Washington University in St. Louis

Olin Business School

Deep Learning for Prediction of Business Outcomes

Final Project

Siamese Bidirectional LSTM model for resume & job description matching

Project Group 1

Bosan Hsu (MSBA)

Young Kim (MSBA)

October 18, 2024

Presentation recording:
https://drive.google.com/drive/folders/1-386UtENx9x28jVW5l3QRPt5eKcneDfS

Table of Contents

Abstract

We apply a Siamese approach of the Long Short-Term Memory (LSTM) network for labeled data consisted of pairs of resume and job description. Unmatching between resumes and job descriptions can attract a huge demand from both sides of job seekers and employers. Solving that business issue could be lucrative. First, in our model, we leverage Word2Vec word-embedding vectors complemented with synonymic information to the LSTMs, which use a fixed size vector to encode the underlying meaning expressed in a sentence. A model's architecture comprises two identical LSTM models, which have the same structure and parameters. After merging the two LSTM structures, we try to feed the output into dense layers. We build three different models for the performance comparison. Finally, we come up with the fitness between resume and job description on the test dataset. The final model's accuracy is 0.87.

1. Introduction

Despite the proliferation of online job portals, obtaining the right kind of information remains a challenge for both job seekers and employers. Job seekers find it difficult to intuitively determine which jobs among thousands of listings are the best fit for them. On the other hand, employers struggle to finalize the recruitment process within deadlines as they receive thousands of resumes. To overcome these two-sided challenges, it is necessary to streamline the entire job search and recruitment process. Recently, job-matching platforms have attempted to address this issue. These efforts mainly involve assigning matching scores based on machine learning algorithms. However, with the recent advancements in deep learning, particularly in natural language processing, it has become possible to assign more accurate matching scores. As a result, HR tech companies specializing in these job-matching algorithms are emerging.

Our goal is to develop a deep learning-based automated job-matching algorithm to assist both job seekers and employers. Our algorithm will recommend whether a job posting of employers and resume of job seekers are matching based on skills, tasks, experiences, and so on. The algorithm will help job seekers by providing personalized job recommendations that match their skills and experience, reducing the overwhelming task of sifting through numerous postings. For employers, this helps employers meet recruitment deadlines more easily and ensures that they find more suitable candidates from a large pool of applications, ultimately improving the quality of hires. In the end, this state-of-the-art technology for sure will be lucrative as it could attract potential demand from both sides in the job market.

2. Literature Review

2.1. Skip-gram model

The Skip-gram model, introduced by Mikolov et al. (2013), is a method for learning distributed word representations that capture syntactic and semantic relationships between words. It predicts the context words surrounding a target word, allowing for efficient training on large text corpora. Key features include negative sampling, which speeds up training by distinguishing between true context words and random "noise" words and subsampling frequent words to improve learning for less common words. Additionally, Skip-gram can learn representations for multi-word phrases and supports analogical reasoning through vector arithmetic. The model's simplicity and scalability make it effective for word-level tasks, though it is less suited for capturing complex sentence structures.

While the Skip-gram model efficiently captures word relationships through vector representations and demonstrates some success with analogical reasoning tasks, its limitations make it

inferior compared to the latest neural networks models. First, the Skip-gram model focuses on individual words and their relationships, often failing to capture the meaning of sentences or longer texts as effectively as models designed for sequence-based inputs, like a Siamese LSTM. In contrast, the Siamese LSTM can process and compare entire sequences, better capturing the deeper semantics required for tasks like resume and job description matching. Second, the Skip-gram predicts surrounding words based on a central word but does not consider the full structure or order of sentences, which is crucial for understanding text similarity. Third, although the Skip-gram model uses simple vector addition for compositionality (e.g., summing vectors for multi-word phrases), this method is limited in capturing complex, non-linear relationships between words and phrases. Finally, while the Skip-gram model performs well on analogy tasks (e.g., "Paris is to France as Berlin is to Germany"), this type of linear relationship learning is not sufficient for tasks requiring nuanced sentence or paragraph-level similarity.

In summary, the Skip-gram model, while effective for word-level tasks, lacks the depth and sophistication of the latest neural networks models when it comes to comparing complex texts, such as resumes and job descriptions, where capturing sequence-level relationships is key.

