Detecting Off-topic Responses to Visual Prompts

Marek Rei
The ALTA Institute
Computer Laboratory
University of Cambridge
United Kingdom
marek.rei@cl.cam.ac.uk

Abstract

Automated methods for essay scoring have made great progress in recent years, achieving accuracies very close to human annotators. However, a known weakness of such automated scorers is not taking into account the semantic relevance of the submitted text. While there is existing work on detecting answer relevance given a textual prompt, very little previous research has been done to incorporate visual writing prompts. We propose a neural architecture and several extensions for detecting off-topic responses to visual prompts and evaluate it on a dataset of texts written by language learners.

1 Introduction

Evaluating the relevance of learner essays with respect to the assigned prompt is an important part of automated writing assessment (Higgins et al., 2006; Briscoe et al., 2010). Existing systems are able to assign high-quality assessments based on grammaticality (Yannakoudakis et al., 2011; Ng et al., 2014), but are known to be vulnerable to memorised off-topic answers which can be a critical weakness in high-stakes testing. In addition, students who have limited relevant vocabulary may try to shift the topic of their answer in a more familiar direction, which most automated assessment systems are not able to capture. Solutions for detecting topical relevance can help prevent these weaknesses and provide informative feedback to the students.

While there is previous work on assessing the relevance of answers given a textual prompt (Persing and Ng, 2014; Cummins et al., 2015; Rei and Cummins, 2016), very little research has been done to incorporate visual writing prompts. In this setting, students are asked to write a short description about an image in order to assess their language skills, and we would like to automatically evaluate the semantic relevance of their answers. An intuitive method for comparing multiple modalities is to map them into a shared distributed space – semantically similar entities will get mapped to similar vector representations, regardless of the information source. Frome et al. (2013) used this principle to improve image recognition, by first training separate visual and textual components, and then mapping the images into the same space as word embeddings. Ma et al. (2015) performed information retrieval tasks with a related model based on convolutional networks. Klein et al. (2015) learned to associate word embeddings to images using Fisher vectors.

In this paper, we start with a similar architecture, based on the approach used by Kiros et al. (2014) for image caption generation, and propose modifications that make the model more suitable for discriminating between relevant and irrelevant answers. The framework uses an LSTM for text composition and a pre-trained image recognition model for extracting visual features. Both representations are mapped to the same space and a prediction is made about the relevance of the text given the image. We propose a novel gating component that decides which parts of the image should be considered for the current similarity calculation, based on first reading the input sentence. Application of dropout to word embeddings and visual features helps increase robustness on an otherwise noisy dataset and assisted in regularising the model. Finally, the standard loss function is replaced with a version of cross-entropy, encouraging the model to jointly optimise over batches. We evaluate on a dataset of short answers by language learners, written in response to visual prompts and our experiments show performance improvements for each of the model modifications.

2 Relevance Detection Model

Automated methods for scoring essays and short answers have made great progress in recent years (Yannakoudakis et al., 2011; Sakaguchi et al., 2015; Alikaniotis et al., 2016; Hussein et al., 2017), achieving accuracies very close to human annotators. However, a known weakness of such automated scorers is not taking into account the topical relevance of the submitted text. Students with limited language skills may attempt to shift the topic of the response in a more familiar direction, which automated systems would not be able to detect. In a high-stakes examination framework, this weakness could be further exploited by memorising a grammatically correct answer and presenting it in response to any prompt. Being able to detect topical relevance can help prevent such weaknesses, provide useful feedback to the students, and is also a step towards evaluating more creative aspects of learner writing. While there is existing work on detecting answer relevance given a textual prompt (Persing and Ng, 2014; Cummins et al., 2015; Rei and Cummins, 2016), only limited previous research has been done to extend this to visual prompts. Some recent work has investigated answer relevance to visual prompts as part of automated scoring systems (Somasundaran et al., 2015; King and Dickinson, 2016), but they reduced the problem to a textual similarity task by relying on hand-written reference descriptions for each image without directly incorporating visual information.

Our proposed relevance detection model takes an image and a sentence as input, and assigns a score indicating how relevant the image is to the text. Formulating this as a scoring problem instead of binary classification allows us to treat the model output as a confidence score, and the classification threshold can be selected at a later stage based on the specific application.

