Tendulkar's Cat and Schrodinger's Bat - Knowledge Enhanced Real Word Error Correction

Shailesh Sridhar, Saahitya E, Satyam Shivam Sundaram, and Bhaskarjyoti Das

Abstract Real word errors are those errors in which a word is lexically correct or syntactically correct in the sentence it belongs to, but is contextually incorrect. As a result, they are difficult to identify and arguably even more difficult to correct. In this work we attempt to create a model based on Total Word Similarity which is able to correct real word errors in sentences based on the simple assumption that a real word error will not be semantically cohesive with respect to its context, and that the actual word which was intended, will be. The main aspects of our approach are the usage of semantic context, knowledge context and syntactic context, all together at once. To this approach we also retain the concept of edit penalty, which is an integral component of most spelling checkers. We modified the Microsoft Research Sentence Completion task of 2011 for measuring the performance of real word error correction models, and achieve 80% accuracy.

1 Introduction

Take the example of the title of this paper. "Tendulkar's Cat and Schrodinger's Bat". A cricket fan who is familiar with physics or vice versa will be able to identify that the words Bat and Cat are probably interchanged. However, a person without the knowledge of Tendulkar or Schrodinger may not be able to do so. It is possible that a

---

Shailesh Sridhar
PES University e-mail: shailesh.sridhar@gmail.com

Saahitya E
PES University e-mail: saahitya.e@gmail.com

Satyam Shivam Sundaram
PES University e-mail: satyspy007@gmail.com

Bhaskarjyoti Das
PES University e-mail: bhaskarjyoti01@gmail.com

person who has simply encountered the terms “Schrodinger’s Cat” and “Tendulkar’s Bat” is able to detect the errors.

The examples cited here are not typical spelling error corrections. A spell-checker used in a word processing software or a mobile phone is not likely to catch these errors. Typical error correction depends on variants of edit distance algorithm and an implementation of a standard lexicon. Such error correction methods primarily focus on spelling errors corrections and only work on correcting errors in words that don’t exist in the standard lexicon. Hence such methods will not spot the real world errors mentioned above as the words are in the standard lexicon but they are contextually wrong. However, correcting real world errors is an important problem and has serious applications in OCR, video subtitle generation, etc.

An important hypothesis regarding real word error correction is that a real word error does not fit in with its surrounding context. In a typical natural language processing pipeline, the goal is to find the meaning of a piece of text and the key steps in meaning disambiguation are resolutions of syntax and semantics. Hence, both syntax and semantics can be good candidates in the word error resolution process. Knowledge is usually not part of the meaning resolution process in natural language processing but can really contribute a lot in resolution of meaning as the usage of words bears lot of relevance to the domain context in which they are used. Hence knowledge context should also be a candidate in the word error correction problem. The work described in this paper uses all three kinds of context while keeping the traditional edit distance approach. Since all three kinds of context can be used in the algorithm, a suitable weight of each should be decided in any such implementation. Hence, it is possible to have candidate words for correcting a real word error by using all three different contexts and the edit distance of each candidate. One can then score each candidate and the candidate with the highest score can determine the correct real world error correction.

Distributional semantics is an alternative to lexical semantics as any lexicon tends to be limited. Distributional semantics comes up with an embedding for each word in a language vocabulary based on usage. The key assumption here is that the embedding of two words will be adjacent in the embedding space if these two words tend to appear in a small window in real usages as observed in a very large corpus of text. Hence,word embedding can be used to have a vector representation of the context in which the candidate word can be used in a sentence.

Knowledge engineering and machine learning so far have continued on two different axis. While knowledge engineering depends on specifics and hierarchy as captured by the “subject, predicate, object” triple, machine learning depends on as much data as possible. Machine learning, in spite of lot of training data, may still deliver doubtful results. Knowledge engineering on the other hand may present a challenge as knowledge graph can keep growing for ever and may never be considered completely done. However, both taken together can provide an interesting approach that harnesses the strength of both. But searching in knowledge graph for relevant context is an approach that is hard to scale. Hence researchers nowadays have come up with approaches that capture the facts of a knowledge graph into a suitable embedding scheme. An embedding approach of knowledge graph also makes it amenable to modern techniques of natural language processing that already uses word embedding. In real word error correction problem as well such an approach makes sense to capture knowledge context. In this work, that knowledge embedding approach has been adopted as well.