2.2. Supervised Sequence Labelling with Recurrent Neural Networks

The article titled "Supervised Sequence Labelling with Recurrent Neural Networks" by Alex Graves (2013) explores the use of recurrent neural networks (RNNs) for sequence labeling tasks such as speech and handwriting recognition. The core focus is on RNNs' ability to capture dependencies across sequences, especially with the integration of Long Short-Term Memory (LSTM) units, which allow for the preservation of information over longer time spans. The article further introduces advanced techniques like bidirectional RNNs and Connectionist Temporal Classification (CTC) for labeling sequences where alignment between input and output is not provided. These methods provide improvements over traditional sequence models, including hidden Markov models (HMMs).

Despite the innovations in sequence modeling, the RNN-based approaches in this paper have limitations when compared to more recent neural architectures, particularly in handling complex, unstructured data. Modern models like Transformers and advanced variations of Siamese LSTM outperform traditional RNNs and LSTMs by leveraging self-attention mechanisms, allowing for parallelization and more effective long-range dependency capturing. Furthermore, these newer models excel in text similarity tasks by better addressing issues such as vanishing gradients and computational inefficiencies, areas where RNNs and LSTMs still struggle.

2.3. Unlabeled Short Text Similarity with LSTM Encoder

Yao et al. (2018) proposes an approach for measuring similarity between short texts using an LSTM-based autoencoder. The model consists of three stages: preprocessing, training, and evaluation. In the preprocessing stage, short texts are converted into word vectors using Word2Vec, followed by normalization to avoid gradient vanishing problems. The model then employs an inception module to extract multi-dimensional features, while an improved LSTM architecture learns word sequence information. Finally, cosine distance is used to compute the similarity between text vectors. The model demonstrates higher accuracy and recall compared to traditional methods such as TF-IDF and edit distance on datasets like the MSR Paraphrase Corpus and Quora Question Pairs.

Despite the improvements this model provides, it still has limitations when compared to more advanced models like the Siamese LSTM. For example, the autoencoder approach lacks the

twin-network architecture that allows the Siamese LSTM to directly compare two sequences in parallel, making the latter more effective for tasks like resume-job description matching. Additionally, the reliance on cosine similarity in this model limits its ability to capture more complex non-linear relationships between texts, something that the Siamese LSTM can handle with more sophisticated distance metrics and loss functions designed for comparison tasks.

3. Problem Description

3.1.Literature Review Summary & Superiority of Our Method

The problem at hand involves matching resumes to job descriptions, a critical task in recruitment and talent acquisition. The goal is to accurately determine the relevance of a candidate's resume to a specific job posting. This challenge goes beyond simple keyword matching; it requires an in-depth understanding of the semantic meaning of the text in both the resume and the job description. Successful automation of this task can help streamline the recruitment process, reducing manual effort and improving the accuracy of candidate-job matching.

Previous approaches to resume-job matching have largely been based on rule-based systems, keyword matching, and vector space models like TF-IDF or Word2Vec. While these approaches capture basic lexical information, they fail to understand the deeper semantic relationships between words and sentences. Rule-based systems are brittle, often missing candidates whose resumes use different terminology than the job description. Models based on Word2Vec embeddings have shown some improvements in capturing context, but they still lack the ability to handle sentence structure and long-term dependencies in text. These methods do not compare resumes and job descriptions at the sequence level, which is crucial for understanding the full context of a candidate's experience and how it aligns with job requirements.

Our proposed solution utilizes a Siamese LSTM architecture, which is superior to traditional methods because it is specifically designed to compare two sequences—resumes and job descriptions—in parallel. This architecture processes each text separately through identical LSTM networks, capturing important contextual and semantic information, and then compares the outputs to determine their similarity. The Siamese LSTM architecture excels in identifying nuanced similarities and differences between two pieces of text, making it ideal for a task like resume-job description matching where understanding the context is key. Unlike traditional models that rely solely on word embeddings or simple cosine similarity, our model is trained to understand complex relationships within and between the sequences.