Kiros et al. (2014) describe a supervised method for mapping an image and a sentence into the same space, which allows them to generate similar vector representations for images that have semantically similar descriptions. We base our approach for multimodal relevance scoring on this architecture, and introduce several modifications in order to adapt it to the task of discriminating between relevant and irrelevant textual answers.

The outline of our framework can be seen in Figure 1. The input sentence is first passed through a Long Short-Term Memory (LSTM, Hochreiter and Schmidhuber (1997)) component, mapping it to a vector representation u. The visual features for the input image are extracted using a model trained for image recognition. The visual representation is then conditioned on the input sentence and mapped to a vector representation v. Both u and v are given as input to a function that predicts a confidence score for the answer being relevant to the image. In the next sections we will describe each of these components in more detail.

2.1 Text Composition

The input to the text composition component is a tokenised sentence. We first map these tokens to an embedding space, resulting in a sequence of vector representations:

$$[w_1, w_2, ..., w_N] \tag{1}$$

Next, we apply dropout (Srivastava et al., 2014) to each of the word embeddings in the sentence. Dropout is a method of regularising neural networks, shown to provide performance imrovements. Neuron activations in a layer are set to zero with probability p, preventing the model from excessively relying on the presence of specific features. The process can also be thought of as training a randomly constructed smaller network at each training iteration, resulting in a full combination model. At test time, all the values are retained, but scaled with $(1 - p)$ to compensate for the difference. While dropout is commonly applied to weights inside the network (Tai et al., 2015; Zhang et al., 2015; Kalchbrenner et al., 2015; Kim et al., 2016), there is also some recent work that deploy dropout directly on the word embeddings (Rocktäschel et al., 2016; Chen et al., 2016). The relevance scoring model needs to handle texts from different domains, including error-prone sentences from language learners, and dropout on the embeddings allows us to introduce robustness into the training process.

We use an LSTM component for processing the word embeddings, building up a sentence representation. It is similar to a traditional recurrent neural network, with specialised gating functions that allow it to dynamically decide which information to carry forward or forget. The LSTM calcu-

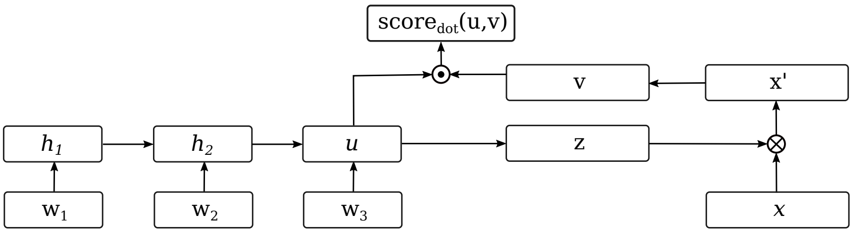

Figure 1: The outline of the relevance detection model. The input sentence and image are mapped to vector representations u and v using modality-specific functions. These vectors are then given to a relevance function which assigns a real-valued score based on their similarity.

lates a hidden representation at word n based on the current word embedding and the previous hidden representation at time step $n-1$:

$$h_n = LSTM(w_n, h_{n-1}) \tag{2}$$

The last hidden representation h_N is calculated based on all the words in the sequence, thereby allowing the model to iteratively construct a semantic representation of the whole sentence. We use this vector $u = h_N$ to represent a given input sentence in the relevance scoring model. Since word-level processing is not ideal for handling spelling errors in learner texts, future work could also investigate character-based extensions for text composition, such as those described by Rei et al. (2016) and Wieting et al. (2016).

2.2 Image Processing

In order to map images to feature vectors, a pre-trained image recognition model is combined with a supervised transformation component. We make use of the BVLC GoogLeNet image recognition model, which is based on an architecture described by Szegedy et al. (2015) and provided by the Caffe toolkit (Jia et al., 2014). The GoogLeNet is a 22-layer deep convolutional network, trained on ImageNet (Deng et al., 2009) data to detect 1,000 different image classes.

An input image is passed through the network and a probability distribution over the possible classes is produced. Instead of using the output layer, we extract the neuron activations at the second-to-last layer in the network – this takes advantage of all the visual feature processing on various levels of the network, but retains a more general distributed representation of the image compared to using the output layer. Similarly to the word embeddings in textual composition, we apply dropout with probability p directly to the image vectors – this introduces variance to the otherwise limited training data, and prevents the model from overfitting on specific features.

