An Interactive Web-Interface for Visualizing the Inner Workings of the Question Answering LSTM

Ekaterina Loginova
DFKI / Saarbrücken, Germany
ekaterina.loginova@dfki.de

Günter Neumann
DFKI / Saarbrücken, Germany
neumann@dfki.de

Abstract

Deep learning models for NLP are potent but not readily interpretable. It prevents researchers from improving a model's performance efficiently and users from applying it for a task which requires a high level of trust in the system. We present a visualisation tool which aims to illuminate the inner workings of a specific LSTM model for question answering. It plots heatmaps of neurons' firings and allows a user to check the dependency between neurons and manual features. The system possesses an interactive web-interface and can be adapted to other models and domains.

1 Introduction

Deep learning models have gained popularity in the last years due to their state-of-the-art performance combined with an end-to-end pipeline. However, even though these models do not require manual feature engineering, this advantage turns into a shortcoming when it comes to the interpretation of the model. Neural networks are considered black boxes by the majority of their users. Such low interpretability leads to a low level of trust in the system's decisions. Therefore, methods for interpreting neural networks are attracting increasing interest due to their need for practical applications.

Existing visualisation methods mainly focus on computer vision tasks. It raises the issue that not all of them can be easily adapted to the NLP domain since text preprocessing operates with noticeably different units. Furthermore, while some neural architectures such as CNNs, provide relatively clear feature illustrations, this is not the case for RNNs, which are dominant for many text processing tasks. Few researchers have addressed the issue of visualising the inner workings of RNNs, especially in an interactive way.

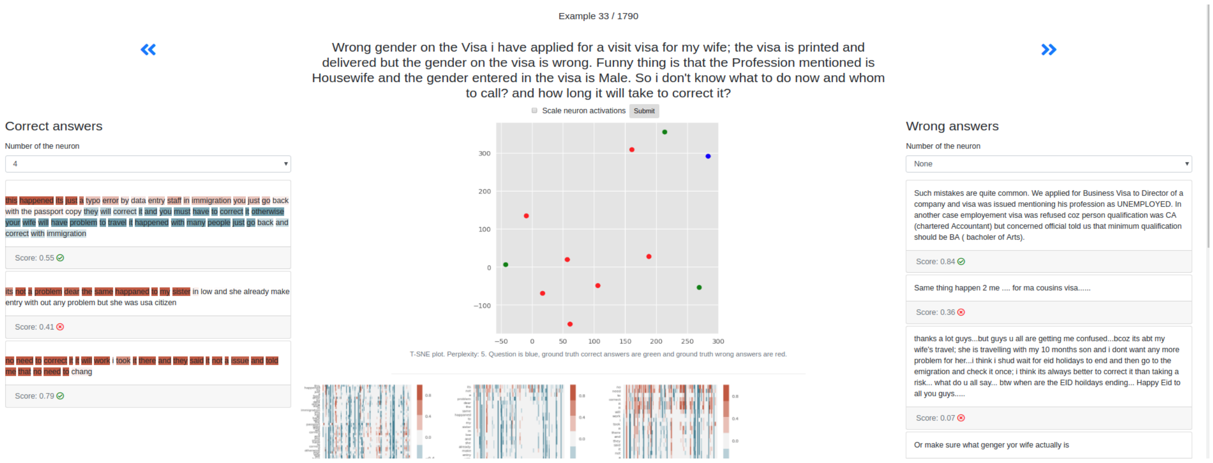
Our ultimate goal is to allow a researcher to check how interpretable the features in an RNN are automatically. The first step would be to check the dependency between a given manual feature and the features produced by the deep learning model. The next one is to develop a method to extract structural patterns from uninterpretable features. The last step would be to generate suggestions that would explain such patterns automatically. In this work, we address the first step and supply a visualisation tool for manually carrying out the second step.

This paper is a report on the visualisation system for LSTMs in the area of question answering. We present a new interactive web-interface which currently focuses on a specific system but can potentially be adapted to other models and domains. The proposed system aims to aid the development of deep learning models in NLP by providing a tool for data visualisation.