3.2.Our Approach

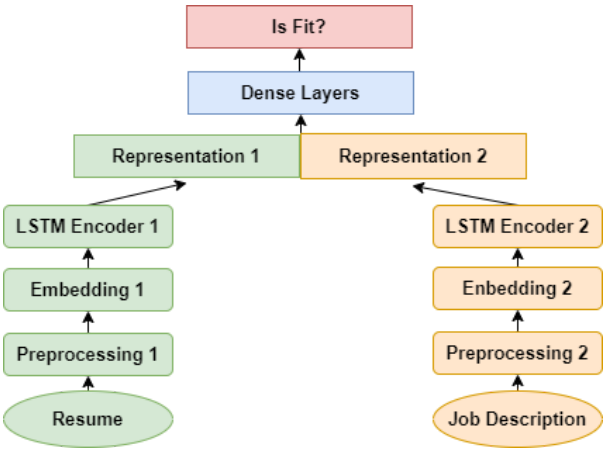

Figure 1. Model Structure

Our approach can be broken down into the following steps. In the preprocessing phase, special characters and irrelevant information are removed. Contractions are expanded into their full forms to maintain consistency in text. Both resumes and job descriptions are truncated or padded to a fixed length (e.g., 375 words) to ensure uniformity in input size. We then initially experiment with Word2Vec embeddings, but later opt to train the embeddings directly with the model, incorporating zero masking to handle variable input lengths. In the main phase, our model utilizes a Siamese LSTM architecture where two identical LSTMs (with 128 units) process the resume and job description independently. The LSTM outputs are subtracted from one another, highlighting the differences between the sequences. The resulting vector is passed through a series of dense layers (128 units with ReLU activation, followed by 64 units with ReLU) and ends with a final dense layer with a sigmoid activation for binary classification (match or no match).

In the training and test phases, we train the model over numerous epochs using a loss function optimized for binary classification. The model is tuned to maximize accuracy and minimize loss, achieving competitive results, which will be further improved by fine-tuning hyperparameters. Finally, the model is evaluated using accuracy and loss metrics. Our preliminary results show strong performance, but we will continue to refine the model based on feedback from the validation set.

4. Dataset Background and Data Preprocessing

Finding an appropriate dataset consisting of labeled pairs of job descriptions and resumes was demanding. We've ended up finding a labeled dataset on the Hugging Face website. Hugging Face is a leading AI company known for its open-source platform that provides tools and datasets for natural language processing (NLP) and machine learning. It offers a popular library called Transformers, which supports state-of-the-art models like BERT, GPT, and T5. The platform also includes the datasets library, making it easier for developers and researchers to access pre-processed datasets.

The webpage of the dataset:
https://huggingface.co/datasets/cnamuangtoun/resume-job-description-fit/tree/main

The Resume-Job-Description Fit dataset consists of two main files: a training set and a test set. During the data preprocessing in MS-Excel, we've excluded one class of 'Potential Fit' in both datasets to get a simpler solution like a binary classification. In addition, we've aliased the label column as 'is_fit' for a clearer understanding. We also changed classes in the 'is_fit' column as numerical variables. (No Fit:0, Fit:1).

	Training Set	Test Set
Number of Samples	4,685	1,315
Columns	resume, job_description, is_fit	resume, job_description, is_fit

Table 1. Description of the dataset

Before building a model, the data preprocessing steps within the ipynb file involve several stages aimed at cleaning and preparing the text data (resumes and job descriptions) for input into a deep learning model. We've removed special characters and stopwords. We then expanded contractions in the input text, such as changing "won't" to "will not". In the next step, we implemented Word Tokenization function to tokenize the input data into words. Finally, we made word vector matrix, applied sequence padding, and converted a list of words back into a single string.

5. Comparable models

5.1. Siamese LSTM Model

The Siamese LSTM model is designed to compare two sequences—resumes and job descriptions—to predict whether the resume is a good match for the job description. The model uses two LSTM networks with shared architecture to process the two inputs, then combines the outputs by taking the difference (via subtraction) and passing this result through a dense network to perform binary classification. Additionally, we applied the learning rate scheduler as a form of function: initial learning rate is 0.001, but after every 3 epochs, the learning rate is halved. This helps the model converge quickly in early stages and fine-tune during later epochs.

In the main part, two input layers are defined, one for resume, another one for job description. Then, two separate LSTM layers are applied, one for each input. The LSTM layers process the sequences and capture dependencies between words over time. The architecture of the LSTM layers is identical, which is a key characteristic of the Siamese model. (units=128, dropout=0.45, recurrent dropout=0.1). After that, the two LSTM layers generate hidden representations of the resume and job description. These representations are **subtracted** from one another. Subtracting allows the model to learn the relationship (or difference) between the two inputs. This subtraction layer is what defines the Siamese architecture: it compares the learned representations of both inputs to model the degree of similarity between them.