2 Literature Survey

2.1 Spelling Error Correction and Real-Word Errors

Shashank Singh et al.[23] discussed error correction and detection methods in text. The authors divide typographic errors into two classes i.e. spelling real word errors. Both these types of errors are not limited to text alone. Henry Liberman et al. [13] explained the frustrating problem of homophone in speech dictation systems. Graeme Hirst et al.[9] refer to real word errors as 'malapropisms'. Their work is based on the basic assumption that real word errors in a document do not fit into the context of the rest of the document. This hypothesis is a very important one and is the basis of our approach as well. The authors use a set of rules to generate possible spelling variations of the identified malapropism. Each possible correction is then evaluated with its neighbouring words for semantic similarity, and the distance metric used is Jiang-Conrath Distance based on WordNet. While a lot of the intuition in this paper is valid, the techniques utilized are quite outdated. In the last decade, word embeddings have emerged and cosine similarity between word vectors has been shown to be a much more reliable metric of word similarity. Further more, using a word to word distance does not capture the context of the entire system

The CoNLL-2014 shared task [18] was focused on grammatical error correction and most attempts were language model based. Eric Mays et al.[16] came up with a statistical approach to address contextual error correction. Swadha Gupta et al.[8] uses a conjunction of trigram approach and Bayesian model trained on Brown corpus for real word error correction. These classical techniques use either n-grams [22] or a probabilistic noisy channel model [10]. These methods perform reasonably well but have their own limitations. N-grams typically are unable to represent long term context. Probabilistic methods can be highly powerful but depend heavily upon the corpus used for training.

In recent times, with the advent of Deep Learning and its applications, contextual spelling error correction has been approached with the latest of neural network models. This is a parallel to the effort that has been happening in speech recognition domain. J Guo et al.[6] investigated attention based sequence to sequence models in speech recognition. Tao Ge et al.[4] investigated the sequence to sequence models in automatic error correction. Recently Chris J Lu et al.[15] used word2vec based embedding for health related questions. Similarly, Z Yuan et al. [28] used neural machine translation for grammatical error correction. P Etoori et al[2] built a RNN based approach for automatic error correction for low resource Indian languages.

Ziang Xie et al.[27] uses a character based attention model towards automatic error correction. Sina Ahmadi[1] models error correction as a mono-lingal machine translation model. In [11], a nested RNN model is used. Shaona Ghosh et al.[5] propose an architecture which consists of a combination of character level CNN and gated recurrent unit (GRU) encoder along with a word level gated recurrent unit (GRU) attention decoder. They even use a relatively smaller corpus(in comparison to the training corpora utilized by most deep learning architectures) of 12 million words.

Comparatively, very few research attempts were made to use knowledge for automatic error correction. Mathew E Peters et al[20] integrate WordNet and a subset of wikipedia into BERT and come up with knowBERT as a knowledge enhanced embedding. However, no attempt has been made to use this in word error correction. Yuan Ling et al.[14] uses knowledge enhanced named entity recognition in biomedical text.

2.2 Sentence Completion

Given a sentence with a blank in it and several options, "sentence Completion" is the task of predicting the appropriate word to fill that blank with respect to the surrounding context.

1. I have seen it on him , and could _____ to it.
   (a) write
   (b) migrate
   (c) climb
   (d) swear
   (e) contribute

Fig. 1: An example of the sentence completion task

The Microsoft Research Sentence Completion Challenge[29] of 2011 had the aim of providing a common metric for evaluation of language models on the spelling completion task. The Figure 1 shows an example of such a task. Further information on the dataset and the nature of the task can be found in the dataset section. The authors themselves suggest an n-gram method and a Latent Semantic Analysis based approach for the task. Geoffrey Zweig et al [30] analyze various other approaches, predominantly modelled around the use of vectors generated using LSA. One of these methods is Total Word Similarity, which is the basis of our model in this work. Total word similarity between a candidate word and the target sentence is defined as sum of cosine similarity between the candidate word and each word of the target sentence. However, most of these methods rely on training on similar data. The Microsoft Dataset consists of sentences taken from Project Gutenberg texts, and the LSA vectors are generated by training on a similar dataset of texts. However, considering the nature of word errors being investigated, one can not depend on this sort of training to generate word vectors. We hence resort to pretrained word embeddings, as described in the following section.

