

Query-Based Retrieval Using Universal Sentence Encoder

Godavarthi Deepthi*, A. Mary Sowjanya

Department of Computer Science and Systems Engineering, Andhra University College of Engineering(A), Visakhapatnam, AP 530003, India

Corresponding Author Email: deepthig123@yahoo.com



<https://doi.org/10.18280/ria.350404>

ABSTRACT

Received: 8 April 2021

Accepted: 5 July 2021

Keywords:

word embeddings, sentence embeddings, infersent model, universal sentence encoder, deep averaging network, dependency parse tree.

In Natural language processing, various tasks can be implemented with the features provided by word embeddings. But for obtaining embeddings for larger chunks like sentences, the efforts applied through word embeddings will not be sufficient. To resolve such issues sentence embeddings can be used. In sentence embeddings, complete sentences along with their semantic information are represented as vectors so that the machine finds it easy to understand the context. In this paper, we propose a Question Answering System (QAS) based on sentence embeddings. Our goal is to obtain the text from the provided context for a user-query by extracting the sentence in which the correct answer is present. Traditionally, infersent models have been used on SQUAD for building QAS. In recent times, Universal Sentence Encoder with USE_{CNN} and USE_{Trans} have been developed. In this paper, we have used another variant of the Universal sentence encoder, i.e. Deep averaging network in order to obtain pre-trained sentence embeddings. The results on the SQUAD-2.0 dataset indicate our approach (USE with DAN) performs well compared to Facebook's infersent embedding.

1. INTRODUCTION

Several tasks in natural language processing and computer vision can be implemented with the features provided by word embeddings. In word embedding, each word is represented using a vector. But the main problem here is to find how the relationships are captured among several words in one vector. One hot encoding is the simplest method in which the words in sequence are encoded though this is good for simple text processing tasks, it is not efficient for complex tasks like identifying common words.

Suppose our query is: Top movie stars in Tollywood and we wish to see results corresponding to movie actors, stars in Tollywood, and top. Instead, if we get results as best movie actors in Tollywood, it means the system failed to identify word similarity between 'top' and 'best' or between 'stars' and 'actors'. Hence the need for word embeddings arises where syntax and semantics of the word can be identified for representing a vector. Word2Vec, FastText, GloVe, and Elmo are the popular word embedding methods. If the text is too large, then it becomes very difficult to extract information from word embeddings. But obtaining embeddings of larger text chunks such as sentences were not successful using ordinary word embeddings. Suppose we have another sentence as 'I hate cluttered places' and after some sentences, we have another sentence 'Even though New York is the over-embellished city I like it. The Machine finds it difficult to identify the difference between 'cluttered places' and 'over-embellished city'. To resolve such issues sentence embeddings can be used. Complete sentences along with their semantic information are represented as vectors so that the machine finds it easy to understand the context, thus retaining good properties through feature inheritance from word embeddings.

Unsupervised methods such as SkipThought or Fast Sent in sentence embedding have not been satisfactory. The Facebook research team employed supervised methods to obtain sentence embeddings in which using sentence encoding architecture sentence vectors were generated first. The input considered here is word vectors. Next, a classifier was used where the input provided is encoded sentences, and sentence vectors training was performed. New research proved that better results could be obtained by using the sentence embeddings obtained from models trained on natural language inference classifier. There are distinct architectures to encode sentences such as LSTM and GRU, BiLSTM with mean/max pooling, Self-attentive network, Hierarchical ConvNet. From 2018, there has been a progressive development towards multi-task learning and supervised learning schemes with various interesting features. So the best way to create sentence embeddings is to create embedding of a single word followed by calculation of whole sentence embedding based on individual word embeddings. The current state of the art algorithm for sentence embedding creation is sentence-BERT. In this paper, we have used face book sentence embeddings for building question answering systems. We have proposed a system for answering questions from the context using two techniques namely Infersent embeddings (GLoVe, fastText) and universal sentence encoder. Our results indicate that Universal Sentence Encoder performs better than infersent embeddings.