The previous process maps the image to a 1024-dimensional vector x, which contains useful visual information but is not optimised for the relevance scoring task. We introduce a gating component which modulates the image vector, based on the textual vector representation from the input sentence. A vector of gating weights is calculated as a nonlinear weighted transformation of the sentence vector u:

$$z = \sigma(uW_z + b_z) \tag{3}$$

where W_z is a weight matrix, b_z is a bias vector, and $\sigma()$ is the logistic activation function with values between 0 and 1. A new image representation x' is then calculated by applying these element-wise weights to the visual vector x:

$$x' = z * x \tag{4}$$

where $*$ indicates an element-wise multiplication. This architecture allows the model to first read the input sentence, determine what to look for in the corresponding image, and block out irrelevant information in the image vector. We also disconnect the backpropagation between vector u and the gating weights z – this forces the model to optimise u only for score prediction, leaving W_z and b_z to specialise on handling the gating.

Finally, we pass the image representation through a fully connected non-linear layer – this allows the model to transform the pre-trained GoogLeNet space to a representation that is specialised for relevance scoring:

$$v = tanh(x'W_x) \tag{5}$$

where W_x is a weight matrix that is optimised during training, and v is the final image vector that is used as input to the relevance scoring component.

2.3 Scoring and optimisation

Based on vector representations for the input sentence (u) and image (v) we now want to assign a score which indicates how related they are. Kiros et al. (2014) used the cosine measure as the similarity function – it measures the angle between two vectors, returning a value in the range $[-1, 1]$, and is commonly used for similarity calculations in language processing:

$$score_{cos}(u, v) = cos(u, v) = \frac{uv}{|u||v|} \tag{6}$$

The model can then be optimised to predict a high score for image-sentence pairs where the image and sentence and related, and a low score for randomly constructed pairs. The loss function is a hinge loss with a margin m; if the score difference between the positive and negative example is greater than m, then no training is required, otherwise the error is backpropagated and weights are updated accordingly:

$$\begin{aligned} Loss_{hinge} = \sum_{i \in I} \sum_{j \in J(i)} & max(-score_{cos}(u_i, v_i) \\ & +score_{cos}(u_j, v_i) + m, 0) \end{aligned} \tag{7}$$

where I is the set of related image-text pairs for training, and $J(i)$ is a set of randomly constructed pairs for entry i. When generating the negative examples, we make sure the resulting set $J(i)$ does not contain any examples with the same image as i – otherwise the model would accidentally optimise related examples towards a low score.

In this work we propose using an alternative scoring function, in order to help discriminate between the answers. We first replace the cosine similarity with a dot-product:

$$score_{dot}(u, v) = uv \tag{8}$$

Next, we create a scoring function by calculating a probability distribution over the current minibatch of examples:

$$score_{exp}(u_i, v_i) = \frac{\exp(score_{dot}(u_i, v_i))}{Z} \tag{9}$$

$$\begin{aligned} Z = & \exp(score_{dot}(u_i, v_i)) \\ & + \sum_{j \in J(i)} \exp(score_{dot}(u_j, v_i)) \end{aligned} \tag{10}$$

	images	sentences
TRAIN	29,000	145,000
DEV	1,014	5,070
TEST	1,000	5,000

Table 1: Number of images and descriptions in the Flickr30k dataset.

The model is then optimised for cross-entropy, which is equivalent to optimising the negative log-likelihood:

$$Loss_{ce} = -\sum_{i \in I} log(score_{exp}(u_i, v_i)) \tag{11}$$

The transition from cosine to dot-product is required in order to facilitate the new scoring function. In this setting, $score_{exp}(u_i, v_i)$ acts as a softmax layer, requiring the input values to be unbounded for functioning correctly, whereas cosine would restrict values to a range between -1 and 1.

The new scoring function based on softmax encourages the model to further distinguish between relevant and irrelevant images. While the hinge loss function is also optimised in minibatches, it independently optimises the relevance score of each training pair, whereas softmax connects the scores for all the pairs into a probability distribution. When this distribution is optimised using cross-entropy, it specifically focuses more on instances that incorrectly have relatively high scores compared to other pairs in the dataset. In addition, optimising towards a larger score for the known correct example also reduces the scores for all other pairs in the batch.

3 Evaluation Setup

Given an image and a text written in response to this image, the goal of the system is to assign a score and return a decision about the relevance of this text. We evaluate the framework on an experimental dataset collected by the English Profile[1], containing 543 answers written by language learners in response to visual prompts in the form of photographs. As part of the instructions, the students were able to select the image that they wanted to write about, and were then free to choose what to write. The length of the collected answers ranges from 1 to 44 sentences.