The current state of the art for this task can be attributed to Aubrie Woods[26]. The author uses a “point wise mutual information” method to generate word vectors for the words. They then perform total word similarity. However,instead of weighting each word evenly as is done with normal standard total word similarity, the authors assign different weights to each word according to the importance it has, on the basis of POS tags and features derived from dependency parsing. We attempt to use this sort of weighting in our model.

2.3 Knowledge Graphs and Knowledge Enhancement

Knowledge Graphs are graphical implementation of knowledgebases. These are large networks of entities (nodes) and their semantic relationships (edges). This kind of graph consists of triples i.e. subject, predicate, object which are arranged in a hierarchy. They serve as a powerful tool that changes the way we do data integration, search analytics, and context-sensitive recommendation. Knowledge enhancement is the concept of incorporating the information from knowledge bases and knowledge graphs into NLP models. Several works have highlighted how Knowledge enhancement can improve performance in various ways. Matthew E. Peters et al.[19] carry our knowledge enhancement on BERT(Bidirectional Encoder Representations from Transformers) to imporve its perplexity and fact-recalling capabilities , among other things. Tianda Li et al.[12] have done several experiments on the importance of knowledge enhancement for the task of Natural Language Inference and conclude that it indeed leads to imporved performance in several aspects.

ConceptNet Numberbatch[24] is a system of word embeddings which incorporate information from a knowledge graph called conceptNet into Glove and Google Word2Vec word embeddings . As described in [25], the authors have shown superior performance of Conceptnet Numberbatch in several tasks compared to standard embeddings. They are easy to use as they can simply be used as a replacement for the original set of word embeddings. In this work, we utilize conceptNet numberbatch embeddings

3 Methodology

In our approach we use the hypothesis that real word errors are not semantically cohesive [9] with respect to their context. We also connect this hypothesis with the task of sentence completion, where the contextually correct word from a set of options is to be selected to fill a blank in a sentence.

