基於 BERT 任務模型之低誤報率中文別字偵測模型

Low False Alarm Rate Chinese Misspelling Detection Model Based on BERT Task Model

沈峻毅　Jyun-Yi Shen
張道行　Tao-Hsing Chang
國立高雄科技大學資訊工程系
Department of Computer Science and Information Engineering
National Kaohsiung University of Science and Technology
1106108125@nkust.edu.tw
changth@nkust.edu.tw

摘要

中文別字偵測技術可以應用在教育及出版等許多實務領域。雖然近期許多研究提出了一些能提高效能的模型，但這些模型卻有著誤報率偏高的缺點。在真實的應用中，減少誤報情況的發生是很重要的，因為使用者在操作時，一直出現誤報的情況，會讓使用者體驗不佳，所以如何生成低誤報率且高效率的模型，成為一個要處理的問題。本文採用 BERT 在 Single Sentence Tagging 任務模型來解決中文別字偵測的問題，並配合此模型設計了訓練資料的大量生成方法。實驗顯示本文所提方法對 SIGHAN 2015 測試資料集的誤報率(False Alarm Rate)達到 0.0297。與先前其他低誤報率方法相比，此方法有最低的誤報率以及最高的召回率。

Abstract

Chinese misspelling detection technology can be applied in fields such as education and publishing. This research topic has garnered considerable attention. Recently, although many studies have proposed models that are based on deep learning and that are capable of improving detection accuracy, these models have the disadvantage of high false alarm rates. In real application scenarios, it is important to reduce the occurrence of false alarms because false alarms, while using the system, lead to poor user experience. Therefore, it is important to create a model with low false alarm rate and high efficiency. In this paper, BERT Single Sentence Tagging task model is used to solve the Chinese misspelling detection problem. To work with this model, mass training data generation methods were designed. Experiments showed that the method employed in this study has a false alarm rate of 0.0297 for the

SIGHAN 2015 test set. Compared to other previous methods with low false alarm rates, this method has the lowest false alarm rate and the highest recall rate.

關鍵字：BERT，中文別字偵測，低誤報率

Keywords: BERT, Chinese misspelling detection, low false alarm rate

一、緒論

錯別字問題是一個持續受到討論的重要議題，特別在語言教育上。一般認知的錯別字應分成兩類，一是字形本身是不存在字的錯字，二則是字本身是存在字但被誤用的別字。但現代人寫文件或報告大多都使用電腦和手機的輸入法，所以本文討論聚焦在別字偵測。目前已經有許多研究與方法被提出，然而，雖然近年來的方法在實驗報告中看起來有不錯的效果，例如 Wang, Song, Li, Han, & Zhang (2018)述及其方法的精確率(precision)、召回率(recall)與 F1 等三項評估指標分別為 0.57、0.70、0.62；FASPell (Hong, Yu, He, Liu, & Liu, 2019) 則提到其三項指標分別為 0.68、0.60、0.64。然而在實際應用中，這些模型仍然遇到很大的挑戰。其中最關鍵的因素是在實際應用時必須將誤報率(False Alarm Rate)儘量壓低。這是因為在大多數情境下別字只是文件中相當少量的發生，若假設某方法將正確字誤判成別字的誤報率達 0.2，那麼即使每 10 句就有一個別字且百分之百被找出，但該方法還是會同時指出 2 個沒有問題的句子有別字，這將造成不大理想的使用者體驗。因此低誤報率是別字自動偵測能否在實務場域被採用的重要因素。

我們認為 Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019) 提出的 BERT for Single Sentence Tagging Task (以下簡稱 BSST)架構相當適合用於別字偵測並克服這個問題。以 BERT 模型為基礎的方法近年在許多自然語言處理(NLP)問題上成為好的解決方案，BSST 是其中之一。BSST 原始用途之一是命名實體識別(Name Entity Recognition, NER)。舉例來說，我們的目標是將句子中的字分成 5 類，分別是名詞詞首、名詞詞幹、形容詞詞首、形容詞詞幹和其他字。假設給 BSST 一個句子「今天天氣真好」，那 BSST 的目標就是對照原句順序依序輸出下列結果:

今	天	天	氣	真	好
名詞詞首	名詞詞幹	名詞詞首	名詞詞幹	其他字	形容詞首

由此模式，我們可以套用在別字偵測。當我們輸入一個句子，BSST 的輸出為是別字以及不是別字兩個類別。如此一來 BSST 便可用來偵測別字。

因此，本文的目的是提出一個以 BSST 模型為基礎的別字偵測方法，並針對 BSST 所需要的大量訓練資料提出一個模擬資料生成方法。以下各節的組織如下: 在第 2 節我們說明別字偵測常見的一些方法及其原理。第 3 節將略述 BERT 與 BSST 的基本架構。由於訓練 BSST 需要龐大的資料量，但別字的語料庫規模多有限，因此第 4 節說明我們產生訓練 BSST 資料的方法，並於第 5 節說明實驗結果。最後對本文提出方法進行討論。

二、文獻回顧

中文別字偵測一直是受到矚目的研究議題。近年來有許多方法被提出，也有許多歸納與分析這些方法的評論性論文 (Wu, Liu, & Lee, 2013; Yu, Lee, Tseng, & Chen, 2014; Tseng, Lee, Chang, & Chen, 2015)。 Chang (1995)提出了早期的中文別字自動偵測的架構，雖然有著誤判率太大、偵測時間長等問題，但也為這領域開啟了研究起點。早期的研究利用混淆字表偵測別字，也就是事先蒐集常見別字，之後為每個別字依照字形、字音、涵義及輸入法等規則建立混淆字表，系統會將句子中的每個字替換為字表中的字，最後再利用模型計算所有修改句的機率。

之後一些研究發現採用字音、字形與語法特徵偵測別字有不錯的效果。Liu, Lai, Tien, Chuang, Wu, & Lee (2011) 除了把相同音的字製作成混淆字集，也使用倉頡輸入法來編碼，目的是可以更快速的去計算字形相似度。Chang, Su, & Chen (2012)基於一個假設: 有別字的詞在斷詞後被切割成多個單字詞。基於這個假設，此方法偵測到句子有連續的單字詞出現，便進一步以各單字與候選正確字的字音、字形相似度以及字詞頻和詞性組合機率，判斷是否有別字。Wang, Liao, Wu, & Chang (2013)先檢查句子裡出是否有出現混淆字集裡的高頻別字，之後使用 CRF 把句子斷詞，最後再使用 tri-gram 模型進行判斷，判斷完後會把少於 3 個字元的詞都當成是別字的可疑字，並依照字形或字音

去替換這些可疑字，再重複輸入進 tri-gram 模型內計算分數，直到找出最高分數的句子。

後續許多研究多基於前面兩個研究的基本架構進行改良或修正。Chang, Chen, & Zheng (2014) 進一步修正了 Chang et al. (2012)的方法，採用筆畫結構取代部件結構評估字形相似度、並加入了常見單字詞別字的規則式方法。Xiong, Zhang, Zhang, Hou, & Cheng (2015)發表的 HANSpeller 採用先前一些方法的基本原理，提出一個兩階段的架構。第一階段此方法會將會句子送入一個簡單的分類器，把過於明顯不是錯別字的選項給篩選掉。第二階段則會把句子翻譯成英文，接著把翻譯後的英文句輸入Microsoft Web n-gram Service，去計算英文翻譯的 n-gram 分數，依照這個分數去篩選剩下的可疑字。類似做法還有 Chu, & Lin (2015)與 Xie, Huang, Zhang, Hong, Huang, Chen, &Huang (2015)提出的方法。Chu, & Lin 是先將句子斷詞，斷詞後將句子內的字或詞用混淆字集做替換，再使用 Google n-gram Viewer 去計算最有可能有別字的選項。Xie et al. (2015)則是採用 Chang et al. (2012) 的假設，先把句子斷詞，再去判斷是否有連續的單字詞，如果有錯字的話，會有很大的機率被斷成連續的單字詞。之後再透過混淆字集與語料庫做修正，最後使用 bi-gram 和 tri-gram 計算並挑選正確的句子。

Wang, Song, Li, Han, &Zhang (2018)也是採用以字音字形為特徵輸入一個分類器預測別字的類似架構，但由於其採用雙向 LSTM 模型(Bi-LSTM)作為預測器，因此需要大量訓練資料。為此，此方法使用光學字符辨識(Optical Character Recognition, OCR) 和自動語音辨識(Automatic Speech Recognition, ASR)在辨識時會將相似字形和字音的字列為候選詞的特性，產生每個字的混淆字集，並透過大量數據集與混淆字集去訓練 Bi-LSTM。其實驗結果的精確率(precision)和召回率(recall)分別達到了 0.54 和0.69。這也顯示需要有足夠貼近真實錯別字的訓練資料是很重要的。

