Knowledge Graph Fusion for Language Model Fine-Tuning

Nimesh Bhana
Computer Science and Applied Mathematics
University of Witwatersrand
Johannesburg, South Africa
2371061@students.wits.ac.za

Terence L. van Zyl
University of Johannesburg
Institute for Intelligent Systems
Johannesburg, South Africa
tvanzyl@gmail.com

Abstract—Language Models such as BERT (Bidirectional Encoder Representations from Transformers) have grown in popularity due to their ability to be pre-trained and perform robustly on a wide range of Natural Language Processing tasks. Often seen as an evolution over traditional word embedding techniques, they can produce semantic representations of text, useful for tasks such as semantic similarity. However, state-of-the-art models often have high computational requirements and lack global context or domain knowledge which is required for complete language understanding. To address these limitations, we investigate the benefits of knowledge incorporation into the fine-tuning stages of BERT. An existing K-BERT model, which enriches sentences with triplets from a Knowledge Graph, is adapted for the English language and extended to inject contextually relevant information into sentences. As a side-effect, changes made to K-BERT for accommodating the English language also extend to other word-based languages. Experiments conducted indicate that injected knowledge introduces noise. We see statistically significant improvements for knowledge-driven tasks when this noise is minimised. We show evidence that, given the appropriate task, modest injection with relevant, high-quality knowledge is most performant.

Index Terms—Language Model, BERT, Knowledge Graph

I. INTRODUCTION

One major challenge in Natural Language Processing (NLP) is assessing the semantic similarity between pairs of text. Since natural language is versatile and ambiguous, it makes rule-based methods difficult to define [1]–[3]. The polysemous property of most words, along with the use of synonyms, abbreviations, negations, etc., pose challenges for most rule-based systems [4]. Additionally, for domain-specific fields such as sports or medicine, knowledge of the underlying concepts is critical to computing accurate similarity measures [5]–[7].

Sources of knowledge such as ontologies, thesauri, and knowledge bases have often been used to improve the performance of various NLP systems [8]–[10]. However, successful integration between the NLP system and knowledge source often requires feature engineering. Furthermore, due to the structural inconsistencies between various knowledge sources, the integration is done in a tightly coupled manner.

The application of pre-trained Language Models for downstream tasks has produced state-of-the-art results in recent years [11], [12]. However, the most performant models often contain billions of parameters, requiring expensive computational equipment. It is therefore vital to make such models as efficient as possible, which will also lead to less energy consumption.

Knowledge-Enhanced Models have demonstrated the power of leveraging external knowledge to improve the performance of pre-trained Language Models while retaining model size. The approach by Liu et al. (2020) [13] combines information from Knowledge Graphs with the ubiquitous BERT model by Delvin et al. (2018) [14] as a means to complement input text with additional domain knowledge or contextual information. Although their results demonstrate benefits for knowledge-driven and domain-specific tasks, the selection mechanism for knowledge injection does not consider sentence context, which may lead to injecting irrelevant information. Our study extends K-BERT by modifying the $K_{Query}$ mechanism to consider semantically important information and assessing its performance on both open-domain and domain-specific tasks. The utilisation of Wikidata as the Knowledge Graph is done loosely to ensure interchangeability with other knowledge sources. Using ablation studies, we examine the type of knowledge most beneficial to the fine-tuning process and associated limitations. Lastly, since K-BERT was initially developed for the Chinese language, there is a lack of research indicating the level of versatility in the underlying architecture when applied to other languages.

To this end, the article presents the results of our investigation, which show that:

1) K-BERT can be successfully adapted to the English domain and possibly other languages.
2) A modified $K_{Query}$ to only inject semantically related information can be beneficial.
3) Inclusion of external knowledge introduces noise.
4) In the absence of noise, external knowledge injection is beneficial to knowledge-driven tasks.

