUN1	0.347	0.595	0.625	0.541	0.580	0.515	0.460	0.302	0.364
RUN2	0.355	0.595	0.623	0.550	0.584	0.513	0.456	0.306	0.366
RUN3	0.363	0.594	0.620	0.554	0.585	0.508	0.447	0.300	0.359
500H	0.294	0.496	0.523	0.301	0.383	0.373	0.255	0.204	0.227
750H	0.297	0.497	0.523	0.305	0.386	0.375	0.264	0.219	0.239

另外從表四可以看到如果怎模型在 HSK 的效果會原比預測 TOCFL 來的好，是因為簡體字是將很多原本繁體字的簡化而成的，所以模型所需要認識的字進而減少也就導致預測出來的句子會比繁體字來的好，可以從 Detection Level 來看除了 Recall 與 F1 其他兩個都比原本的分數來的高，而 FPR 也比當初的分數來的低這就可以看出如果好好的加以調整這個模型是可以比原本來的更好。

表 四、實驗一 NLP-TEA3 HSK 測試結果

Runs	FPR	Detection Level				Identification Level			
		Acc.	Pre	Rec	F1	Acc.	Pre	Rec	F1
RUN1	0.401	0.614	0.600	0.630	0.615	0.571	0.530	0.437	0.479
RUN2	0.419	0.611	0.595	0.644	0.618	0.566	0.523	0.450	0.484
RUN3	0.401	0.614	0.600	0.630	0.615	0.572	0.530	0.435	0.478
500H	0.250	0.619	0.648	0.484	0.554	0.528	0.455	0.362	0.403
750H	0.258	0.617	0.643	0.487	0.554	0.521	0.444	0.357	0.396

(三)實驗二

將原先實驗一的資料加入後兩屆所提供的訓練資料，讓模型可以識別更多的文字，但後來加入的資料多數為簡體中文，對於 TOCFL 的結果從表五可以看到在效果上實驗二的結果會比實驗一來的差，識別過多的簡體中文反而會對繁體中文造成一定的影響，在進行修正的過程裡模型會將原本 TOCFL 答案修正為簡體中文，而在最後將預測結果轉換為比賽格式會與最初的測試資料進行比對，過程中因識別出簡體中文就會導致錯誤的出現。

而在 HSK 的表現可以由表六得知，在兩個等級的 Recall 有了成長，因加入前兩屆的資料讓模型識別更多的句子與不同的錯誤方式，在面對尚未識別過的句子能夠抓出更多的錯誤。

表 五、實驗二 NLP-TEA3 TOCFL 測試結果

Runs	FPR	Detection Level				Identification Level			
		Acc.	Pre	Rec	F1	Acc.	Pre	Rec	F1
500H-1	0.294	0.496	0.523	0.301	0.383	0.373	0.255	0.204	0.227
750H-1	0.297	0.497	0.523	0.305	0.386	0.375	0.264	0.219	0.239
500H-3	0.300	0.491	0.515	0.297	0.377	0.374	0.260	0.205	0.230
750H-3	0.304	0.494	0.518	0.305	0.384	0.369	0.255	0.211	0.231

表 六、實驗二 NLP-TEA3 HSK 測試結果

Runs	FPR	Detection Level				Identification Level			
		Acc.	Pre	Rec	F1	Acc.	Pre	Rec	F1
500H-1	0.250	0.619	0.648	0.484	0.554	0.528	0.455	0.362	0.403
750H-1	0.258	0.617	0.643	0.487	0.554	0.521	0.444	0.357	0.396
500H-3	0.285	0.609	0.626	0.499	0.555	0.509	0.431	0.365	0.395
750H-3	0.283	0.613	0.630	0.504	0.560	0.510	0.431	0.375	0.401

(四)實驗三

此次實驗是由將重新訓練 CRF 與 Seq2Seq 進行比對，此次的 CRF 與實驗一和二最大的不同是不使用 Chen 所使用的搭配詞僅使用原始的詞與詞性所得到的結果，而 Seq2Seq 將實驗一與二透過預測訓練資料產生不同的句子來擴大因訓練資料，但與實驗一相同在這次實驗中只採用 NLP-TEA2 與 NLP-TEA3 的資料集以此做為限制完成當時的任務。

如表七、八所示，CRF 相較於 Seq2Seq 注重於 Precision 反而忽略 Recall 才會使得得到較好的 FPR，然而 Seq2Seq 傾向於句子全面性的修改，雖然 Precision 下降但 Recall 的升高反而在三等級的 F1 都拿到比 CRF 還要好的成果。

表 七、實驗三 NLP-TEA3 TOCFL 測試結果

Runs	FPR	Detection Level				Identification Level			
		Acc.	Pre	Rec	F1	Acc.	Pre	Rec	F1
CRF	0.042	0.521	0.772	0.129	0.221	0.484	0.610	0.058	0.106
500H	0.308	0.494	0.519	0.307	0.388	0.378	0.271	0.220	0.243
750H	0.319	0.494	0.517	0.318	0.394	0.369	0.263	0.227	0.244