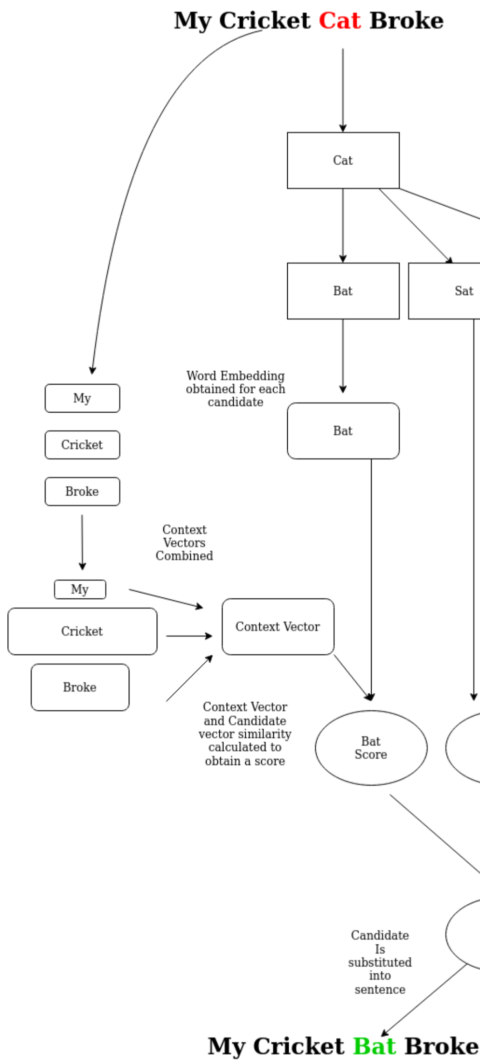

Fig. 2: Choosing candidate word based on semantic context and edit penalty

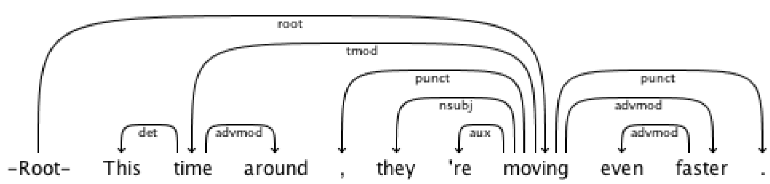

Fig. 3: A sample dependency parse showing the dependencies words in a sentence may have on each other

The model we propose here is inspired by the one proposed by Pieter Fivez et al.[3], which utilizes a word embeddings based approach to represent context and predict the correct spelling by identifying the spelling variation which is most similar to the context vector. This approach is a variant of Total Word Similarity, even though the name of this algorithm is not mentioned explicitly in the work. The word embeddings used are trained on medical text using a skip gram model. The model performs well on the task of performing spelling correction on clinical free-text. In our work, instead of the medical domain specific word embeddings generated by training, we propose using pre-trained word embeddings which have a generic purpose. We also add the concept of weighting of the context words so as to produce a more accurate representation of the context, by applying more and less weight to more and less important words. To increase the performance, we consider the knowledge context as well taking advantage of Concepnet[24].

Our approach is based on four main components. The first three components are ways of representing the context.

- Semantic Context: Word Vector using distributional semantics
- Weighted context : Application of Appropriate Weights to Context words
- Knowledge Context : Knowledge Enhanced Word Vector
- Penalty based on edit distance: This is not a way of improving the generation of the context vector but the main building block in existing spelling error correction in software.

The Figure 2 provides an example of how choice of candidate words can be done solely based on semantic context and edit penalty.

3.1 Semantic Context and the Total Word Similarity Algorithm

Word Embeddings represent words as dense unit vectors of real numbers, where vectors that are close together are semantically related. This approach to machine learning about semantics is sometimes referred to as distributional semantics as words which are used in similar context and in similar fashion end up being close together in embedding space. By adding up all the vectors(embeddings) that form the context of a word we can arrive at a vector which is a reasonable representation of the context. By finding the cosine similarity between the word vector and the context vector, we obtain a score of how well a given word fits in its context. By calculating the scores for all possible candidates for replacing this word, we can identify which word best fits in the context. We utilise this method for determining the appropriate correction for the error word.

3.2 Weighted Context

While simply summing up the context vectors results in a good representation of the context, it is possible to obtain a better representation by finding the weighted sum of the context vectors, where the weight of each word is proportional to its importance in the context w.r.t to the error word. We use two types of weighting in this work.

- Reciprocal Weighting Similar to the weighting used in [3], applies a weight of 1/d to the context word in question, where d represents the distance to the error word in number of words. It is based on the simple assumption that words which are farther away from the error word are likely to be less important in representing the context with respect to the error word.
- Syntactic Context Based on the algorithm used in [26] for applying weights to PMI values, we weight context word vectors based on their belonging to the following feature sets:

  1. **Reduced Context:**This feature set consists of words in the sentence after removing determiners, coordinating conjunctions, pronouns and proper nouns. These words are identified in [26] as not important
  2. **Dependencies:**Sentence words that share a semantic dependency with the candidate word(s) are included in this set. Absent from the set of dependencies are words removed during the first phase. The Figure 3 shows such a dependency graph produced by a dependency parser.
  3. **Keywords:** Providing the model with a collection of salient tokens effectively increases the tokens' associated weights. Using common nouns is an effective method to do this.

The weights applied to each context word is then simply the number of the above feature sets related to it. Stanford NLP Group's Stanza toolkit[21] is used to carry out all POS tagging and Dependency parsing.