2. RELATED WORK

Chinese Intelligent question answering system in the closed domain proposed by Cai et al. [1] was developed on the basis

of CNN-BiLSTM, coattention, and attention mechanisms. The framework which extracts patterns in query graph from the knowledge graph proposed by Jayaswal and Dixit [2] has been used for structured queries construction in RDF question answering task. A hybrid query expansion approach based on word embeddings and other lexical resources was presented by Esposito et al. [3]. A study explaining how deep learning models can be used for the classification of a question in the Turkish language was made, in which the structure of a word is obtained by appending suffixes to root [4]. Jung and Kim [5] demonstrated a method for SPARQL queries generation from natural language questions present in Korean. Song et al. [6] proposed text matching models like triple CNN and two attention based triple CNN models for improving IR based QA system for e-commerce-AliMe. The usage of word embeddings that captures semantic information from contexts for question vectorization was explained by Othman et al. [7]. QA system for evaluated medical questions was described by Abacha and Demner-Fushman [8] which recognizes question entailment. To solve question answering problems related to the medical field supervised sentence embedding framework was developed by Hao et al. [9]. Yang et al. [10] used SQUAD dataset that resolved the problems present in earlier question answering datasets. A question answering system based on machine learning was developed by Dimitriadis and Tsoumakas [11] which exploits neural networks using word embeddings. The advantage of word embeddings and the methods required for creation was explained and implemented by a convolutional neural network that was trained on pre-trained word vectors [12]. A set of paragraphs in the SQUAD dataset was used as a knowledge base for generating responses to questions without having knowledge about the paragraphs to which they belong [13]. Conneau et al. [14] obtained the knowledge of Sentences universal representation which is also referred to as a Universal Sentence Encoder model which is trained on a huge dataset and is shifted to remaining tasks. Yang et al. [15] introduced a multilingual model based on CNN and transformer architecture, target performance for the tasks that require models for capturing multilingual semantic similarity.

3. MATERIALS AND METHODS

3.1 Dataset

Stanford Question Answering Dataset [SQuAD] is a reading comprehension dataset that consists of queries posted on articles in Wikipedia by crowd workers in which for every query answer will be a piece of text from reading comprehension. This dataset is larger compared to earlier comprehension datasets.

3.2 Sentence embedding techniques

We have used two sentence embedding techniques for retrieving the answer from the provided context for a query. Here the answer for every query is present in the context itself hence it is a closed dataset. So we tried to obtain a sentence in which the correct answer was present.

3.2.1 Infersent

Stanford Natural Language Inference data was used by the team for model training for Natural Language Inference problem. Finding semantic relationships between the first

sentence(premise) and a second sentence(hypothesis) is the main target of NLI. It performs well for various tasks. Entailment, contradiction, neutral are the categories here.

Entailment is the relationship between sentences. In semantic relationships, understanding sentences are important. NLI helps in finding the best sentence embeddings for various problems in NLP. We consider sentences pair for encoding so that required sentence embeddings will be generated. Later relations between embeddings can be extracted using concatenation, element-wise product, and absolute element-wise difference.

For pre-trained word embeddings, Infersent uses GLoVe vectors, and the latest version Infersent2 uses fastText. Infersent can be used in two ways. Either, by using a pre-trained embedding layer or by building our own Infersent. We have used pre-trained embeddings in this work. After the models are loaded, the sentence can be encoded to vectors. Therefore, we have also used the two available training sentence embedding models Infersent with GLoVe and Infersent with fastText for semantic representation of sentences which was trained on natural language inference data. The context was broken down into multiple sentences using TextBlob. We have used two infersent models to obtain the vector representation for every sentence. Then cosine similarity was used for every sentence - question pair for feature creation. We have already created a vocabulary from training data and used it for infersent model training. So, we may have any number of words in the sentence. Once the training of the model is completed, a sentence is provided as input to the encoder function and the output obtained is a 4096-dimensional vector. These embeddings are used to find the sentence similarity between 2 sentences.