表 八、實驗三 NLP-TEA3 HSK 測試結果

Runs	FPR	Detection Level				Identification Level			
		Acc.	Pre	Rec	F1	Acc.	Pre	Rec	F1
CRF	0.071	0.573	0.818	0.273	0.409	0.530	0.755	0.187	0.300
500H	0.317	0.608	0.615	0.530	0.570	0.510	0.439	0.398	0.416
750H	0.333	0.600	0.603	0.529	0.564	0.500	0.428	0.393	0.410

(五)實驗四

重複實驗三之實驗，將 CRF 與 Seq2Seq 訓練集放大到使用 NLP-TEA2 至 NLP-TEA5 所有的資料集，同樣的 Seq2Seq 使用與實驗三同樣的方式將資料放大一倍。

如表九、十所示，在 Recall 的方面都有所成長相對的會導致誤判的上升，但這樣使得 F1 來得比實驗三都來的好。

表 九、實驗四 NLP-TEA3 TOCFL 測試結果

Runs	FPR	Detection Level				Identification Level			
		Acc.	Pre	Rec	F1	Acc.	Pre	Rec	F1
CRF	0.062	0.528	0.743	0.159	0.262	0.481	0.560	0.070	0.125
500H	0.327	0.492	0.515	0.323	0.397	0.363	0.261	0.243	0.252
750H	0.328	0.494	0.517	0.327	0.400	0.360	0.255	0.239	0.247

表 十、實驗四 NLP-TEA3 HSK 測試結果

Runs	FPR	Detection Level				Identification Level			
		Acc.	Pre	Rec	F1	Acc.	Pre	Rec	F1
CRF	0.096	0.614	0.832	0.380	0.522	0.550	0.775	0.265	0.400
500H	0.335	0.606	0.608	0.544	0.574	0.499	0.423	0.409	0.416
750H	0.350	0.604	0.603	0.555	0.578	0.487	0.409	0.411	0.410

四、結論

從實驗一至實驗四看下來，雖然無法比當時 CRF 所得到的分數來得高，但從這幾次的實驗裡我們看到 Seq2Seq 的可能性，而從實驗一 Chen 的表現我們可以知道以現在的 Seq2Seq 還是很難與當時 CRF 做抗衡在各方面的分數都是低於他們的表現，實驗二是不局限於 2016 年 NLP-TEA2 當時資料集，透過將 NLP-TEA2 至 NLP-TEA5 所有的資料集全部套入到模型裡做訓練，可以得知在增加資料的同時也會讓 Recall 隨之地增加利用這一項原理再加上 Tao Ge 等人的方法進行了實驗三與實驗四，將訓練資料集利用訓練好的模型產生不同的錯誤模式並加入到原先的訓練集增加到兩倍的訓練量，讓平時一對一的模型看到各種不同的錯誤方式，來達成二對一甚至是三對一的方式來使得自己的模型更加的靈活而若是再搭配 PreTrain 的技巧，例如：Word2Vec[8]或是 BERT[9]的等等技巧，將文字轉成向量能夠讓詞與詞有良好的連接。

參考文獻

[1] Ilya Sutskever, Oriol Vinyals, Quoc V. Le, 2014, “Sequence to Sequence Learning with

Neural Networks”, In Advances in neural information process-ing systems, pages 3104–3112.

[2] Tao Ge, Furu Wei, Ming Zhou, 2018, “Fluency Boost Learning and Inference for Neural Grammatical Error Correction”, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 1–11.

[3] Po-Lin Chen, 2017, “Chinese Grammatical Error Diagnosis System”, Retrieved from http://ir.lib.cyut.edu.tw:8080/handle/310901800/34228

[4] Taku Kudo, 2007, “CRF++：Yet Another CRF toolkit”, https://taku910.github.io/crfpp/.

[5] NLP-TEA3 CGED Shared Task, 2016, https://www.aclweb.org/anthology/W16-4906.

[6] Dzmitry Bahdanau, KyungHyun Cho, Yoshua Bengio, 2014, “Neural Machine Translation by Jointly Learning to Align and Translate”, arXiv preprint arXiv:1409.0473.

[7] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua Bengio, 2014, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling”, arXiv preprint arXiv:1412.3555v1.

[8] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2* (NIPS'13), C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.), Vol. 2. Curran Associates Inc., USA, 3111-3119.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, 2018, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding ,(Submitted on 11 Oct 2018 (v1), last revised 24 May 2019 (this version, v2))

[10] Eric Sven Ristad, Peter N. Yianilos, 1998, IEEE,” Learning String-Edit Distance”, IEEE Transctions On Pattern Analysis And Machine Intelligence, Vol. 20, No. 5.