A Neural, Interactive-predictive System for Multimodal Sequence to Sequence Tasks

Álvaro Peris and **Francisco Casacuberta**
Pattern Recognition and Human Language Technology Research Center
Universitat Politcnica de Valncia, Valncia, Spain
{lvapeab, fcn}@prhlt.upv.es

Abstract

We present a demonstration of a neural interactive-predictive system for tackling multimodal sequence to sequence tasks. The system generates text predictions to different sequence to sequence tasks: machine translation, image and video captioning. These predictions are revised by a human agent, who introduces corrections in the form of characters. The system reacts to each correction, providing alternative hypotheses, compelling with the feedback provided by the user. The final objective is to reduce the human effort required during this correction process.

This system is implemented following a client–server architecture. For accessing the system, we developed a website, which communicates with the neural model, hosted in a local server. From this website, the different tasks can be tackled following the interactive-predictive framework. We open-source all the code developed for building this system. The demonstration in hosted in http://casmacat.prhlt.upv.es/interactive-seq2seq.

1 Introduction

The sequence to sequence problem involves the transduction of an input sequence $\mathbf{x}$ into an output sequence $\hat{\mathbf{y}}$ (Graves, 2012). In the last years, many tasks have been tackled under this perspective using neural networks with extraordinary results: neural machine translation (NMT; Sutskever et al., 2014; Bahdanau et al., 2015), speech recognition and translation (Chan et al., 2016; Niehues et al., 2018), image and video captioning (Xu et al., 2015; Yao et al., 2015), among others.

These systems are usually based on the statistical formalization of pattern recognition (e.g. Bishop, 2006). Following this probabilistic framework, the objective is to find most likely output sequence $\hat{\mathbf{y}}$, given an input sequence $\mathbf{x}$, according to a model $\boldsymbol{\Theta}$:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} \; \mathrm{p}(\mathbf{y} \mid \mathbf{x}; \boldsymbol{\Theta}) \qquad (1)$$

In the last years, $\boldsymbol{\Theta}$ has been frequently implemented as a deep neural network, trained in an end-to-end way. These neural systems have consistently outperformed other alternatives in the aforementioned problems. However and despite these impressive advances, the systems are not perfect, and still make errors (Koehn and Knowles, 2017).

In several scenarios, and especially in machine translation, fully-automatic systems are usually used for providing initial predictions to the input objects. These predictions are later revised by a human expert, who corrects the errors made by the system. This is known as post-editing and, in some scenarios, it increases the productivity with respect to performing the task from scratch (Alabau et al., 2016; Arenas, 2008; Hu and Cadwell, 2016).

1.1 Interactive-predictive pattern recognition

As an alternative to the static, decoupled post-editing, other strategies have been proposed, aiming to improve the productivity of the correction phase. Among them, the interactive-predictive pattern recognition (Foster et al., 1997) results particularly interesting. Under this framework, the static correction stage shifts to an iterative human-computer collaboration process.

The user interacts with the system by means of a feedback signal f. The system suggests then an alternative hypothesis $\tilde{\mathbf{y}}$, compatible with the feedback. The inclusion of the feedback into the general pattern recognition rewrites Eq. (1) introduc-

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pages 81–86
Florence, Italy, July 28 - August 2, 2019.

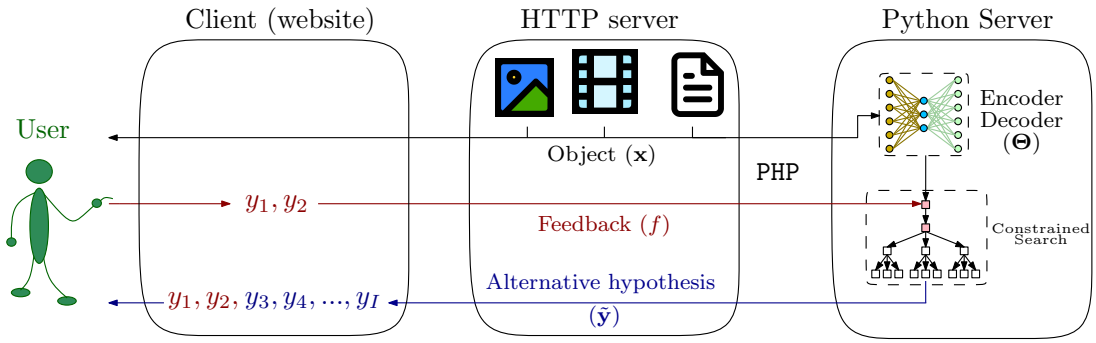

Figure 1: System architecture. The client, a website, presents the user several input objects (images, videos or texts) and a prediction. The user then introduces a feedback signal, for correcting this prediction. After being introduced, the feedback signal is sent to the server—together with the input object—for generating an alternative hypothesis, which takes into account the user corrections.