II. RELATED WORK

To semantify input text, Pilehvar et al. (2017) [15] integrates sense-level knowledge into a CNN text classifier by disambiguating input text using information from a WordNet semantic network. This aim is to disambiguate text before being fed into a system. They highlight that simple input
disambiguation can bring about a performance gain for longer texts but is ineffective for shorter texts.

More modern research investigates the concept of incorporating external knowledge into pre-trained Language Models. Nguyen et al. (2019) [16] can achieve performance that slightly outperformed BERT. They exploit the similarity of word contexts built by word embeddings and semantic relatedness between concepts based on external knowledge sources such as WordNet and Wiktionary. Using their approach, they can more accurately assess the similarity between two short texts. However, their approach is supervised and requires feature engineering.

Alternative approaches involve amalgamating a separate word embedding with its associated node embedding from a Knowledge Graph. Construction of embeddings from Knowledge Graphs utilises Graph Convolutional Networks to produce embeddings for every entity or node. Since these embeddings are produced separately, their vector spaces are inconsistent with each other. The K-BERT model by Liu et al. (2020) [13] avoids this by injecting information before the embedding process. They inject knowledge into sentences using Chinese Knowledge Graphs. These knowledge-rich sentences are then passed through to a BERT-based model to learn more meaningful sentence representations. This work extends K-BERT by adapting the model to the English domain.

### III. Method: English Adapted K-BERT

#### A. Knowledge Graph

This work employs Wikidata as the primary knowledge source to retrieve information. As with most knowledge graphs, it can be stored in a triplet format, i.e. (subject, predicate, object). An example statement could be (Michelle Obama, wifeOf, Barack Obama). All properties and relationships of data items in Wikidata can be described with the triple format. An advantage of using Wikidata is that data items are inherently unambiguous.

1) **Data Preprocessing.** Due to the size of Wikidata, integration of the entire Knowledge Graph is expensive and infeasible. We reduce this size by only considering English data items in domains: business, sports, humans, cities, and countries. Furthermore, only properties belonging to the set \{label, alias, description, subclass of, instance of\} are used. These properties were selected because they maximise descriptive detail while minimising storage requirements.

#### B. Term-Based Sentence Tree

Since Chinese is logographic, K-BERT was designed to work on a character level. Model inputs such as the sentence tree and visible matrix must be adapted to accommodate the alphabetic English language on a word level. However, a further extension to the multi-word-level or term-level is also necessary as there are cases where entities may span across two or more words. Knowledge injection is done per group of tokens instead of a single token. Specifically, given an input sentence \(s = \{w_0, w_1, \ldots, w_n\}\) comprised of tokens \(w_i\) in vocabulary \(V\), enclose contiguous related tokens \(w_a\) to \(w_q\) together in a group \(\{\}\). It is not required that a group consists of multiple tokens. Therefore, we can have a particular group \(w_j\), where \(w_j\) has no other continguously related tokens. Knowledge is injected per group of related tokens to produce an output sentence tree \(t\):

\[
t = \left[\{w_0\}, \{w_1\}, \ldots, \{w_i\}, \ldots, \{w_n\}\right]
\]

where \(w \in V\) is the set of entities in the Knowledge Graph \(K\), \(r \in V\) is the set of relations/properties in \(K\), and \(k\) represents the number of triplets inserted. Braces enclose related tokens, and their knowledge, if any, is inserted directly afterwards. Construction of the visible matrix and positional encoding can be done similarly. However, an additional “unrolling” step is required to associate positions per token. As the last step, the visible matrix ensures token groups and their knowledge all “see” each other by appointing a value of 1 in their cell. This optimisation alleviates duplicating the same knowledge for each token in the group, thereby minimising sequence lengths provided to the model.