近年來由於深度語意網路的發展，有研究開始採用這類模型辨識別字。例如 FASPell (Hong, Yu, He, Liu, & Liu, 2019) 使用 BERT 的 masked language model，將所有的字都當成可疑字，並逐步把每個可疑字遮蔽，再來預測被遮蔽的字是什麼字。如果預測的結果內沒有原先被遮蔽字的話，那就認定被遮蔽字是別字。SpellGCN (Cheng, Xu, Chen, Jiang, Wang, Chu, & Qi, 2020)也是採用 BERT 的 masked language model，比較不同的是，他們將字音與字形相似度做成graphs，再使用graph convolutional network (GCN)去計算最佳解。Zhang, Huang, Liu, & Li (2020)則將模型分成偵測層與校正層，偵測層使用雙向

GRU 模型 (Bi-GRU)，校正層使用BERT模型；偵測層的輸入是句子的embedding，輸出是代表這個字是否為別字的機率。將這些機率做 soft masking 的計算，計算結果再輸入校正層。校正層會去預測被遮蔽的字，再將預測出來的結果和偵測層輸入的embedding相加，作為每個字的最終特徵。將這些特徵輸入到 softmax 分類器，再由分類器去篩選哪個候選字是最佳解，最後輸出校正句子。

這類模型的優勢在於能使用前後文語意特徵作為判別依據，因為先前方法所採用的別字出現在連續單字詞的假設無法處理有別字的詞仍是一個成詞的問題、也無法處理字音、字形相似度與 n-gram 模型以外的例子，雖然這些模型的 F1值都有不錯的成績，但還是沒有適合的方式去解決誤判率過高的問題。

三、BERT for Single Sentence Tagging Task

BSST 主要是由 BERT 模型的輸出加上一層簡單的線性分類器所組成，如圖 1 所示。該模型一次接受一個句子輸入，句首輸入符號 CLS 以便於 BERT 模型分析。由於每個句子長度不一，故須設定一個最大值 n，如果句長小於 n 時，則會使用 zero padding 方式補滿。BERT 模型輸出的結果會進一步輸入給 n 個線性分類器(Linear classifier)，這些分類器會再依據 BERT 輸出的語意特徵作出對每個字做出是否為別字的決定，以圖 1 來說，輸出 0 為正確字，1 為別字。

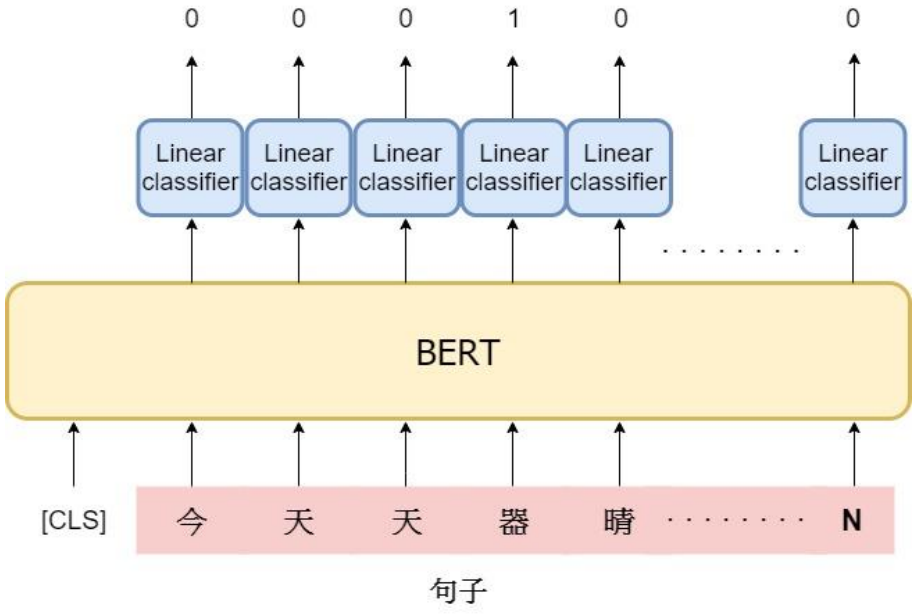

圖 1、BERT for Single Sentence Tagging Task 之架構圖

這個模型的核心 BERT 是由多個 Transformer 的 Encoder (以下簡稱 TE)所組成的，如圖 2 所示，圖中 E1 到 En 為輸入字的 embedding，輸出的 T1 至 Tn 為 E1 至En 在此句語境中的語意向量，n 為輸入句子長度的最大值。輸入與輸出間的中間層使用 12 層的 TE，內部的計算機制則使用 Self-Attention (Vaswani et al., 2017)。該機制有助於整合句子前後詞的語意，對於模型理解整個句子有相當大的幫助。每個字在每一層的 Transformer 要做 12 個 Head 的 Self-Attention，並把 12 個 Head 的 Self- Attention 結果做運算再輸入到下一層 Transformer 中，最後輸出該字的語意向量。