ing a restriction on the search space:

$$\tilde{\mathbf{y}} = \underset{\mathbf{y} \text{ compatible with } f}{\arg\max} \; \mathrm{p}(\mathbf{y} \mid \mathbf{x}, f; \boldsymbol{\Theta}) \qquad (2)$$

The most paradigmatic application of the interactive-predictive pattern recognition framework is machine translation. The addition of interactive protocols to foster productivity of translation environments have been studied for long time, for phrase-based models (Alabau et al., 2013, 2016; Barrachina et al., 2009; Federico et al., 2014; Green et al., 2014) and also for NMT systems (Knowles and Koehn, 2016; Peris et al., 2017; Peris and Casacuberta, 2019; Wuebker et al., 2016).

The system we are presenting in this work is an extended version of Peris and Casacuberta (2019), who presented a NMT system that accepted a prefix feedback: the user corrected the first wrong character of the sentence. Hence, the system reacted to the feedback by providing an alternative suffix. This protocol can be implemented as a constrained beam search. Moreover, the system can be retrained incrementally, as soon as a corrected sample is validated, following an online learning scenario.

We generalize this interactive-predictive NMT system to cope with alternative input modalities, namely images and videos. The system can be accessed following a client–server interface. We developed a client website, that access to our servers, in which the interactive-predictive systems are deployed. A live demo of the system can be accessed in: `http://casmacat.prhlt.upv.es/interactive-seq2seq`.

In the following sections, we describe the main architecture, features and usage of our interactive-predictive system. We also describe the frontend of our demonstration website and present an example of interactive session.

2 System description

The core of our system is a neural sequence to sequence model, developed with NMT-Keras (Peris and Casacuberta, 2018). This library is built upon Keras (Chollet et al., 2015) and works for the Theano (Theano Development Team, 2016) and Tensorflow (Abadi et al., 2016) backends. The system is deployed as a Python-based HTTP server that waits for requests. The user interactions are introduced through a (client) HTML website. The website is hosted on a Nginx server that manages the interactions using Javascript and communicates with the Python server, using the PHP curl tool. All code is open-source and publicly available[1,2].

NMT-Keras extends the (already extensive) Keras functionalities, providing a flexible, easy to use framework upon which build neural models. Among the features brought by NMT-Keras, some of them are particularly useful for sequence-to-sequence tasks: extended recurrent neural networks, with embedded attention mechanisms and conditional LSTM/GRU units (Sennrich et al., 2017), multi-head attention layers, positional encodings and position-wise feed-forward networks for building Transformer models (Vaswani et al.,

[1]Python server source code: `https://github.com/lvapeab/interactive-keras-captioning`.

[2]HTML server source code: `https://github.com/lvapeab/inmt_demo_web`.