#### C. Contextualised Knowledge Injection

In this section, we highlight the approach taken to inject knowledge into sentences. For an input sentence \(s\), perform named entity recognition to extract entities such as names of people, places, sports teams, etc. We accomplish this with a small pre-trained model from spaCy based on a transition-based parser [17], [18] and an “embed, encode, attend, predict” framework [19]. After that, the extracted entities are queried from the Knowledge Graph to retrieve a list of triplets. This processing is done by \(K_{query}\):

\[
E = K_{query}(s, K)
\]

where \(E\) is a collection \([\{w_i, r_{ij}, w_{ij}\}, \ldots, \{w_i, r_{jk}, w_{jk}\}]\) of triplets. The function, \(K_{inject}\), then injects \(E\) into the correct position and generates the sentence tree \(t\) as:

\[
t = K_{Inject}(s, E)
\]

However, additional processing to find the most relevant triplets is done before injection occurs. Hence, instead of injecting the first \(n\) entities as done by Liu et al. (2020) [13], we use a pre-trained Transformer model to determine which retrieved entities from \(E\) are most relevant to the current sentence \(s\). This injection is done by concatenating each entity and their related properties into a single text sequence \(seq_i\). The pre-trained Transformer model \(T\) then generates contextualised embeddings \(\text{emb}_i\) for each sequence \(seq_i\) using:

\[
\text{emb}_i = T(seq_i)
\]

1https://www.wikidata.org/wiki/Help:Items (2016)
2https://www.wikidata.org/wiki/Wikidata:Main_Page (2022)
3https://spacy.io/
Contextualised embedding $\text{emb}_s$ is generated for sentence $s$ in the same manner. Similarity between embeddings is computed using cosine similarity metric $\|i\|_\text{cos}$. The entity corresponding to the most similar embedding is selected to be injected using:

$$\max_{i \in I} \max_{j \in J} \left( \|\text{emb}_{t_i}, \text{emb}_{s_j}\|_\text{cos} \right)$$

where $I$ is the set of distinct entities. In order to limit the amount of noise injected, we introduce a threshold parameter such that information is only injected if $\|\text{emb}_{t_i}, \text{emb}_{s_j}\|_\text{cos} > \text{threshold}$.

D. Corrections & Optimisations

1) Sequence Truncation: Truncation has been modified to be more “equal” when sentence pairs are fed into K-BERT. Given a max_length, the number of tokens in S1 and S2 is each limited to size max_length / 2. Unoccurred positions are assigned as leftover and given to the remaining sentence that requires it. The difference is illustrated in Fig. 1 for sentences S1 and S2.

![Fig. 1. Comparison of truncation methods](image)

2) Memory Usage: Dimension of the visible matrix is (max_length x max_length). Implies that memory usage grows exponentially as sequence length increases. Such growth can become infeasible for large datasets. Since the visible matrix is symmetrical, we optimise memory usage by immediately storing non-duplicate entries from the visible matrix into a vector of size $N(N+1)/2$. When the matrix is required to be fed into K-BERT, we convert the vector back into a symmetrical matrix - done per batch. Lastly, since the visible matrix strictly contains binary values (“visible”, “invisible”), we restrict values to one byte each. With the described optimisations, the visible matrix memory consumption is reduced by a factor of four at minimum.

IV. Evaluation

A. Datasets

This work focuses on the task of semantic similarity. We consider two widely used and credible datasets. In the public domain, the Semantic Textual Similarity Benchmark (STS-B) [20] was used.

Since domain-specific datasets equivalent to STS-B are expensive to construct, we consider a text classification dataset ag_news_subset [21]. This dataset contains extracts from news articles, and the task is to classify articles into the correct category from set {World, Sci/Tech, Sports, Business}.

| Dataset     | Train | Validation | Test  |
|-------------|-------|------------|-------|
| STS-B       | 5,749 | 1,500      | 1,379 |
| ag_news_subset | 110,400 | 9,600      | 7,600 |

B. Experimental Setup

This section describes the experimental setup for comparing BERT to K-BERT. Hyperparameters and additional configurations were kept consistent across both models. The software used and steps taken to attain the results is made available at https://github.com/Nimesh18/K-BERT-ENG. Each model was fine-tuned for ten epochs. The best-performing model on the validation set was saved and used to evaluate the test set. Batch sizes of 16 and 32 were used for STS-B and ag_news_subset, respectively. Being cognizant of the catastrophic forgetting problem [22], [23], only modest learning rates were considered as hyperparameters ($2e^{-5}$ to $5e^{-5}$). Based on the validation set, the best performing learning rates were $4e^{-5}$ and $5e^{-5}$ for STSb and ag_news_subset, respectively.