3.2.2 Universal sentence encoder

This embedding is used where the text is encoded into high dimensional vectors which are used for various tasks such as text classification, clustering, semantic similarity, etc. The pre-trained encoder is available in Tensorflow-hub. Transformer encoder and Deep Averaging Network are the 2 variations in USE. The first one has more accuracy but computationally it is very expensive. The second one is computationally less intensive but is less accurate. Variable-length English text when given as input gives a 512-dimensional vector as output. This model is now trained with a deep averaging network encoder. Both these models can take a sentence or word as input and produce embeddings. They work as follows:

1. Sentence tokenization is performed after conversion into lower case.

2. The sentence is then converted into a 512-dimensional encoder based on encoder type. Transformer encoder uses self-attention mechanism whereas Deep averaging network performs unigram and bigram embedding computation which is passed to a deep neural network for obtaining 512 dimensional final embeddings for sentence.

3. We can use these embeddings for unsupervised and supervised tasks such as SkipThoughts, NLI, etc. Then we can reuse the trained model for new sentence embedding of 512 dimensions generation.

To use this embedding, we installed Tensorflow and Tensorflow hub. We have loaded the model from TFhub. The context taken in Figure 1 was broken down into multiple sentences using TextBlob as shown in Figure 2.

After obtaining the vector representation for every sentence, cosine similarity was measured for every sentence - question pair for feature creation represented in Figure 3.

predicted[" context "][20584]

'In addition to Somali , Arabic , which is also an Afro-Asiatic tongue , is an official national language in both Somalia and Djibouti. Many Somalis speak it due to centuries-old ties with the Arab world, the far-reaching influence of the Arabic media, and religious education. Somalia and Djibouti are also both members of the Arab League.'

[' In addition to Somali , Arabic , which is also an Afro-Asiatic tongue , is an official national language in both Somalia and Djibouti . '

' Many Somalis speak it due to centuries-old ties with the Arab world, the far-reaching influence of the Arabic media, and religious education. '

' Somalia and Djibouti are also both members of the Arab League. ']

Figure 1. Paragraph

context	question	id	answer_start	text	sentences	target	sent_emb	quest_emb	cosine_sim
Beyoncé Giselle Knowles-Carter (/bi:ˈjɒnsez/ b...	When did Beyoncé start becoming popular?	56be85543aeaaa14008c9063	269	in the late 1990s	[Beyoncé Giselle Knowles-Carter (/bi:ˈjɒnsez/ ...	1	[[0.06352395, 0.17397285, 0.10705582, 0.073170...	[[0.11226777, 0.054507274, 0.107090585, 0.0954...	[0.46874332427978516, 0.44890081882476807, 0.4...
Beyoncé Giselle Knowles-Carter (/bi:ˈjɒnsez/ b...	What areas did Beyoncé compete in when she was...	56be85543aeaaa14008c9065	207	singing and dancing	[Beyoncé Giselle Knowles-Carter (/bi:ˈjɒnsez/ ...	1	[[0.06352395, 0.17397285, 0.10705582, 0.073170...	[[0.12392643, 0.08186243, 0.05142356, -0.01727...	[0.38697999715805054, 0.3235163688659668, 0.36...
Beyoncé Giselle Knowles-Carter (/bi:ˈjɒnsez/ b...	When did Beyoncé leave Destiny's Child and bec...	56be85543aeaaa14008c9066	526	2003	[Beyoncé Giselle Knowles-Carter (/bi:ˈjɒnsez/ ...	3	[[0.06352395, 0.17397285, 0.10705582, 0.073170...	[[0.04034418, 0.12923701, 0.03330859, 0.022752...	[0.2884094715118408, 0.26048868894577026, 0.33...
Beyoncé Giselle Knowles-Carter (/bi:ˈjɒnsez/ b...	In what city and state did Beyoncé grow up?	56bf6b0f3aeaaa14008c9601	166	Houston, Texas	[Beyoncé Giselle Knowles-Carter (/bi:ˈjɒnsez/ ...	1	[[0.06352395, 0.17397285, 0.10705582, 0.073170...	[[0.082195975, 0.0778922, 0.016079443, 0.04441...	[0.42351508140563965, 0.3781578540802002, 0.41...