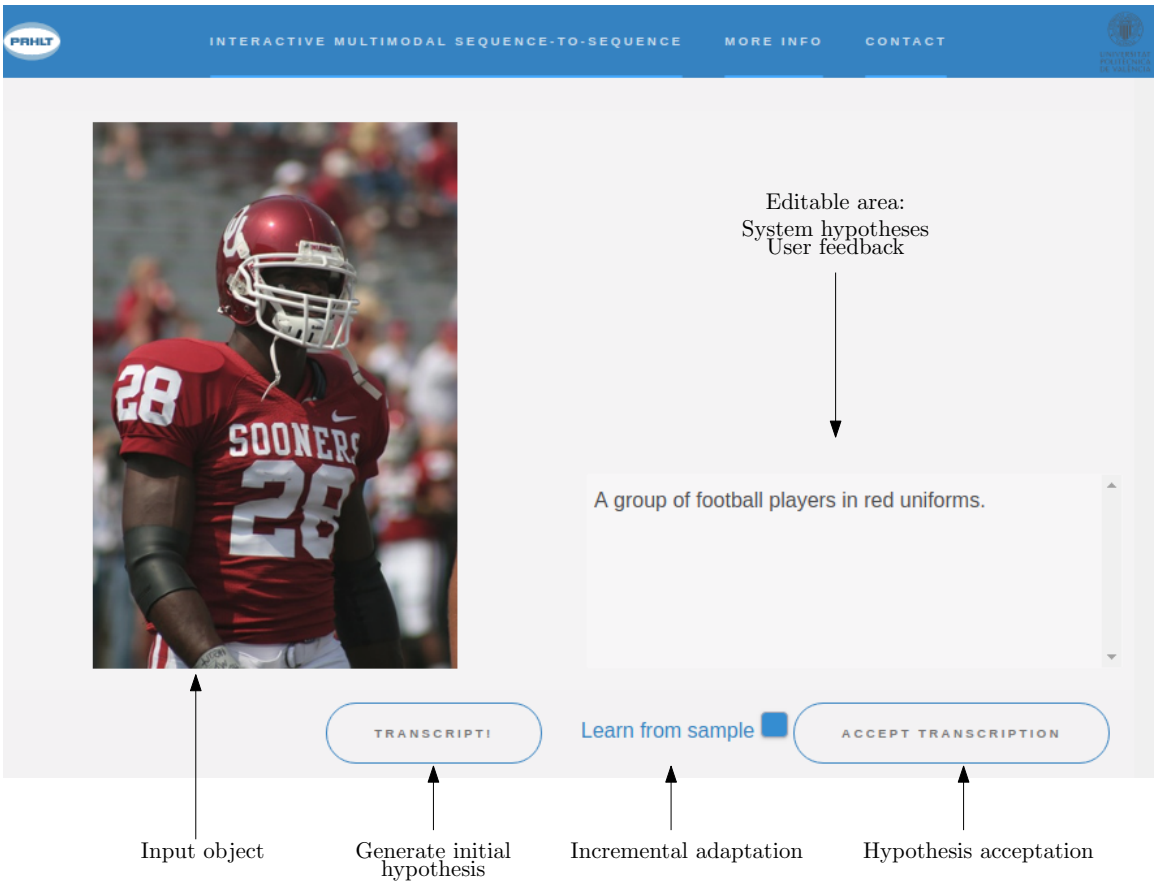

Figure 2: Frontend of the client website. As the button "Transcript!" is clicked, an initial hypothesis for the input object—in this case, an image—appears in the right area. The user then introduces corrections of this text. The system reacts to each translation, producing alternative hypotheses, always compliant with the user feedback. Once a correct caption of the image is reached, the user clicks in the "Accept translation" button, validating the hypothesis.

2017) and a modular handler for processing different data modalities, including text, images, videos or categorical labels.

Within this framework, we built our neural systems, which are leveraged via our interactive client–server application. The neural systems are deployed in a server, waiting for requests. When the client ask for a prediction, they react, generate the prediction and deliver it back to the client.

2.1 Usage of the interactive system

Our interactive-predictive system works as follows: initially, an input object is presented to the user in the client website. The user requests an automatic prediction of it. Next, the client communicates the server via PHP. The server queries the neural system, which produces an initial hypothesis applying Eq. (1). The hypothesis is then sent back to the client website.

Next, the interactive-predictive process starts: the user searches in this hypothesis the first error, and introduces a correction with the keyboard (writing one or more characters). When the user stops typing the correction, the system reacts, sending to the server a request containing the input object and the user feedback (the sequence of characters that conform the correct prefix). Then, the neural model implements Eq. (2) and produces an alternative hypothesis, such that it completes the correct prefix. This is implemented as a constrained beam search, as described in Peris et al. (2017); Peris and Casacuberta (2019). This iteration of the process is illustrated in Fig. 1.

This protocol is repeated until the user finds satisfactory the hypothesis given by the system. Then, it is validated. As soon as the sentence is validated, the system can be incrementally updated with this sample, following an online learning setup (Peris and Casacuberta, 2019). Hence, in future interactions, the system will be progressively updated, tailoring to a given domain or to the user preferences. These adaptive systems have

0	**System**	A group of football players in red uniforms.
1	**User**	*A* **f**group of football players in red uniforms.
	System	*A f*ootball player in a red uniform is holding a football.
2	**User**	*A football player in a red uniform is* **w**holding a football.
	System	*A football player in a red uniform is w*earing a football.
3	**User**	*A football player in a red uniform is wearing a* **h**football.
	System	*A football player in a red uniform is w*earing a helmet.
4	**User**	*A football player in a red uniform is wearing a helmet.*

Figure 3: Interactive-predictive session for correcting the caption generated in Fig. 2. At each iteration, the user introduces a character correction (boxed). The system modifies its hypothesis, taking into account this feedback: keeping the correct prefix (green) and generating a compatible suffix.

shown to be effective for reducing the human effort spent in the process (Karimova et al., 2018).