Hyperparameters were tuned according to the best performance on the validation set. The best threshold parameter, described in section III, was 0.5 for STS-B and 0.6 for ag_news_subset. Our available resources necessitated a balance between computational requirements and performance. Therefore, a maximum sequence length of 128 for ag_news_subset and 256 for STS-B was set. Additionally, the pre-trained BERT BASE was used instead of the better performing, yet computationally demanding, BERT LARGE.

Adam optimiser with weight decay and scheduler parameters remained unchanged from the work by Liu et al. (2020) [13] and Devlin et al. (2018) [14]. That is, Adam with L2 weight decay of 0.01, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and linear decay of learning rate.

This work uses the pre-defined STS-B dataset splits as indicated in Table I but shuffles the training set before every run. The same is true for ag_news_subset. However, since it does not have a predefined validation set, the last 8% of the training set served as the validation set throughout all experiments.

C. Experiment

In order to limit biases and highlight statistical significance, results are averaged over ten runs. Each of these runs uses consecutive seeds (8 to 17) to ensure the reproducibility of results.

We aim to attain insight into i) what type of knowledge is most beneficial and ii) at what point does knowledge incorporation become detrimental to performance. The following experiments are performed to gain more insight:

1) Knowledge Ablation: The type of knowledge injected (as described in Section III-A) can be categorised into aliases, categorical, and descriptive information. To identify what information is most beneficial, we exclude
each type from being injected into K-BERT and rerun experiments.

2) Knowledge-Gating: Since K-BERT requires sequences to be truncated passed a maximum sequence length, injecting additional information can lead to important information from the original sentence being truncated. To see the extent to which this is true, we perform Knowledge-Gating. That is, only inject information into sentences when sequence lengths are below the maximum sequence length - thereby avoiding truncation.

3) Manual Knowledge Injection: A manual selection of knowledge is performed to identify deficiencies in the automated similarity-based approach presented in $K_{\text{Inject}}$ for relevant knowledge fusion. The manual approach is expected to contain minimum noise since humans can match relevant knowledge to sentence context to a much higher degree. Manual Knowledge Injection is done for all dataset splits as described in detail below.

1) Manual Knowledge Injection: More specifically, the systematic approach for performing Manual Knowledge Injection requires manually matching extracted entities $E$ to the context of sentence $s$. When sentence $s$ lacks context, description information is typically selected to be injected. In situations where descriptive information would introduce noise or be inappropriate, we default to categorical or alias information. While most of this process is subjective, the primary objective of only injecting beneficial information for the problem at hand is maintained throughout. For example, the occurrence of an entity "Manchester United" in ag_news_subset will likely be associated with external knowledge "football club" - if the terms do not already exist in the original sentence $s$.

V. RESULTS

Table II shows the Knowledge Ablation results on the STS-B dataset. We report total Mean Squared Error (MSE) loss and Spearman correlation. The correlations are between the predicted and actual scores. Combinations of knowledge categories from set \{ALIAS, CAT, DESC\} were removed from K-BERT, and the results are presented here. The knowledge categories consecutively correspond to \{aliases, categorical information, descriptive information\}. Additionally, a model with minimal noise, K-BERT$_{\text{MANUAL}}$, indicates that knowledge selection is made manually by humans (as described in Section IV-C1) instead of the similarity-based approach of K-BERT.