Figure 2. paragraph split into sentences using TextBlob

Figure 3. Measured cosine similarity

	column_cos_0	column_cos_1	column_cos_2	column_cos_3	column_cos_4	column_cos_5	column_cos_6	column_cos_7	column_cos_8	column_cos_9	target
0	0.468743	0.448901	0.457972	0.450240	1.0	1.0	1.0	1.0	1.0	1.0	1
1	0.386980	0.323516	0.361124	0.354981	1.0	1.0	1.0	1.0	1.0	1.0	1
2	0.288409	0.260489	0.330082	0.282117	1.0	1.0	1.0	1.0	1.0	1.0	3

Figure 4. Missing values filled with 1

Once the training of the model is completed, a sentence is provided as input to the model and the output obtained is a 512-dimensional vector. These embeddings are used to find the sentence similarity between 2 sentences.

Two methods have been used for the above two sentence embedding models to solve the problem:

Unsupervised Method:

Here we have not used any target variable. The sentence from the comprehension with minimum distance from a query is retrieved as output. We used cosine similarity to detect the sentence with minimum distance from the question. This method is not suitable for huge data with target labels that are provided to us. But for small data without training, it gives good results because of face book sentence embedding.

Supervised Learning Method:

In this model, the target variable is transformed from text to sentence index which has that text. Here the length of our paragraph was restricted to 10 sentences for simplicity as most of the paragraphs have sentences less than 10 or very few sentences. So I have considered only 10 labels for prediction.

Then cosine similarity was used to build a single feature for every sentence. For paragraphs having less number of sentences, we have replaced its feature value with 1 as it is the

maximum possible cosine distance so that we can have 10 sentences in total as shown in Figure 4.

Question:

"What did Beyoncé's mother own when Beyoncé was a child?"

Context:

'**Beyoncé Giselle Knowles was born in Houston, Texas, to Celestine Ann "Tina" Knowles (née Beyoncé), a hairdresser and salon owner, and Mathew Knowles, a Xerox sales manager.** Beyoncé's name is a tribute to her mother's maiden name. Beyoncé's younger sister Solange is also a singer and a former member of Destiny's Child. Mathew is African-American, while Tina is of Louisiana Creole descent (with African, Native American, French, Cajun, and distant Irish and Spanish ancestry). Through her mother, Beyoncé is a descendant of Acadian leader Joseph Broussard. She was raised in a Methodist household.'

Text:

Salon

As the bolded sentence index is 0 target variable will be 0 here.

Corresponding to every sentence in the comprehension we will have 10 features. We have filled missing values for the remaining columns with 1 as those sentences are not present

in the comprehension.

Dependency Parsing

“**Dependency Parse Tree**” was an important feature used in our problem to provide context as this has also improved the model accuracy. Spacy tree parsing was used here for navigation through the tree. A parse tree was constructed for the question as shown in Figure 5.

Question:

In what city and state did Beyonce grow up?

```
[to_nltk_tree(sent.root).pretty_print() for sent in en_nlp (predicted.iloc[3,2]).sents]
```

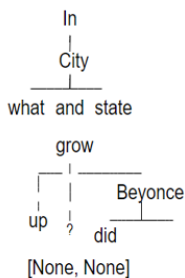


Figure 5. Parse tree for question

Sentence with an answer:

Born and raised in Houston, Texas, she performed in various singing and dancing competitions as a child, and rose to fame in the late 1990s as lead singer of R&B girl-group Destiny's Child. The answer is shown in Figure 6.

Sentences roots in the paragraph:

The sentence roots in the paragraph are shown in Figure 7.

We have matched question root to all roots or sub roots of a sentence. We will get several roots as we have various verbs in a sentence. If the question root is present in sentence roots then there is a possibility for getting an answer to that question

by that sentence. So for every sentence one feature was created and the value can be either 0 or 1. If the question root is present in the sentence roots it is represented with 1 else 0 shown as follows in Figure 8.

Before comparing question root and roots of sentences we have performed stemming. After integrating cosine distance and root match in comprehension for 10 sentences total we got 20 features while 0-9 is the range for the target variable. After training, we used logistic regression, support vector machine, k-nearest neighbors, random forest, gradient boosting, classifier techniques.