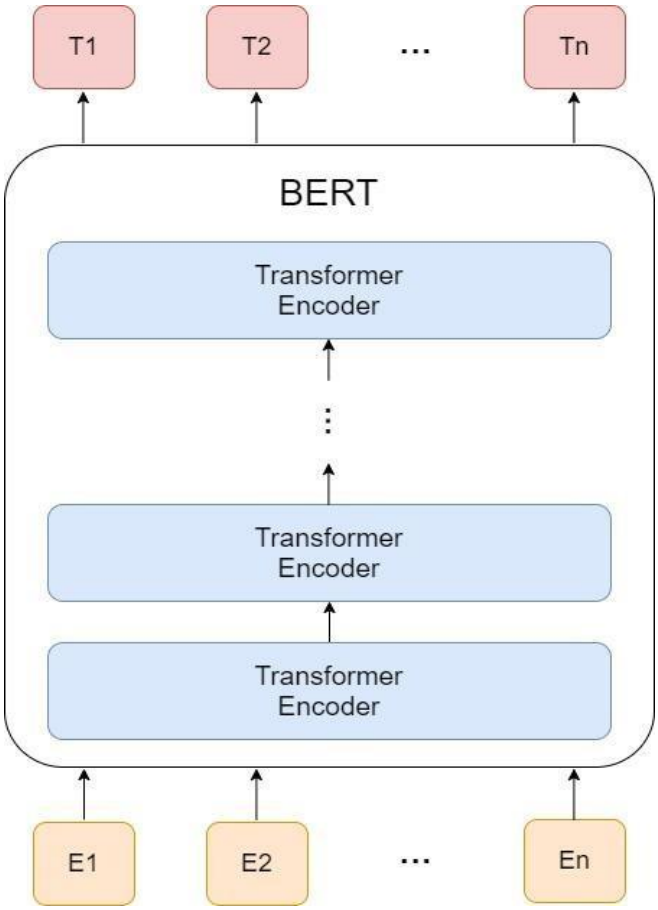

圖 2、BERT 架構圖

四、訓練資料的生成

在第二節中曾提到對於深度學習模型如何產生訓練資料是相當關鍵的問題。訓練資料必須盡可能滿足兩個條件： 資料量大且為真實資料。然而，就目前已知的資料集而言，資料量仍有所不足。此外，這些資料集雖然都是真實資料，但由於資料量不足，對真

實環境中別字發生樣態的涵蓋率不足。而訓練資料需要同時有無別字以及有別字的句子集(以下分別稱為正常集與別字集)，因此除了偏誤語料，也需要同性質的正常語料。根據上述需求，本文提出一個訓練資料生成方法，藉由現有大量語料生成模擬真實資料的訓練語料。

首先，我們使用聯合報 2002~2009 年新聞報導為語料來源(以下稱為源語料)。這個來源不但有足夠的語句，而且當時報導內容幾乎完全沒有別字，所以適合作為當作正常集來源。我們從源語料中隨機抽取 40 萬句組成正常集。

對於別字集則利用下列步驟產生。第一、我們建立每個中文字元的候選字集(以下簡稱選字集)，建立的方法是將所有中文字元兩兩計算在字音與字形相似性，計算方法是使用 Chang et al. (2014)的方法。此方法在字音部分利用兩個字的聲母、介母、韻母和聲調的相同與相異計算出一個相似機率。在字形部分利用筆畫結構以最長共同子串列(LCS)為基礎的方法計算相似度。計算出所有的字音和字形分數後，我們以一個加權線性方程式算出相似值，最後針對每個字蒐集與該字最相似的前 k 個字形成該字的選字集。第二、從源語料另外隨機抽取與正常集不重複的 40 萬句。第三，對於每個別字集的句子，都根據中研院平衡語料庫統計的字頻表作為機率計算依據挑選句中將被變更的字。但每個字被挑中的次數有限制，若某字的別字句數量已達上限，則不在被挑選的範圍內。這個限制是確保別字集不會被少數高頻字的別字所涵蓋。第四、從被挑選字的選字集中隨機挑選一個字替換原先的字模擬成錯別字。