For ag_news_subset, we report classification accuracies for BERT and all K-BERT ablation variations in Table III. Compared to STS-B, sequence length limitations is a more stringent factor for ag_news_subset due to its much longer paragraphs of text. The 128 sequence length limitation we impose on it has further exacerbated this factor. Knowledge-Gating experiments have been performed to investigate the effect of retaining maximal information from the original sentences. The results of this are shown in column Knowledge-Gating=on.

| Model       | MSE Loss   | $p$  | Spearman   | $p$  |
|-------------|------------|------|------------|------|
| BERT        | 1.662 / 2.014 ±0.054 | 0.6  | 89.67 / 85.73 ±0.49 |
| K-BERT      | 1.675 / 2.028 ±0.063 | 0.6  | 89.58 / 85.66 ±0.56 |

VI. ANALYSIS

To identify the significance of the results, we perform one-tail Student t-tests between test results from BERT and K-BERT. Results were recorded after repeating the experiment ten times using different seeds. Independent sample t-tests are performed between BERT results and K-BERT Knowledge Ablations. We begin this section with an analysis of the results of the STS-B dataset. Analysis of ag_news_subset results is done after that.

A. STS-B Analysis

Inspecting the K-BERT model results in Table II, we see an overall reduction in performance with the addition of knowledge in the BERT model. Thereby implying that the knowledge is introducing noise. It is expected that reducing this noise will improve performance. The Knowledge Ablation experiments described in section IV look to evaluate if this noise is the cause of the decline in performance.

After performing Knowledge Ablation, overall performance improves to a marginal degree compared to the standard K-BERT. Descriptive information tends to be the most beneficial type of knowledge to inject, with categorical information also providing some benefit. These insights can be identified by inspecting the changes in metrics when descriptive and categorical information is removed from the model. The ALIAS + CAT variant produced the lowest MSE loss, while on average, ALIAS + DESC had the highest correlation coefficients.

However, overall results for this dataset indicate that noise exists in all knowledge types. Moreover, a manual injection of knowledge could not improve performance at all. Similar to other research [24] which injects knowledge in pre-trained language models for the GLUE benchmark dataset [25], it can be concluded that the inclusion of knowledge for this problem is not beneficial.

B. AG News Subset Analysis

In contrast to STS-B, the ag_news_subset dataset appears to benefit more from the inclusion of knowledge. K-BERT has an improved average test accuracy compared to BERT, as seen in Table III. Removing noise through Knowledge Ablation does not benefit the test performance compared to K-BERT.
TABLE III: Knowledge Ablation and Knowledge-Gating results on ag-news_subset dataset. We report mean (μ-validation / μ-test ±σ) accuracies taken across ten runs, (p-values) in bold indicate a statistically significant improvement on the baseline: BERT

| Model          | Off  | Knowledge-Gating (p-value) | On   | (p-value)   |
|----------------|------|----------------------------|------|-------------|
| BERT           | 94.55 / 94.60 ±0.1947 |                              |      |             |
| K-BERT         | 94.55 / 94.65 ±0.2805  | (0.3518)                     | 94.58 / 94.69 ±0.1269 | (0.1328) |

After Removing

| Model | Off  | Knowledge-Gating (p-value) | On   | (p-value)   |
|-------|------|----------------------------|------|-------------|
| ALIAS | 94.60 / 94.60 ±0.2272 | (0.5039) | 94.52 / 94.67 ±0.1138 | (0.1854) |
| CAT   | 94.53 / 94.61 ±0.1144 | (0.4791) | 94.53 / 94.67 ±0.1474 | (0.2072) |
| DESC  | 94.54 / 94.63 ±0.1409 | (0.3517) | 94.62 / 94.63 ±0.2274 | (0.3911) |
| CAT + DESC | 94.61 / 94.59 ±0.2071 | (0.5538) | 94.55 / 94.60 ±0.2070 | (0.5290) |
| ALIAS + DESC | 94.53 / 94.59 ±0.1378 | (0.5640) | 94.61 / 94.67 ±0.0989 | (0.1745) |
| ALIAS + CAT  | 94.57 / 94.63 ±0.1585 | (0.3665) | 94.53 / 94.54 ±0.1113 | (0.7948) |

K-BERT\_MANUAL | 95.16 / 95.24 ±0.1361 | (<0.001) | 95.15 / 95.23 ±0.1824 | (<0.001) |

However, these test accuracies are generally improved with Knowledge-Gating=on, when truncation is avoided. As we have seen with STS-B, all knowledge types introduce noise. Descriptive and categorical information still appears to be the most beneficial for this classification problem, with aliases providing a more significant benefit than in STS-B.