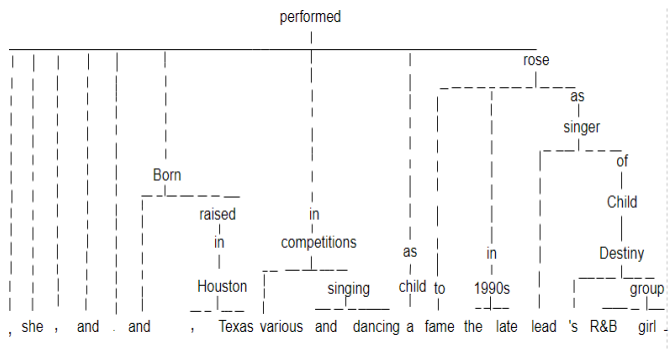


Figure 6. Parse tree for sentences containing the answer

```
for sent in doc.sents :
    roots = [st.stem(chunk.root.head.text.lower()) for chunk in sent.noun_chunks]
    print(roots)

['is', 'is', 'sing', 'songwrit', 'produc']
['in', 'houston', 'perform', 'in', 'as', 'in', 'as', 'of']
['by', 'fath', 'becam', 'of', 'of']
['saw', 'saw', 'of', 'in', 'est', 'as', 'earn', 'feat', 'singl', 'in', 'singl']
```

Figure 7. Sentence roots in paragraph

column_root_0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000	1.000000	0.000000
column_root_1	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
column_root_2	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
column_root_3	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
column_root_4	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
column_root_5	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
column_root_6	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
column_root_7	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
column_root_8	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
column_root_9	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
column_cos_0	0.468743	0.386980	0.288409	0.423515	0.375012	0.323629	0.371301	0.466113	0.472609	0.381410	0.302438	0.394674	0.291745	0.358852
column_cos_1	0.448901	0.323516	0.260489	0.378158	0.377584	0.281815	0.352307	0.407298	0.466455	0.357376	0.321990	0.405156	0.264773	0.346359
column_cos_2	0.457972	0.361124	0.330082	0.411773	0.388293	0.299792	0.322418	0.352197	0.506221	0.388683	0.349872	0.472221	0.297830	0.366812
column_cos_3	0.450240	0.354981	0.282117	0.410191	0.364631	0.325556	0.309273	0.429140	0.485644	0.390432	0.260980	0.377200	0.202214	0.371133
column_cos_4	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
column_cos_5	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
column_cos_6	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
column_cos_7	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
column_cos_8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
column_cos_9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
target	1.000000	1.000000	3.000000	1.000000	1.000000	1.000000	3.000000	2.000000	1.000000	1.000000	3.000000	3.000000	3.000000	1.000000

Figure 8. Values obtained after matching question root with sentence roots

4. EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Inferred with GLoVe

This gives an accuracy of 64% for logistic regression, 66% for random forest classifier, 68% for XGBoost, 67% for support vector machine, 58% for k-nearest neighbor, 68% for MLP classifier model for validation set. MLP classifier and XGBoost performed well compared to other models.

4.2 Inferred with FastText

This gives an accuracy of 64.7% for logistic regression, 64.8% for random forest classifier, 65.8% for XGBoost,

65.3% for support vector machine, 56.5% for k-nearest neighbor, 65.4% for MLP classifier model for validation set. MLP classifier and XGBoost performed well compared to other models.

4.3 Universal sentence encoder with deep averaging network (USE_{DAN})

This gives an accuracy of 69.5% for logistic regression, 69% for random forest classifier, 69.9% for XGBoost, 69.5% for support vector machine, 61.2% for k-nearest neighbor, 69.7% for MLP classifier model for validation set. MLP classifier and XGBoost performed well compared to other models and the results are shown in Table 1.

Table 1. Comparison of accuracy on SQUAD-2.0 results

Algorithms	Inferred with GLoVe	Inferred with FastText	Universal Sentence Encoder with DAN
XGBoost	68.5	65.8	69.9
MLP Classifier	68	65.4	69.7
Support Vector Machine	67	65.3	69.5
Logistic Regression	64.6	64.7	69.5
Random Forest classifier	66.8	64.8	69
K-nearest neighbor	58.3	56.5	61.2

5. CONCLUSIONS

Sentence embedding techniques play a key role in retrieving answers from the context for a given question compared to word embeddings. There are unsupervised methods in sentence embeddings like SkipThought, FastSent but their performance is not satisfactory hence supervised methods are applied which perform well in answer extraction. In our paper, we have used two embedding methods such as inferred with fastText and USE_{DAN} (Universal Sentence Encoder with DAN) and the latter has given good accuracy compared to other models. In the future, we plan to use other embedding techniques to improve the accuracy further using deep learning techniques.