The paper is divided into three sections. The first section provides a brief overview of the related work. The system is described in the second section, and our conclusions are drawn in the final section.

2 Related work

A considerable amount of literature has been published on neural attention models (Vaswani et al., 2017; Li et al., 2015b). Most of them contain heat maps to illustrate the work of the attention mechanism. These plots provide insight into what the model sees as the more important parts of the sentence when making a prediction. For instance, self-attention might learn patterns related to syntactical properties of the text (Vaswani et al., 2017) or sentiment aspects (Li et al., 2015b). Recently researchers have also focused on illustrating the behaviour of machine translation sequence-to-



Example 33 / 1790

«

Wrong gender on the Visa i have applied for a visit visa for my wife; the visa is printed and delivered but the gender on the visa is wrong. Funny thing is that the Profession mentioned is Housewife and the gender entered in the visa is Male. So i don't know what to do now and whom to call? and how long it will take to correct it?

»

Correct answers

Number of the neuron

4 ▾

this happened its just a typo error by data entry staff in immigration you just go back with the passport copy they will correct it and you must have to correct it otherwise your wife will have problem to travel it happened with many people just go back and correct with immigration

Score: 0.55 ✓

its not a problem dear the same happaned to my sister in low and she already make entry with out any problem but she was usa citizen

Score: 0.41 ⊗

no need to correct it it will work i took it there and they said it not a issue and told me that no need to chang

Score: 0.79 ✓

□ Scale neuron activations Submit

T-SNE plot. Perplexity: 5. Question is blue, ground truth correct answers are green and ground truth wrong answers are red.

Wrong answers

Number of the neuron

None ▾

Such mistakes are quite common. We applied for Business Visa to Director of a company and visa was issued mentioning his profession as UNEMPLOYED. In another case employement visa was refused coz person qualification was CA (chartered Accountant) but concerned official told us that minimum qualification should be BA (bacholer of Arts).

Score: 0.84 ✓

Same thing happen 2 me .... for ma cousins visa......

Score: 0.36 ⊗

thanks a lot guys...but guys u all are getting me confused...bcoz its abt my wife's travel; she is travelling with my 10 months son and i dont want any more problem for her...i think i shud wait for eid holidays to end and then go to the emigration and check it once; i think its always better to correct it than taking a risk... what do u all say... btw when are the EID hoildays ending... Happy Eid to all you guys.....

Score: 0.07 ⊗

Or make sure what genger yor wife actually is

Figure 1: The second part of the interface. Left and right columns contain the ground truth correct and wrong answers respectively, highlighted according to a chosen neuron. The question is stated in the middle, a t-SNE plot is given under it. The heatmaps of neuron activations are placed below.

sequence architectures (Lee et al., 2017). Such systems include visualisations of the search tree for the beam search along with attention.

More recent research by (Karpathy et al., 2015) reveals that a small percentage of cells in LSTM learns interpretable patterns. For instance, a cell might correspond to the position in the line, indicate the depth of the nested structure or turn on inside quotes. Our system is greatly inspired by this research.

In (Li et al., 2015a) authors investigate visualisations of compositionality in NLP, basing their work on computer vision approaches. They provide t-SNE (van der Maaten and Hinton, 2008) plots for clause compositions and introduce saliency heatmaps. The latter indicates how neurons contribute to the final decision based on the first order derivatives.

Moreover, (Strobelt et al., 2016) demonstrated a system for visual analysis of hidden state dynamics in recurrent neural networks. The system, called LSTMVis, allows the researcher to check the hypothesis about local state changes against a similar pattern in the entire dataset, and align it with the textual annotations.

Finally, (Jia and Liang, 2017) show that a question answering system can be surprisingly unreliable when presented with artificial adversarial examples. They suggest that this can happen because a neural network learns heuristics, which are easy to fool. Hence, we hypothesise that a system which uses meaningful (from a human perspective) features might not only be more user-friendly but also demonstrate higher robustness.