Although the reported average accuracies indicate gradual performance improvements, the t-test results conclude that the difference is statistically insignificant. Further noise removal through K-BERT\_MANUAL, however, produces the best results compared to every K-BERT and BERT variation. When Knowledge-Gating=on, performance is negatively affected to a very marginal degree. This result highlights that, in the absence of noise, additional knowledge provides contextual information that can be more beneficial for this problem than the current information in the original sentences. Since the extracted entities for this problem, such as Intel, Microsoft, Apple, Toyota, Yankees occur relatively frequently within the dataset, the associated additional knowledge enforces computed embeddings to be aggregated into more well-defined clusters. With a p-value= 0.0, K-BERT\_MANUAL has shown a statistically significant benefit for the inclusion of knowledge.

In conclusion, the fusion of knowledge from the Wikidata Knowledge Graph has potential benefits. When inspecting Table III, K-BERT with Knowledge-Gating=on produces better results and has more significant t-test results for most models. However, all autonomous approaches and knowledge types introduce some noise which causes a decline in performance. When minimising the amount of noise, the STS-B dataset exhibits no benefit from the fusion of knowledge, while the ag-news_subset dataset produces a 0.7% improvement over BERT. Despite the potential benefits of incorporating external knowledge, achieving successful integration with an appropriate problem autonomously is non-trivial.

VII. CONCLUSION AND FUTURE WORK

This article presents an extended K-BERT model, adapted to accommodate the English language, along with a new Wikidata Knowledge Graph. The sentence tree and visible matrix have been changed to work on a term level instead of the character level meant for Chinese. Sequence truncation has been made to be done more equally between sentence pairs. Additional optimisations permit larger datasets to use less memory.

Knowledge Ablation studies indicate that while knowledge injection benefits performance on average, it introduces noise which hurts performance. Reducing the amount of noise leads to more significant results on the ag-news_subset dataset. However, the same does not hold for STS-B. As corroborated in [24], [26], knowledge-enhanced models such as ERNIE are unstable on smaller datasets such as STS-B and ultimately perform worse. The drop in performance can be attributed to STS-B benefitting more from improved linguistic representation than structured facts. Therefore, we conclude that given the appropriate problem, injecting knowledge sparingly with relevant, high-quality information is preferable.

Since the contextual mechanism in $K_{Query}$ was intended to work on the general text, replacing the Knowledge Graph with free text is possible. Additionally, the multilingual capability of Wikidata allows K-BERT to accommodate various other languages, given the appropriate pre-trained Language Model. Future work should explore advanced contextual mechanisms which consider additional factors other than similarity. Possibilities include a pre-disambiguation step or machine learning models to decide on the entity to inject.

REFERENCES

[1] E. Marsi and E. Krahmer, “Automatic tree matching for analysing semantic similarity in comparable text,” in Essential Speech and Language Technology for Dutch: Results by the STEVIN programme. 2013, pp. 129–145.

[2] M.-C. Marneffe and C. Manning, “The stanford typed dependencies representation,” COLING Workshop on Cross-framework and Cross-domain Parser Evaluation, Jan. 2008.

[3] S. Schuster and C. D. Manning, “Enhanced English Universal Dependencies: An improved representation for natural language understanding tasks,” in Proceedings of the Tenth International Conference on Lan-
Language Resources and Evaluation, European Language Resources Association, May 2016, pp. 2371–2378.

R. Krovetz, “Homonymy and polysemy in information retrieval,” in Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics, ser. ACL ’98/EACL ’98, Madrid, Spain: Association for Computational Linguistics, 1997, pp. 72–79.