REFERENCES

- [1] Cai, L.Q., Wei, M., Zhou, S.T., Yan, X. (2020). Intelligent question answering in restricted domains using deep learning and question pair matching. *IEEE Access*, 8: 32922-32934. <https://doi.org/10.1109/ACCESS.2020.2973728>
- [2] Jayaswal, R., Dixit, M. (2021). A framework for anomaly classification using deep transfer learning approach. *Revue d'Intelligence Artificielle*, 35(3): 255-263. <https://doi.org/10.18280/ria.350309>
- [3] Esposito, M., Damiano, E., Minutolo, A., De Pietro, G., Fujita, H. (2020). Hybrid query expansion using lexical resources and word embeddings for sentence retrieval in question answering. *Information Sciences*, 514: 88-105. <https://doi.org/10.1016/j.ins.2019.12.002>
- [4] Yilmaz, S., Toklu, S. (2020). A deep learning analysis on question classification task using Word2vec representations. *Neural Computing and Applications*, 32: 2909-2928. <https://doi.org/10.1007/s00521-020-04725-w>
- [5] Jung, H., Kim, W. (2020). Automated conversion from natural language query to SPARQL query. *Journal of Intelligent Information Systems*, 55(3): 501-520. <https://doi.org/10.1007/s10844-019-00589-2>
- [6] Song, S., Wang, C., Chen, H., Chen, H. (2020). TCNN: Triple convolutional neural network models for retrieval-based question answering system in e-commerce. *Companion Proceedings of the Web Conference 2020*, pp. 844-845. <https://doi.org/10.1145/3366424.3382684>
- [7] Othman, N., Faiz, R., Smaili, K. (2019). Enhancing question retrieval in community question answering using word embeddings. *Procedia Computer Science*, 159: 485-494. <https://doi.org/10.1016/j.procs.2019.09.203>
- [8] Abacha, A.B., Demner-Fushman, D. (2019). A question-entailment approach to question answering. *BMC Bioinformatics*, 20(1): 511. <https://doi.org/10.1186/s12859-019-3119-4>
- [9] Hao, Y., Liu, X., Wu, J., Lv, P. (2019). Exploiting sentence embedding for medical question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(1): 938-945. <https://doi.org/10.1609/aaai.v33i01.3301938>
- [10] Yang, C.J., Ishfaq, H. (2017). Question answering on squad using deep learning. <https://doi.org/10.13140/rg.2.2.20892.69767>
- [11] Dimitriadis, D., Tsoumakas, G. (2019). Word embeddings and external resources for answer processing in biomedical factoid question answering. *Journal of Biomedical Informatics*, 92: 103118. <https://doi.org/10.1016/j.jbi.2019.103118>
- [12] Mandelbaum, A., Shalev, A. (2016). Word embeddings and their use in sentence classification tasks. *arXiv preprint arXiv:1610.08229*.
- [13] Cakaloglu, T., Szegedy, C., Xu, X. (2020). Text embeddings for retrieval from a large knowledge base. *International Conference on Research Challenges in Information Science*, pp. 338-351. https://doi.org/10.1007/978-3-030-50316-1_20
- [14] Conneau, A., Kiela, D., Schwenk, H., Barrault, L., Bordes, A. (2017). Supervised learning of universal

sentence representations from natural language inference data. arXiv preprint arXiv:1705.02364.

[15] Yang, Y., Cer, D., Ahmad, A., Guo, M., Law, J., Constant, N., Abrego, G.H., Yuan, S., Tar, C., Sung,

Y.H., Strope, B., Kurzweil, R. (2019). Multilingual universal sentence encoder for semantic retrieval. arXiv preprint arXiv:1907.04307.