After subtraction, the result is passed through a dense layer with 64 units and a ReLU activation function to capture non-linear relationships in the merged representation. The final layer (Dense(1)) is used to make the prediction. It uses a sigmoid activation function for binary classification, where the output will be valued between 0 and 1, indicating the likelihood of a match between the resume and the job description. The model is compiled using the Adam optimizer with a learning rate of 0.0001 and a gradient clipping value of 1.0 to prevent exploding gradients. The loss function is binary cross-entropy, which is commonly used for binary classification tasks. The model tracks accuracy as the performance metric. Finally, the early stopping was applied to prevent overfitting by stopping training when the validation loss stops improving.

The below is the Siamese LSTM model's performance:

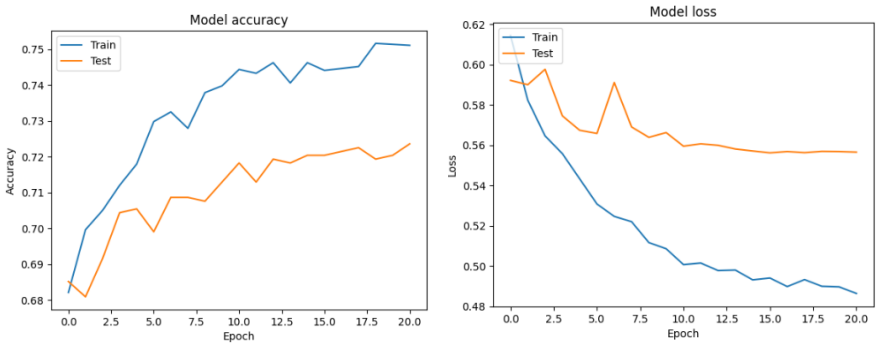

Figure 2. Model accuracy & loss graphs (Siamese LSTM)

The graph shows how the accuracy of the model changes over the 20 epochs for both the training set (blue line) and the test/validation set (orange line). The training accuracy steadily increases over time, starting from approximately 0.68 and reaching 0.75 at the end of the 20th epoch. This indicates that the model is learning the patterns in the training data well. The validation accuracy follows a similar trend, but after an initial increase, it plateaus and fluctuates around 0.71 from epoch 10 onward.

The fact that the training accuracy continues to rise while the validation accuracy plateaus suggests the model may be starting to overfit to the training data around the 10th epoch. Overfitting occurs when the model becomes too tailored to the training data and does not generalize well to unseen (validation/test) data.

5.2. Simple LSTM Model

This model uses both inputs of resumes and job descriptions, but it keeps itself much simpler than a Siamese network. The overall approaches of this model are almost the same with a Siamese LSTM model, but it **concatenates** representation of inputs instead of performing subtraction. it does not explicitly model the relationship between the two inputs using a subtraction or other advanced comparison mechanism. Instead, it just concatenates the learned representations from each input and feeds them to a dense layer for classification. The performance of this model should be inferior to the Siamese LSTM model because it does not explicitly compare the two inputs in a sophisticated manner.

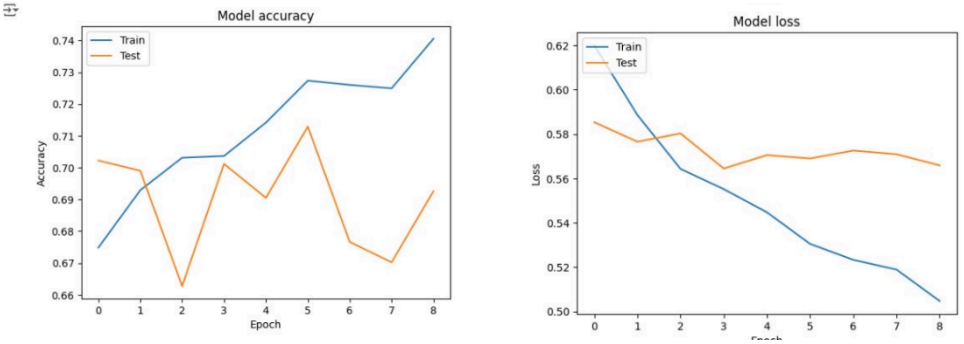

Figure 3. Model accuracy & loss graphs (Simple LSTM)