M. Ormerod, J. Martínez del Rincón, and B. Devereux, “Predicting semantic similarity between clinical sentence pairs using transformer models: Evaluation and representational analysis,” JMIR Med Inform, vol. 9, May 2021.

O. O. Oni and T. L. van Zyl, “A comparative study of ensemble approaches to fact-checking for the fever shared task,” in 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), IEEE, 2020, pp. 1–8.

T. Manaka, T. Van Zyl, A. N. Wade, and D. Kar, “Using machine learning to fuse verbal autopsy narratives and binary features in the analysis of deaths from hyperglycaemia,” in Proceedings of SACAIR2021, vol. 1, 2022, pp. 90–106.

W. Majavu, T. L. van Zyl, and T. Marwala, “Classification of web resident sensor resources using latent semantic indexing and ontologies,” in 2008 IEEE International Conference on Systems, Man and Cybernetics, IEEE, 2008, pp. 518–523.

M. Rodriguez and M. Egenhofer, “Determining semantic similarity among entity classes from different ontologies,” IEEE Transactions on Knowledge and Data Engineering, vol. 15, pp. 442–456, Mar. 2003.

G. Majumder, D. P. Pakray, A. Gelbukh, and D. Pinto, “Semantic textual similarity methods, tools, and applications: A survey,” Computacion y Sistemas, vol. 20, pp. 647–665, Dec. 2016.

N. Kooverjee, S. James, and T. Van Zyl, “Inter- and intra-domain knowledge transfer for related tasks in deep character recognition,” in 2020 International SAUPEC/RobMech/PRASA Conference, IEEE, 2020, pp. 1–6.

D. Llamas, A. Tordesillas, and T. Von Zyl, “Author identification from handwritten characters using siamese cnn,” in 2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC), IEEE, 2019, pp. 1–6.

W. Liu, P. Zhou, Z. Zhao, et al., “K-BERT: enabling language representation with knowledge graph,” in The Thirty-Fourth AAAI Conference on Artificial Intelligence, 2020, pp. 2901–2908.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.

M. T. Pilevar, J. Camacho-Collados, R. Navigli, and N. Collier, “Towards a seamless integration of word senses into downstream nlp applications,” Oct. 2017, pp. 1857–1869.

H. T. Nguyen, P. H. Duong, and E. Cambria, “Learning short-text semantic similarity with word embeddings and external knowledge sources,” Knowledge-Based Systems, vol. 182, Oct. 2019.

G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, “Neural architectures for named entity recognition,” in Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Jun. 2016, pp. 260–270.

M. Honnibal, Parsing english in 500 lines of python, 2013.

——, Embed, encode, attend, predict: The new deep learning formula for state-of-the-art nlp models, 2016.

D. C. Lin, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, “Semeval-2017 task 1: Semantic textual similarity - multilingual and cross-lingual focused evaluation,” Aug. 2017.

X. Zhang, J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” in Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, ser. NIPS’15, MIT Press, 2015, pp. 649–657.

M. McCloskey and N. J. Cohen, “Catastrophic interference in connectionist networks: The sequential learning problem,” in ser. Psychology of Learning and Motivation, G. H. Bower, Ed., vol. 24, Academic Press, 1989, pp. 109–165.

C. Sun, X. Qiu, Y. Xu, and X. Huang, “How to fine-tune bert for text classification?” in Chinese Computational Linguistics, M. Sun, X. Huang, H. Ji, Z. Liu, and Y. Liu, Eds., Cham: Springer International Publishing, 2019, pp. 194–206.

N. Zhang, S. Deng, X. Cheng, et al., “Drop redundant, shrink irrelevant: Selective knowledge injection for language pretraining,” in Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, Aug. 2021, pp. 4007–4014.

A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, “GLUE: A multi-task benchmark and analysis platform for natural language understanding,” in Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 353–355.

Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, “Ernie: Enhanced language representation with informative entities,” in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Jul. 2019, pp. 1441–1451.