3 System

Our system consists of two different interfaces. The first one visualises the scores and attention distribution produced by the model on answers to a user-defined question. The second one allows a user to iterate through fixed question-answer pairs to investigate the inner workings of a model. These inner workings are displayed in three separate views: General, Neuron and Correlation. In the General one, a user can observe t-SNE plots and heat maps for all neurons on the texts of the current question-answer pair from the dataset (see Figure 1 for the layout demonstration). In the Neuron view, the user can investigate the behaviour of one particular neuron further. Finally, in the Correlation part, the user can see the statistical measures for the dependency between a chosen manual feature and all neurons.

3.1 Deep learning model

Our system currently works with the Attentive (Bahdanau et al., 2014) QALSTM model. The method is essentially the same as (Tan et al., 2016) with some adjustments. Most importantly, we use two stacked shared LSTMs instead of a single one (the number of units is 96 and 64 respectively). We used Adam (Kingma and Ba, 2014) optimisation instead of SGD and cross entropy loss instead of margin ranking loss. Moreover, we modified the parameters: the embeddings are not trainable, dropout is not used, and the learning rate is set to 0.001. The main dataset used in the current stage is SemEval 2017 Task 3 Subtask A

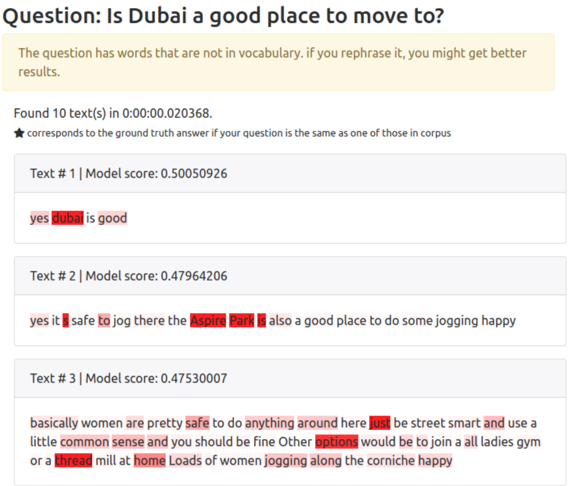
(Nakov et al., 2017). The word embeddings are word2vec (Mikolov et al., 2013) of dimension 300 pre-trained on Google News. The texts are lower-cased and tokenised using built-in Keras functions.

Generally speaking, any pre-trained Keras model involving an attention mechanism and an LSTM can be loaded into the interface. The only requirement would be to provide the layer names to retrieve softmax attention scores and LSTM weights.

3.2 Key features

A pre-trained model is loaded from a Keras checkpoint: the weights are obtained from a .h5 file and the architecture from a .json. In case of an error, a message is shown to the user with the description. The dataset with the texts of question-answer pairs is loaded in either of two ways (denoted as D-I and D-II). It can be preloaded from a custom pickle file containing a pandas data frame. This method is suitable if the candidate answers are known at the test phase. Otherwise, the dataset can be saved into an index schema of documents which is compatible with the Whoosh (Chaput, 2007) library. This will allow the system to retrieve candidate answers based on a keyword match. It should be noted that it is relatively easy to modify the code to load other datasets as long as they have the fields pool (array of incorrect answer ids), answer_ids (array of correct answer ids), question (question text) and answer (candidate answer text). Lastly, pre-trained word embeddings and the tokeniser are loaded from pickled Keras objects.

Candidate answer retrieval and scoring. To begin with, the system receives a question from the user via a text field input. The question is filtered by length: too long or empty questions are discarded. Then, it is preprocessed to exclude out-of-vocabulary (OOV) words. In addition, it is spell-checked by the Whoosh library. If the question contains words which lemmata are not in the vocabulary of the system, a warning is displayed. As mentioned at the beginning of this section, our system is separated into two interfaces depending on whether the question can be formulated by the user or it is fixed in advance. The answers in the first part are sorted by their score. In case the question was present in the dataset, and we know the correct answer, we mark its text with a star icon for a quicker performance assessment by the user. The second interface allows the user to iterate through the fixed question-answer pairs to investigate the inner workings of the model. The answers in this part are separated into columns, where the left column contains the ground truth correct answers and the right one the ground truth incorrect answers. The prediction of the model is illustrated with an icon (cross mark for predicting the answer as incorrect, checkmark for the correct), and the original score is also displayed. This leads to a clearer picture of the mistakes the model makes, which will hopefully help researchers in eliminating possible sources of errors.