3 System showcase

To show the interactive-predictive protocol described in the previous sections, we developed a website which hosts a demonstration of the system. Our demonstration system handles three different problems, regarding three different data modalities: text-to-text (NMT), image-to-text (image captioning) and video-to-text (video captioning). For tackling these tasks, we use a similar model: a neural encoder–decoder, based on recurrent neural networks with attention (Bahdanau et al., 2015; Xu et al., 2015; Yao et al., 2015). Our framework has also support for Transformer-like architectures (Vaswani et al., 2017).

The NMT task regards the translation of texts from a medical domain. The system is similar to the one used by Peris and Casacuberta (2019), and was trained on the UFAL corpus (Bojar et al., 2017). The image and video captioning systems were trained on the Flickr8k (Hodosh et al., 2010) and MSVD (Chen and Dolan, 2011) datasets, respectively. The images were encoded using an Inception convolutional neural network (Szegedy et al., 2016) trained on the ILSVRC dataset (Russakovsky et al., 2015). The decoder receives the representation previous to the fully-connected work. In the case of the video captioning system, we applied a 3D convolutional neural network (Tran et al., 2015), for obtaining time-aware features.

Finally, as aforementioned in previous sections, the systems can be retrained after the validation of each sample. In our demonstration, the systems are updated via gradient descent, but using a learning rate of 0, which prevents a degradation of the model due to accidental misuse.

3.1 Example: image captioning

We show and analyze an image captioning example. The NMT and video captioning tasks are similar. Fig. 2 shows the demo website, for the image captioning task. In the left part of the screen, the input object is shown, in this case, an image. As the user clicks in the "Transcript!" button, the system generates a caption of the image, displaying it in an editable area on the right part of the screen. The user can then introduce the desired corrections to this hypothesis. As a correction is introduced, the system reacts, providing an alternative caption, but always considering the feedback given by the user.

As can be seen in Fig. 2, the caption generated by the system has some errors. Fig. 3 an shows the interactive-predictive captioning session, for obtaining a correct sample. With three interactions, the system was able to obtain a correct caption for the image.

It is particularly interesting to observe that the system correctly accounts for the singular/plural concordance of the clause *in red uniform(s)*, depending on the subject (*A football player*/*A group of football players*).

4 Conclusions and future work

We presented a demonstration of a interactive-predictive neural system for multimodal sequence to sequence tasks. We described its client–server architecture and developed a website for ease the usage of the system.

As future work, we would like to improve the frontend of our website. Inspecting the attributes of black-box neural models is a relevant research topic, and it is under active development (e.g. Zeiler and Fergus, 2014; Ancona et al., 2017). Visualizing these relevant attributes would help to understand the model predictions and behavior.

Moreover, a more sophisticated frontend would allow to implement interesting features, such as mapping the attention weights through the input sequence or the implementation of more complex interaction protocols, such as touch-based interaction (Marie and Max, 2015) or segment-based interaction (Peris et al., 2017). We intend to offer the different functionalities of the toolkit as REST services, for improving the reusability of the code. It is also planned to release the library in a Docker container in order to ease the deployment of future applications.

Acknowledgments

We acknowledge the anonymous reviewers for their helpful suggestions. The research leading to these results has received funding from the Generalitat Valenciana under grant PROMETEOII/2014/030 and from TIN2015-70924-C2-1-R. We also acknowledge NVIDIA Corporation for the donation of GPUs used in this work.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation*, volume 16, pages 265–283.

Vicent Alabau, Ragnar Bonk, Christian Buck, Michael Carl, Francisco Casacuberta, Mercedes García-Martínez, Jesús González-Rubio, Philipp Koehn, Luis A. Leiva, Bartolomé Mesa-Lao, Daniel Ortiz-Martínez, Hervé Saint-Amand, Germán Sanchis-Trilles, and Chara Tsoukala. 2013. CASMACAT: An open source workbench for advanced computer aided translation. *The Prague Bulletin of Mathematical Linguistics*, 100:101–112.