3.3 Knowledge Context

In some cases simply utilizing information obtained from distributional semantics is not enough to adequately represent words in vector space. Recent research[25] has shown that enhancing word embeddings with knowledge helps to improve the performance of many natural language processing task. Instead of calculating path distance between different concepts in the knowledge graph, Concept Numberbatch provides a ready to use word embeddings that incorporate knowledge. It uses Glove and Word2Vec embeddings fortified with knowledge from conceptnet and are shown to outperform several word embeddings. We use these embeddings here in the same manner as we use normal word embeddings.

3.4 Edit Penalty

Edit Penalty refers to the penalizing of every candidate by dividing the final scores by their edit difference from the original error word. This is an important building block in all spelling checkers and operates based on the assumption that a human is more likely to make a smaller number of errors than a larger number with respect to the word he/she intended to type.

We incorporate all these features and compare and contrast to obtain the best results. The Figure 4 shows the high level flow of the model though it does not explicitly mention the syntactic and knowledge context. The Figure 5 shows the overall algorithm.

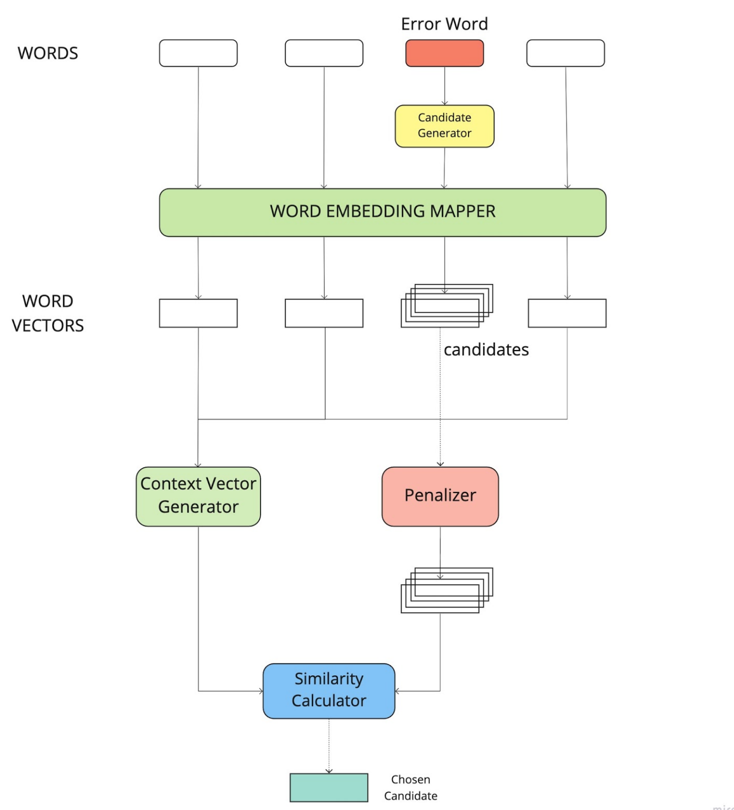

Fig. 4: The high level flow of our model

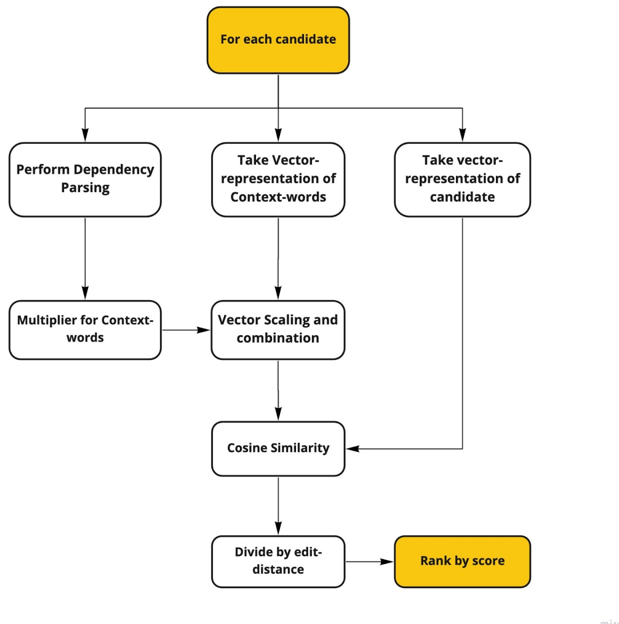

Fig. 5: Algorithm flow of our model

4 Dataset

Our test data set consists of 825 sentences with 4 candidates, exactly one being correct. The original MSR Sentence Completion Challenge Data [29], consists of 1,040 sentences, Each sentence consists of one ground truth and 4 imposter sentences where a specific word has been replaced with a syntactically correct but semantically incorrect impostor word. We modified the original data set to align with our task of real-word error correction. The Table 1 lists the source, size and description of the dataset used.

Source	Size	Description
Microsoft Research Sentence Completion Dataset	825 sentences	Real-Word candidate generation, Edit-distance, Randomized Selection

Table 1: Source, size and description of the dataset

Our data set consists of sentences from the MSR dataset, replacing specific manipulated words with blanks, each with 4 options, among which only one option is correct. We generated real-word candidates for each of the correct options within 2 Damerau-Levenshtein distance from the correct option. We used Wordnet synset as the dictionary and a list of all the words within Damerau- Levenshtein distance of 2 from the correct option were generated from the dictionary. We randomly choose 3 options from the generated list, discarding lists with less than 3 options, along with the corresponding sentence. For example,

- Generated dataset options : citation, titration, saturation, actuation, situation
- Original dataset options : situation, fence, food, instrument, building