Question: Is Dubai a good place to move to?

The question has words that are not in vocabulary. if you rephrase it, you might get better results.

Found 10 text(s) in 0:00:00.020368.

★ corresponds to the ground truth answer if your question is the same as one of those in corpus

Text # 1 | Model score: 0.50050926

yes dubai is good

Text # 2 | Model score: 0.47964206

yes it s safe to jog there the Aspire Park is also a good place to do some jogging happy

Text # 3 | Model score: 0.47530007

basically women are pretty safe to do anything around here just be street smart and use a little common sense and you should be fine Other options would be to join a all ladies gym or a thread mill at home Loads of women jogging along the corniche happy

Figure 2: The heatmap of attention scores.

In D-II setting, after that, Whoosh's MultifieldParser processes the question and attempts to retrieve candidate answers. If no candidates are found at this step, it extends the search by also checking the answer texts corresponding to the questions in the index. In other words, it checks not only the question-question similarity but also answer-question. If still no results are retrieved, the user receives an appropriate error message with a prompt to reformulate his question. If candidate answers were found, they are preprocessed. At this step, custom modifications, such as removing punctuation, can be added. In D-I setting, the candidate answers are provided and thus retrieved directly.

After that, the pre-trained deep learning model is applied to the texts of the question and the candidate answers to receive their scores. In our case, those are cosine similarities between LSTMs embeddings of the question and the candidate answer.

Attention visualisation. The next key feature of our system is the visualisation of the attention mechanism. In this part, we follow a traditional approach of using a heat map on the text. Words are highlighted in red based on their attention scores: the higher the score, the more intense the colour. The attention scores used in deciding the opacity of the colour are scaled to make them more visible. A user can still see the exact score of the word in a tooltip by hovering over it. Optionally, the user can adjust the attention threshold A to only highlight words with an attention score higher than A. Figure 2 shows an example of the attention visualisation.

t-SNE. Another part of the visualisations is specific to the model we currently use. The objective of this model is to learn a mapping to a new embedding space where the question has a smaller cosine distance to its correct answers than to the incorrect ones. In order to investigate the resulting space, we plot a t-SNE projection of the LSTM output embeddings on each QA-pair. The red dots correspond to the wrong answers, green - the correct ones and blue - the question. The perplexity parameter, which is known to affect the meaningfulness of t-SNE plots greatly, can be adjusted by the user with a corresponding text field input. The default value is 5.

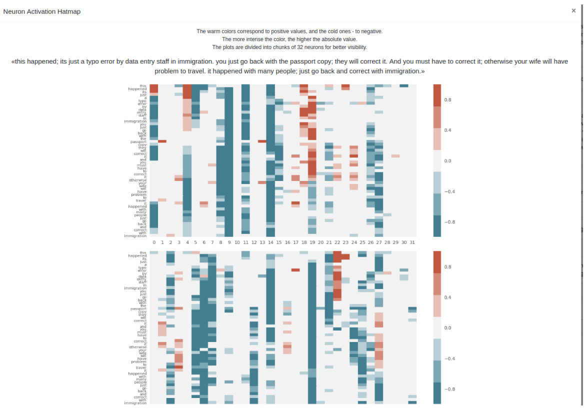
Weights visualisation. Besides visualising attention, we develop an idea proposed by (Karpathy et al., 2015), as we suggest that the neuron behaviour in a question answering system might differ across the answers categories. The illustration is generated in two modes: heatmaps and highlighting the text. Figure 1 demonstrates the highlighted text mode and figure 3 - the corresponding heatmaps. The red colour corresponds to positive values and the blue to the negative ones. The brighter the colour, the larger the absolute value of the neuron output. The current model works on a word level as opposed to character level in (Karpathy et al., 2015). A researcher is encouraged to manually analyse whether the neurons align with easily interpretable patterns in the text. If a particular neuron is of interest, the user may see heatmaps in detail by navigating to the Neuron view. There the heatmaps spanning only this particular neuron on a subset of texts are plotted. Figure 4 illustrates this mode. The user can generate more heatmaps on random texts to check for the consistency of a pattern. It should be noted that for large systems or long texts the plots can become difficult to analyse. We attempt to alleviate this by splitting the heatmaps into chunks by ten words and 32 neurons.