6. Siamese Bidirectional LSTM Model

In the third model, we replaced a standard LSTM with a Bidirectional LSTM as well as the

increase of units up to 256. A Bidirectional LSTM processes the input sequence in both forward and backward directions. This allows the model to capture information from both past (previous words) and future (next words) contexts in the sequence. This can significantly improve performance in sequence-processing tasks like text comparison, where understanding the full context of a sentence is important. In addition, we have doubled the number of units in the LSTM layers from 128 to 256. Increasing the number of units allows the model to capture more complex patterns and dependencies in the data. This can lead to better performance, especially when working with larger datasets. Other changes like L2 regularization and learning rate were applied to reduce overfitting or to help the model converge faster.

Below is the Siamese Bidirectional LSTM model's performance:

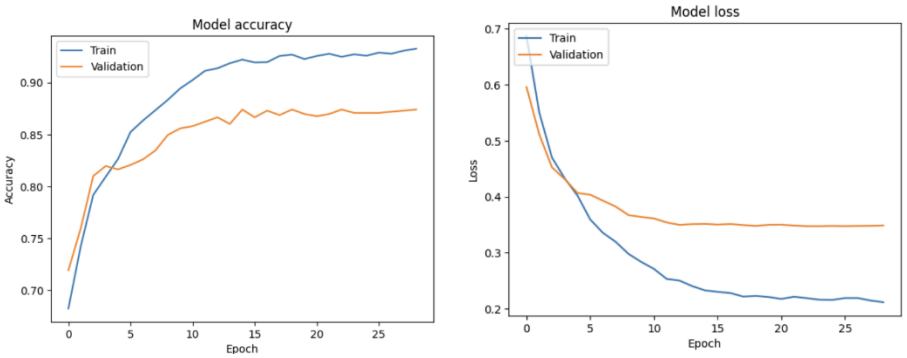

Figure 4. Model accuracy & loss graphs (Siamese Bidirectional LSTM)

7. Model Performance Comparison

	Validation Accuracy	Validation Loss
Siamese LSTM	0.71	0.55
Simple LSTM	0.69	0.56
Siamese Bidirectional LSTM	0.87	0.35

Table 2. Model Performance Comparison

The performance of the Simple LSTM Model is the worst of all, the validation accuracy kept fluctuating through every epoch. Siamese LSTM and Siamese Bidirectional LSTM models both have increasing validation accuracy and have similar patterns. However, the Siamese Bidirectional LSTM model has a higher validation accuracy, indicating it's a better model.

Hence, we applied the Siamese Bidirectional LSTM model to our test data, which is in an independent CSV file. The test accuracy did not perform well, with a rate of 0.5452. This may result from several reasons, and the test and training data might have different characteristics.

8. Conclusion, Discussion and Future Work

In this study, we developed simple LSTM, Siamese LSTM, and Siamese Bidirectional LSTM model-based architecture to analyze the matching between resumes and job descriptions. After comparing the three models' outputs, the Siamese Bidirectional LSTM model provided the highest validation accuracy, which indicated that this approach allows for more accurate matching by understanding the deeper relationships between candidate qualifications and job requirements, outperforming other models and providing a significant step forward in automating the recruitment

process.

By focusing on the sequence-level comparison, our Siamese Bidirectional LSTM model's ability to differentiate between similar but distinct job descriptions and resumes through learned semantic relationships proved particularly effective. Additionally, our preprocessing steps, such as text normalization and padding, ensured that our model handled varying input lengths efficiently.

However, the model also encountered some limitations. While it performed well in comparing resumes and job descriptions, there is a risk of overfitting. Another limitation is that our test model only has 58.48%. Furthermore, each of our models required at least 1.5 to 2 hours to build, even when our epochs and vector size were not as big as what the previous researchers had used, limiting our model's learning ability and accuracy.

Future works will focus on several key areas to enhance the performance and scalability of our job-matching model:

1. *Model Optimization*: As mentioned, our work has many limitations. Improving the fine-tuning of hyperparameters and increasing model complexity should be our priorities. Finding a way to reduce the running time is even more important, as we'll be able to adjust the model more often.