As mentioned in the methodology section, all the information our model required to be able to make predictions was found within the pre-trained word embeddings. No additional training was done on any corpus.

5 Results

GloVe%	Uniform Weighting	Reciprocal Weighting	Syntactic Context Weighting
Without Edit Penalty	67.1 **(Baseline)**	66.7	63.7
With Edit Penalty	77.2	78.0	75.3

Table 2: Percentage accuracy for GloVe-enhanced real world error correction.

ConceptNet (Knowledge Enhanced)%	Uniform Weighting	Reciprocal Weighting	Syntactic Context Weighting
Without Edit Penalty	65.1	66.5	69.9
With Edit Penalty	76.5	77.3	**80.0**

Table 3: Percentage accuracy for ConceptNet-enhanced real world error correction.

The Table 2 and Table 3 lists the results obtained. Embedding like Glove is used for ascertaining the semantic context. Knowledge context is incorporated by Conceptnet numberbatch embedding. Syntactic context is used as one of the weight options in our algorithm. Edit penalty is a building block that is used an an option as well.

After running the experiments on the different models, we find that we achieve that the best results with the model that uses the ConceptNet embedding, syntactic context weighting and edit distance penalty to successively correct the maximum number of real world errors in the data set chosen.

While both word embeddings perform very similarly, it is the conceptnet numberbatch embeddings which were responsible for our best result. Our hypothesis is that the additional knowledge possessed by the conceptnet numberbatch embeddings (due to its use of the ConceptNet semantic graph) made a difference and allowed better representation of the words in the sentence and in turn the context of the sentence.

Reciprocal weighting impacts performance very slightly, but positively, showing that while applying weights to different words that are inversely proportional to the distance from the real world error definitely matters, the representation of context using reciprocal weighting is not significantly better than that using uniform weighting.

However, syntactic context based weighting identifies the most important words in the sentence and weights the important words more significantly, and this in conjunction with the knowledge advantage possessed by conceptnet results in models with higher accuracy.

Applying an Edit Penalty improves the performance of the model by a very significant amount. This is due to the fact that people making real world errors tend to make errors that are slightly different from the actual word and hence candidates generated from the model with smaller edit distances are more likely to be the actual word than other candidates with greater edit distances.

6 Conclusion

The outcome of this work is a successful model based on the Total Word Similarity algorithm which corrects Real Word Errors, those errors in which a word is lexically correct or syntactically correct in the sentence it belongs to, but is contextually incorrect.