五、實驗

本文將以 SIGHAN 2015 (Tseng, Lee, Chang, & Chen, 2015)資料集作為評估本文所提方法的測試集。此測試集共有 1100 句，其中正常句與別字句各有 550 句。由於我們的模型暫不處理長度超過 60 字的句子，因此經排除過長句子後共有正常句有 538 句，別字句有 534 句。

實作 BSST 部分我們則是依據 Wolf et al. (2019)的 HuggingFace's Transformers 所設計的 BERT for Token Classification 開源碼。測試各模型與模式的評估指標與 SIGHAN 2015 相同，各指標計算方式及使用符號說明如下。

誤報率(False-Alarm Rate)：FP / (FP+TN)

正確率(Accuracy)：(TP+TN) / (TP+FP+TN+FN)

精確率(Precision)：TP / (TP+FP)

召回率(Recall)：TP / (TP+FN)

F1：2* Precision* Recall/(Precision+ Recall)。

其中

TP：所有被辨識為有別字的別字句數量。

TN：所有被辨識為正常的正常句數量。

FP：沒有別字卻被辨識為有別字的正常句數量。

FN：有別字卻被辨識為正常句的別字句數量。

表 1 比較本文所提方法與其他誤報率(False-Alarm Rate)低於 0.1 的先前方法的效能。先前方法包括 NTOU (Chu, & Lin , 2015)以及 NCTU+NTUT (Wang, & Liao, 2015)的兩個 Run，這些方法的數據引自 SIGHAN 15 對測試集的實驗結果(Tseng, Lee, Chang, & Chen, 2015)，由於高誤報率的方法在應用上的限制，在此不列入比較。由表 1 可知，本文所提方法比先前誤報率最低方法的誤報率還低 42%，但召回率與 F1 比先前方法最佳者還高 12%。這顯示本文所提方法在降低誤報率同時也能提升別字的偵測率，更接近真實應用需求。

表 1、誤報率低於 0.1 之模型的評估比較

Models	**誤報率**	**正確率**	**精確率**	**召回率**	**F1**
本文所提方法	0.0297	0.6446	0.9135	0.3165	0.4701
NTOU	0.0909	0.5445	0.6644	0.1800	0.2833
NCTU+NTUT-Run1	0.0509	0.6055	0.8372	0.2618	0.3989
NCTU+NTUT-Run2	0.0655	0.6091	0.8125	0.2836	0.4205

表 2 說明本文所提方法在不同訓練資料產生模式下造成的效能差異。一個訓練資料只有別字集而沒有正常集(以下簡稱無正常集)；另一個是別字集的句子在選定別字發生位置而挑選替換字時，不是從選字集中挑選，而是採用隨機選擇任一中文字元(以下簡稱無選字集)。由表 2 可知，無正常集模式的誤報率大幅提高。我們認為是因為該模型並未有學習正常句子的經驗，以至於完全沒有錯誤的句子中若有些微語意不一致，

就會被判斷成別字句。而無選字集的模式表現更差，我們認為原因是因為隨機挑選中文字元的方式和真實的情況是有相當大的落差，無選字集模式替換的別字有很高的可能性並不會出現在句子中的那個位置，對模型而言即使有很多的訓練資料也很難擷取語意不一致的正確模式。而選字集有效限縮別字選擇範圍，可以讓模型比較容易捕捉發生語意不一致的模式。

表 2: 訓練資料產生方法差異之效能比較

訓練集產生模式	誤報率	正確率	精確率	召回率	F1
本文所提方法	0.0297	0.6446	0.9135	0.3165	0.4701
無正常集	0.6022	0.4534	0.4564	0.5094	0.4814
無選字集	0.5428	0.3246	0.2589	0.1910	0.2198

六、結論與未來工作

本文提出一個以 BERT 為基礎的別字偵測方法，以及一個產生能訓練此模型的訓練資料的模擬資料生成方法。初步的實驗結果顯示本文所提方法能有效降低誤報率，也能維持整體效能。我們認為能有這樣的效果，除了採用語意面向的辨識方法外， 足夠的訓練資料以及選字集的產生是模擬實際別字發生的過程是重要的因素。