Vicent Alabau, Michael Carl, Francisco Casacuberta, Mercedes Garca-Martnez, Jess Gonzlez-Rubio, Bartolom Mesa-Lao, Daniel Ortiz-Martnez, Moritz Schaeffer, and Germn Sanchis-Trilles. 2016. *New Directions in Empirical Translation Process Research*, New Frontiers in Translation Studies, chapter Learning Advanced Post-editing.

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. 2017. Towards better understanding of gradient-based attribution methods for deep neural networks. *arXiv:1711.06104*.

Ana Guerberof Arenas. 2008. Productivity and quality in the post-editing of outputs from translation memories and machine translation. *Localisation Focus*, 7(1):11–21.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *arXiv:1409.0473*.

Sergio Barrachina, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio Lagarda, Hermann Ney, Jesús Tomás, Enrique Vidal, and Juan-Miguel Vilar. 2009. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer.

Ondej Bojar, Barry Haddow, David Mareek , Roman Sudarikov, Aleš Tamchyna, and Duan Vari. 2017. Report on building translation systems for public health domain (deliverable D1.1). Technical Report H2020-ICT-2014-1-644402, Health in my Language (HimL).

William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4960–4964.

David L Chen and William B Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 190–200.

François Chollet et al. 2015. Keras. `https://github.com/keras-team/keras`. GitHub repository.

Marcello Federico, Nicola Bertoldi, Mauro Cettolo, Matteo Negri, Marco Turchi, Marco Trombetti, Alessandro Cattelan, Antonio Farina, Domenico Lupinetti, Andrea Martines, Alberto Massidda, Holger Schwenk, Loïc Barrault, Frederic Blain, Philipp Koehn, Christian Buck, and Ulrich Germann. 2014. The matecat tool. In *Proceedings of the 25th International Conference on Computational Linguistics: System Demonstrations*, pages 129–132.

George Foster, Pierre Isabelle, and Pierre Plamondon. 1997. Target-text mediated interactive machine translation. *Machine Translation*, 12:175–194.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *arXiv:1211.3711*.

Spence Green, Jason Chuang, Jeffrey Heer, and Christopher D. Manning. 2014. Predictive translation memory: A mixed-initiative system for human language translation. In *Proceedings of the Annual Association for Computing Machinery Symposium on User Interface Software and Technology*, pages 177–187.

Micah Hodosh, Peter Young, Cyrus Rashtchian, and Julia Hockenmaier. 2010. Cross-caption coreference resolution for automatic image understanding. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 162–171.

Ke Hu and Patrick Cadwell. 2016. A comparative study of post-editing guidelines. In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation*, pages 34206–353.

Sariya Karimova, Patrick Simianer, and Stefan Riezler. 2018. A user-study on online adaptation of neural machine translation to human post-edits. *Machine Translation*, 32(4):309–324.

Rebecca Knowles and Philipp Koehn. 2016. Neural interactive translation prediction. In *Proceedings of the Association for Machine Translation in the Americas*, pages 107–120.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39.

Benjamin Marie and Aurélien Max. 2015. Touch-based pre-post-editing of machine translation output. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1040–1045.

Jan Niehues, Ngoc-Quan Pham, Thanh-Le Ha, Matthias Sperber, and Alex Waibel. 2018. Low-latency neural speech translation. In *Proceedings of the 19th Annual Conference of the International Speech Communication Association*, pages 1293–1297.

Álvaro Peris and Francisco Casacuberta. 2018. NMT-Keras: a very flexible toolkit with a focus on interactive NMT and online learning. *The Prague Bulletin of Mathematical Linguistics*, 111:113–124.

Álvaro Peris and Francisco Casacuberta. 2019. Online learning for effort reduction in interactive neural machine translation. *Computer Speech & Language*, 58:98–126.

Álvaro Peris, Miguel Domingo, and Francisco Casacuberta. 2017. Interactive neural machine translation. *Computer Speech & Language*, 45:201–220.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations at the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 27, pages 3104–3112.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826.

Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv:1605.02688*.

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Joern Wuebker, Spence Green, John DeNero, Sasa Hasan, and Minh-Thang Luong. 2016. Models and inference for prefix-constrained machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 66–75.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the International Conference on Machine Learning*, pages 2048–2057.

Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. 2015. Describing videos by exploiting temporal structure. In *Proceedings of the International Conference on Computer Vision*, pages 4507–4515.

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *Proceedings of the European Conference on Computer Vision*, pages 818–833.