We enhance the model far beyond its baseline of 60% using the vanilla Total Word Similarity algorithm, by incorporating the semantic context, syntactic context and knowledge context of a sentence into its functioning. Knowledge context is brought by using the knowledge enhanced conceptnet numberbatch as the primary word embeddings, which have been shown to be perform better compared to other more conventional choices of word embeddings. Dependency Parsing and resultant feature set based techniques, along with reciprocal-weighting are incorporated for providing syntactic context and allowing the model to make use of the structure of the sentence around the error to improve its prediction. Edit-penalty, a time-tested basic building block of standard spelling checkers is also incorporated in order to further enhance the model's performance.

We test the model on a version of the Microsoft Research Sentence Completion task,which has been appropriately modified for performing real word error correction tasks. We combine the techniques mentioned above, which encompass several ways of representing the context of a sentence, and provide our model with the best information possible, to achieve our best accuracy of 80%.

As a next step, we plan to extend the work by creating a larger dataset and investigating the effectiveness in speech to text translation that shows lot of errors due to homophones and can immensely benefit from this work [17] [7].

References

[1] Ahmadi, S.: Attention-based encoder-decoder networks for spelling and grammatical error correction. arXiv preprint arXiv:1810.00660 (2018)

[2] Etoori, P., Chinnakotla, M., Mamidi, R.: Automatic spelling correction for resource-scarce languages using deep learning. In: Proceedings of ACL 2018, Student Research Workshop, pp. 146–152 (2018)

[3] Fivez, P., Šuster, S., Daelemans, W.: Unsupervised context-sensitive spelling correction of clinical free-text with word and character n-gram embedding. In: 16th Workshop on Biomedical Natural Language Processing of the Association for Computational Linguistics, pp. 143–148 (2017)

[4] Ge, T., Wei, F., Zhou, M.: Reaching human-level performance in automatic grammatical error correction: An empirical study. arXiv preprint arXiv:1807.01270 (2018)

[5] Ghosh, S., Kristensson, P.O.: Neural networks for text correction and completion in keyboard decoding. arXiv preprint arXiv:1709.06429 (2017)

[6] Guo, J., Sainath, T.N., Weiss, R.J.: A spelling correction model for end-to-end speech recognition. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5651–5655. IEEE (2019)

[7] Gupta, A., Goyal, P., Sarkar, S., Gattu, M.: Fully contextualized biomedical NER. In: L. Azzopardi, B. Stein, N. Fuhr, P. Mayr, C. Hauff, D. Hiemstra (eds.) Advances in Information Retrieval - 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14-18, 2019, Proceedings, Part II, *Lecture Notes in Computer Science*, vol. 11438, pp. 117–124. Springer (2019). DOI 10.1007/978-3-030-15719-7_15. URL https://doi.org/10.1007/978-3-030-15719-7_15

[8] Gupta, S., et al.: A correction model for real-word errors. Procedia Computer Science **70**, 99–106 (2015)

[9] Hirst, G., Budanitsky, A.: Correcting real-word spelling errors by restoring lexical cohesion. Natural Language Engineering **11**(1), 87–111 (2005)

[10] Kernighan, M.D., Church, K.W., Gale, W.A.: A spelling correction program based on a noisy channel model. In: Proceedings of the 13th conference on Computational linguistics-Volume 2, pp. 205–210. Association for Computational Linguistics (1990)

[11] Li, H., Wang, Y., Liu, X., Sheng, Z., Wei, S.: Spelling error correction using a nested rnn model and pseudo training data. arXiv preprint arXiv:1811.00238 (2018)

[12] Li, T., Zhu, X., Liu, Q., Chen, Q., Chen, Z., Wei, S.: Several experiments on investigating pretraining and knowledge-enhanced models for natural language inference. CoRR **abs/1904.12104** (2019). URL http://arxiv.org/abs/1904.12104

[13] Lieberman, H., Faaborg, A., Daher, W., Espinosa, J.: How to wreck a nice beach you sing calm incense. In: Proceedings of the 10th international conference on Intelligent user interfaces, pp. 278–280 (2005)