雖然誤報率有效的降低，但若召回率可以再提升，則應用的範圍將可以更擴展。我們認為有幾個值得努力的方向。第一，這次的實驗因為受限於系統的設計，所以暫不處理六十字以上的句子，未來也希望可以將此缺點加以改良，往後也會嘗試各種的測試資料，去更進一步的證明此系統的可行性。第二，本次實驗可以初步證明不同的訓練資料產生方式能有效的降低誤報率，但對於BERT是否有助於降低誤報率，我們還需要更完整及更多的實驗才能證明。第三，別字對句子各面向造成的影響進行更多地探索。近期採用深度學習模型的方法多基於別字造成語意不協調性的假設，不過這些方法仍然注意到字音與字形相似性仍是一個重要的特徵，因此在選字集階段採用不同的方法使用這些特徵。我們認為可以嘗試在決策階段融合不同面向的特徵。第四、我們可以進一步檢視目前無法辨識的別字具有的特性。在本文中我們專注在降低誤報率的設計，我們可以嘗試進一步分析未被正確辨識的別字句，探索本文所提方法未知的侷限，進而找出提升召回率的方法。

七、致謝

本研究部分經費由科技部計畫(MOST 107-2511-H-992-001-MY3)支持。

參考文獻

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Chang, C. H. (1995). A new approach for automatic Chinese spelling correction. In *Proceedings of Natural Language Processing Pacific Rim Symposium* (Vol. 95, pp. 278-283).

Chang, T. H., Chen, H. C., & Zheng, J. L. (2014, October). Using Chinese Orthography Database to Correct Chinese Misspelling Words With Graphemic Similarity. In *Proceedings of the 26th Conference on Computational Linguistics and Speech Processing (ROCLING 2014)* (pp. 153-162).

Chang, T. H., Su, S. Y., & Chen, H. C. (2012, December). Automatic correction for graphemic Chinese misspelled words. In *24th Conference on Computational Linguistics and Speech Processing, ROCLING 2012* (pp. 125-139).

Chu, W. C., & Lin, C. J. (2015, July). NTOU Chinese spelling check system in SIGHAN-8 bake-off. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing* (pp. 137-143).

Cheng, X., Xu, W., Chen, K., Jiang, S., Wang, F., Wang, T., Chu, W., & Qi, Y. (2020, July). SpellGCN: Incorporating Phonological and Visual Similarities into Language Models for Chinese Spelling Check. *In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 871-881).

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*.

Hong, Y., Yu, X., He, N., Liu, N., & Liu, J. (2019, November). FASPell: A Fast, Adaptable, Simple, Powerful Chinese Spell Checker Based On DAE-Decoder Paradigm. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)* (pp. 160-169).

Liu, C. L., Lai, M. H., Tien, K. W., Chuang, Y. H., Wu, S. H., & Lee, C. Y. (2011). Visually and phonologically similar characters in incorrect Chinese words: Analyses,

identification, and applications. *ACM Transactions on Asian Language Information Processing (TALIP)*, *10*(2), 1-39.

Tseng, Y. H., Lee, L. H., Chang, L. P., & Chen, H. H. (2015, July). Introduction to sighan 2015 bake-off for chinese spelling check. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing* (pp. 32-37).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).

Wang, Y. R., & Liao, Y. F. (2015, July). Word vector/conditional random field-based Chinese spelling error detection for SIGHAN-2015 evaluation. In Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing (pp. 46-49).

Wang, D., Song, Y., Li, J., Han, J., & Zhang, H. (2018). A hybrid approach to automatic corpus generation for Chinese spelling check. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 2517-2527).

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistav, P., Rault, T., Louf, L., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., Rush, A. M., & Brew, J. (2019). HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv*, arXiv-1910.

Wu, S. H., Liu, C. L., & Lee, L. H. (2013, October). Chinese spelling check evaluation at SIGHAN Bake-off 2013. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing* (pp. 35-42).

Xie, W., Huang, P., Zhang, X., Hong, K., Huang, Q., Chen, B., & Huang, L. (2015, July). Chinese spelling check system based on n-gram model. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing* (pp. 128-136).

Xiong, J., Zhang, Q., Zhang, S., Hou, J., & Cheng, X. (2015, June). HANSpeller: a unified framework for Chinese spelling correction. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 20, Number 1, June 2015- Special Issue on Chinese as a Foreign Language*.

Yu, L. C., Lee, L. H., Tseng, Y. H., & Chen, H. H. (2014, October). Overview of SIGHAN 2014 bake-off for Chinese spelling check. In *Proceedings of The Third CIPS-SIGHAN*

Joint Conference on Chinese Language Processing (pp. 126-132).

Zhang, S., Huang, H., Liu, J., & Li, H. (2020, July). Spelling Error Correction with Soft-Masked BERT. *In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 882-890).