Correlation coefficients. We hypothesise that there is a dependency between a manual feature and a neuron. However, manually checking the heatmaps over a corpus for each neuron is time-consuming. Hence, we provide the researchers with additional information that might indicate promising pairs of features and neurons. The additional information consists of three statistical measures: the Pearson and the Spearman rank correlation coefficients and the Mutual Information score. As nature (discrete or continuous) of a user-defined manual feature is not known in advance, we provide all three scores by default. If the value of the coefficient is higher than a threshold of T, the text of the indicator will be highlighted in green. If the value is lower than $-T$, it will be highlighted in red. The user can adjust this threshold value with a corresponding text field input. The default value of T is 0.5. The user can input their own features. The format is a nested Python list of the features values for words in texts. Alternatively, the user can choose from one of the suggested features. The suggested features include the following: is the token a stop-word? Does the token consist of alphabetic character? The length of the token in characters? Is the token a noun/a verb/an adjective? Is the token a named entity? Is the token a question word (what, how)?

The values for these features are computed with the SpaCy (Honnibal and Johnson, 2015) library. Besides, the user can see some numerical characteristics of neuron values: maximum, minimum, median and mean values. He can also choose the number of texts to use, and whether they are sampled randomly or sequentially from the dataset. In case the user would like to check the dependency on a particular subset, he may input the exact indices for the question-answer pairs from the dataset. By default, the first ten instances in the dataset are used to speed up the computation process.

3.3 Technical implementation

The visualisation application is a client-server system with a web interface. It uses JQuery on the client side and Python on the server side. The application is built with the Flask (Ronacher, 2018)



Neuron Activation Hatmap

The warm colors correspond to positive values, and the cold ones - to negative.
The more intense the color, the higher the absolute value.
The plots are divided into chunks of 32 neurons for better visibility.

«this happened; its just a typo error by data entry staff in immigration. you just go back with the passport copy; they will correct it. And you must have to correct it; otherwise your wife will have problem to travel. it happened with many people; just go back and correct with immigration.»

Figure 3: A heatmap of all the neuron's firings on a given text.

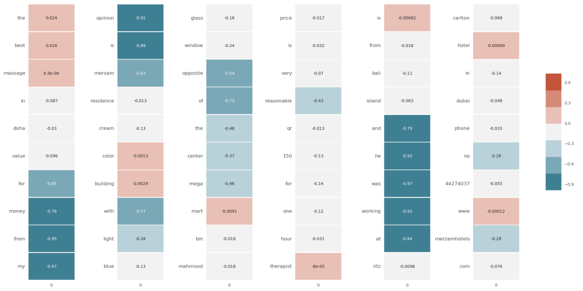


Figure 4: A heatmap of the given neuron's firings on a given text. Each cell corresponds to a word (in left to right order). The red colour corresponds to positive values and the blue to the negative ones. The more intense the colour, the larger the absolute value of the neuron output.

framework. For text preprocessing we use SpaCy (Honnibal and Johnson, 2015) and NLTK (Loper and Bird, 2002). Heatmaps and t-SNE results are plotted with the matplotlib (Hunter, 2007) library and sklearn (Pedregosa et al., 2011). The main deep learning framework is Keras (Chollet, 2015), but there is also a preliminary attempt to include PyTorch (Paszke et al., 2017) models. Statistical measures were calculated using scipy (Jones et al., 2001–).