[14] Ling, Y., Hasan, S.A., Farri, O., Chen, Z., van Ommering, R., Yee, C., Dimitrova, N.: A domain knowledge-enhanced lstm-crf model for disease named entity recognition. AMIA Summits on Translational Science Proceedings **2019**, 761 (2019)

[15] Lu, C.J., Aronson, A.R., Shooshan, S.E., Demner-Fushman, D.: Spell checker for consumer language (cspell). Journal of the American Medical Informatics Association **26**(3), 211–218 (2019)

[16] Mays, E., Damerau, F.J., Mercer, R.L.: Context based spelling correction. Information Processing & Management **27**(5), 517–522 (1991)

[17] Murphy, T., McIntosh, T., Curran, J.R.: Named entity recognition for astronomy literature. In: L. Cavedon, I. Zukerman (eds.) Proceedings of the Australasian Language Technology Workshop, ALTA 2006, Sydney, Australia, November 30-December 1, 2006, pp. 59–66. Australasian Language Technology Association (2006). URL https://www.aclweb.org/anthology/U06-1010/

[18] Ng, H.T., Wu, S.M., Briscoe, T., Hadiwinoto, C., Susanto, R.H., Bryant, C.: The conll-2014 shared task on grammatical error correction. In: Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task, pp. 1–14 (2014)

[19] Peters, M.E., Neumann, M., IV, R.L.L., Schwartz, R., Joshi, V., Singh, S., Smith, N.A.: Knowledge enhanced contextual word representations. In: K. Inui, J. Jiang, V. Ng, X. Wan (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pp. 43–54. Association for Computational Linguistics (2019). DOI 10.18653/v1/D19-1005. URL https://doi.org/10.18653/v1/D19-1005

[20] Peters, M.E., Neumann, M., Logan, I., Robert, L., Schwartz, R., Joshi, V., Singh, S., Smith, N.A.: Knowledge enhanced contextual word representations. arXiv preprint arXiv:1909.04164 (2019)

[21] Qi, P., Zhang, Y., Zhang, Y., Bolton, J., Manning, C.D.: Stanza: A python natural language processing toolkit for many human languages. CoRR **abs/2003.07082** (2020). URL https://arxiv.org/abs/2003.07082

[22] Samanta, P., Chaudhuri, B.B.: A simple real-word error detection and correction using local word bigram and trigram. In: Proceedings of the 25th conference on computational linguistics and speech processing (ROCLING 2013), pp. 211–220 (2013)

[23] Singh, S., Singh, S.: Review of real-word error detection and correction methods in text documents. In: 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), pp. 1076–1081. IEEE (2018)

[24] Speer, R., Chin, J., Havasi, C.: ConceptNet 5.5: An open multilingual graph of general knowledge. pp. 4444–4451 (2017). URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14972

[25] Speer, R., Lowry-Duda, J.: Conceptnet at semeval-2017 task 2: Extending word embeddings with multilingual relational knowledge. CoRR **abs/1704.03560** (2017). URL http://arxiv.org/abs/1704.03560

[26] Woods, A.: Exploiting linguistic features for sentence completion. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 438–442. Association for Computational Linguistics, Berlin, Germany (2016). DOI 10.18653/v1/P16-2071. URL https://www.aclweb.org/anthology/P16-2071

[27] Xie, Z., Avati, A., Arivazhagan, N., Jurafsky, D., Ng, A.Y.: Neural language correction with character-based attention. arXiv preprint arXiv:1603.09727 (2016)

[28] Yuan, Z., Briscoe, T.: Grammatical error correction using neural machine translation. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 380–386 (2016)

[29] Zweig, G., Burges, C.J.: The microsoft research sentence completion challenge. Tech. Rep. MSR-TR-2011-129 (2011). URL https://www.microsoft.com/en-us/research/publication/the-microsoft-research-sentence-completion-challenge/

[30] Zweig, G., Platt, J.C., Meek, C., Burges, C.J.C., Yessenalina, A., Liu, Q.: Computational approaches to sentence completion. In: The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 1: Long Papers, pp. 601–610. The Association for Computer Linguistics (2012). URL https://www.aclweb.org/anthology/P12-1063/