2. *Data Augmentation and Expansion*: One potential reason for poor testing accuracy may be that the training and testing datasets have different industry compositions. A better way to train the model may involve collecting and labeling different industries and teaching them separately.

3. *Real-Life Application*: The ultimate goal is to make our model usable for HRs or applicants and maximize its real-world impact. This may include integrating the Siamese LSTM-based system into existing HR platforms or improving the interpretability of the model's predictions by developing methods for explaining why a particular resume-job pair was matched (or not).

   An example of a real-life application:

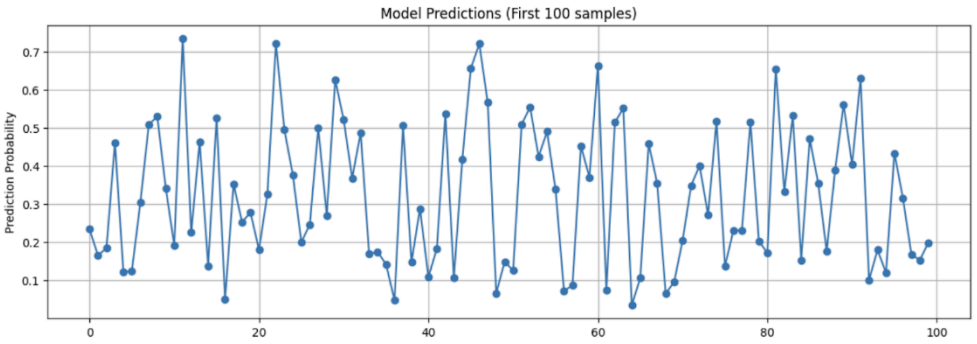

Figure 5. Prediction probability for each pair of resume and job description

	Sample Index	Predicted Label
0	1	✗
1	2	✗
2	3	✗
3	4	✗
4	5	✗
5	6	✗
6	7	✗
7	8	✓
8	9	✓
9	10	✗
10	11	✗
11	12	✓
12	13	✗
13	14	✗
14	15	✗
15	16	✓
16	17	✗
17	18	✗
18	19	✗
19	20	✗

Figure 6. Individual comparison for pairs of resumes and job descriptions

HRs can immediately decide which applicants' resumes should be further reviewed after analyzing the model. They may see which candidates are relatively higher fits or simply see which candidates fulfill the minimum requirements.

References

[1] Cohen, E. (2017). *How to Predict Quora Question Pairs Using Siamese Manhattan LSTM*. MLReview Blog. https://blog.mlreview.com/implementing-malstm-on-kaggles-quora-question-pairs-competition-8b31b0b16a07.

[2] Graves, A. (2012). Supervised Sequence Labelling with Recurrent Neural Networks. Springer.

[3] Lavi, D., Medentsiy, V., & Graus, D. (2021). conSultantBERT: Fine-tuned Siamese Sentence-BERT for Matching Jobs and Job Seekers. In RecSys in HR 2021, Amsterdam, the Netherlands, October 1. https://doi.org/10.20944/preprints202403.0111.v1:contentReference[oaicite:1]{index=1}

[4] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. arXiv preprint arXiv:1310.4546v1.

[5] Mueller, J., & Thyagarajan, A. (2016). Siamese Recurrent Architectures for Learning Sentence Similarity. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16), pp. 2786-2792.

[6] Qostal, A., Moumen, A., & Lakhrissi, Y. (2024). Resumes Classification Using Neural Network Approaches Combined with BERT and Gensim: CVs of Moroccan Engineering Students. Preprints.org. https://doi.org/10.20944/preprints202403.0111.v1:contentReference[oaicite:2]{index=2}

[7] Rezaeipourfarsangi, S., & Milios, E. E. (2023). AI-powered Resume-Job matching: A document ranking approach using deep neural networks. In ACM Symposium on Document Engineering 2023 (DocEng '23), Limerick, Ireland, August 22-25. https://doi.org/10.1145/3573128.3609347:contentReference[oaicite:0]{index=0}

[8] Yao, L., Pan, Z., & Ning, H. (2018). Unlabeled Short Text Similarity With LSTM Encoder. IEEE Access, 7, 3430-3438. https://doi.org/10.1109/ACCESS.2018.2885698.