3.4 Use cases

Regarding possible use cases, we can suggest at least five possible scenarios.

The first is to investigate possible sources of error by analysing the model's hidden vectors and attention scores.

The second promising application is to illustrate the difference between models. This can be done by loading two different models and comparing their heatmaps and attention distributions. For example, if two models with similar performance are given, the preference might go to the one with more explainable features.

The third scenario is a simplification of a model into a rule-based or machine learning approach. It can be seen as a compromise where we use a deep learning approach to extract features that might not be obvious for a human, and then transfer them to other models.

The fourth way to use the system is to try to interpret the features learnt by the network. For instance, while exploring the heatmaps, we noticed that the model seems to highlight words in the answers which are semantically relevant to the question (i.e., ”money” when a person asks for a bank recommendation). It also often reacts to the question phrases (”how much”, and so on).

Finally, we believe that interactive visualisations of LSTM hidden vectors might be enticing and helpful in education for students and beginner level practitioners.

4 Conclusion

Deep learning models for text processing are powerful, but not easily interpretable. This low interpretability leads to low trust in the systems decision and difficulties in improving its performance. Thus, there is a need for efficient visualisation tools that will illuminate the details of a neural network's decision making.

We have presented a prototype of a visualisation system for the RNN model in the question answering domain. This paper outlines the key features and structure of the system, along with the details of the technical implementation. The system displays heat maps for attention scores and firings of neurons and outputs correlation coefficients between the neurons and manual features. The approach we develop would lend itself well for use by machine comprehension researchers and developers.

Concerning possible improvements, there are five main directions. The first is to employ an automatic search for a structural pattern in neurons firing. The second is to transform neurons into transferable features which can be adapted to other models and tasks. For instance, if we see that the same type of a feature is extracted by several successful question answering systems, it might make sense to apply them in dialogue generation. The third is the need for more advanced statistical analysis. The fourth possible improvement is the incorporation of a direct comparison of two models in an interactive mode. Finally, the system can be extended to character-based models.

Future work will concentrate on extending the system to support other frameworks and visualisation techniques, such as saliency heatmaps. We also plan to include use cases for different domains, e.g. machine translation.

5 Acknowledgements

This work was partially supported by the German Federal Ministry of Education and Research (BMBF) through the project DEEPLEE (01IW17001).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Matt Chaput. 2007. Whoosh. https://bitbucket.org/mchaput/whoosh.

Franois Chollet. 2015. keras. https://github.com/fchollet/keras.

Matthew Honnibal and Mark Johnson. 2015. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378, Lisbon, Portugal. Association for Computational Linguistics.

J. D. Hunter. 2007. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *CoRR*, abs/1707.07328.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. 2001–. SciPy: Open source scientific tools for Python. [Online; accessed 29 May 2018].

Andrej Karpathy, Justin Johnson, and Li Fei-Fei. 2015. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Jaesong Lee, Joong-Hwi Shin, and Jun-Seok Kim. 2017. Interactive visualization and manipulation of attention-based neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 121–126. Association for Computational Linguistics.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2015a. Visualizing and understanding neural models in nlp. *arXiv preprint arXiv:1506.01066*.

Jiwei Li, Xinlei Chen, Eduard H. Hovy, and Dan Jurafsky. 2015b. Visualizing and understanding neural models in NLP. *CoRR*, abs/1506.01066.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, pages 63–70, Stroudsburg, PA, USA. Association for Computational Linguistics.

L.J.P. van der Maaten and G.E. Hinton. 2008. Visualizing high-dimensional data using t-sne.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. 2017. Semeval-2017 task 3: Community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 27–48.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Armin Ronacher. 2018. Flask. https://github.com/pallets/flask.

Hendrik Strobelt, Sebastian Gehrmann, Bernd Huber, Hanspeter Pfister, and Alexander M Rush. 2016. Visual analysis of hidden state dynamics in recurrent neural networks. *arXiv preprint arXiv:1606.07461*.

Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2016. Improved representation learning for question answer matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 464–473.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.