[1] http://www.englishprofile.org/

This dataset contains real-world examples for the task of visual relevance detection, and therefore also proposes a range of challenges. The answers are provided by students in various stages of learning English, which means the texts contain numerous writing errors. Spelling mistakes prevent the model from making full use of word embeddings, and previously unseen grammatical mistakes will cause trouble for the LSTM composition function. The students have also interpreted the open writing task in various different ways – while some have answered by describing the content of the image, others have instead talked about personal memories triggered by the image, or even created a short fictional story inspired by the photo. This has led to answers that vary quite a bit in writing style, vocabulary size and sentence length.

Ideally, we would like to train the model on examples where pairs of images and sentences are specifically annotated for their semantic relevance. However, since the collected dataset is not large enough for training neural networks, we make use of the Flickr30k (Young et al., 2014) dataset which contains implicitly relevant pairs of images and their corresponding descriptions. Flickr30k is an image captioning dataset, containing 31,014 images and 5 hand-written sentences describing each image. We use the same splits as Karpathy and Li (2015) for training and development; the dataset sizes are shown in Table 1. During training, the model is presented with 32 sentences and their corresponding images in each batch, making sure all the images within a batch are unique. The loss function from Section 2.3 is then minimised to maximise the predicted scores for the 32 relevant pairs, and minimise the scores for the $32 * 32 - 32 = 992$ random combinations.

Theano (Bergstra et al., 2010) was used to implement the neural network model. The texts were tokenised and lowercased, and sentences were padded with special markers for start and end positions. The vocabulary includes all words that appeared in the training set at least twice, plus an extra token for any unseen words. Words were represented with 300-dimensional embeddings and initialised with the publicly available vectors trained with CBOW (Mikolov et al., 2013). All other parameters were initialised with random values from a normal distribution with mean 0 and standard deviation 0.1.

	ACC	AP	P@50
Random	50.0	50.0	50.0
LSTM-COS	68.2	71.6	81.0
+ gating	69.6	74.6	84.4
+ cross-ent	71.1	79.0	**92.2**
+ dropout	**75.4**	**81.9**	89.8

Table 3: Results on the dataset of short answers written by language learners in response to visual prompts. Reporting accuracy, average precision, and precision at rank 50.

We trained for 300 epochs, measuring performance on the development set after every full pass over the data, and used the best model for evaluating on the test set. The parameters were optimised using gradient descent with the initial learning rate at 0.001 and the ADAM algorithm (Kingma and Ba, 2015) for dynamically adapting the learning rate during training. Dropout was applied to both word embeddings and image vectors with $p = 0.5$. In order to avoid any outlier results due to randomness in the model, which affects both the random initialisation and the sampling of negative image examples, we trained each configuration with 10 different random seeds and present here the averaged results.

4 Experiments

We evaluate the visual relevance detection model by training on Flickr30k and testing on the dataset of learner responses to visual prompts. In order to handle multiple sentences in the written responses, every sentence is first scored individually and the scores are then averaged over all the sentences. For every textual answer in the dataset, we create a negative datapoint by pairing it with a random image. The task is then to accurately detect whether the pair is truly relevant or randomly created, by assigning it high or low relevance scores. In order to convert the model output to a binary classification, we employ leave-one-out optimisation – one example at a time is used for testing, while the others are used to calculate the optimal threshold for accuracy. We also report average precision and precision at detecting irrelevant answers in the top 50 returned instances, which measure the quality of the ranking and do not require a fixed threshold.

Results for the different system architectures

0.65	In this picture there are lot of people and each one has a different attitude.
0.81	In the foreground, people are waiting for the green light in order to cross the street.
-2.75	While a child is talking with an adult about something that is on the other side of the road, instead a women, with lots of bag in her left hand, is chatting with her mobile telephone.
0.63	Generally speaking, the picture is full of bright colours and it conveys the idea of crowded city.
-2.38	Looking at this pictures reminds me of the time I went scuba diving in the sea.
-2.16	It’s fascinating, because you are surrounded by water and fishes and everything seems so coulorful and adventurous.
-1.40	Another good part of diving is coming up.
-1.70	You swim to the surface and you see the sunlight coming nearer and nearer until you get out and can breathe ”real” air again.

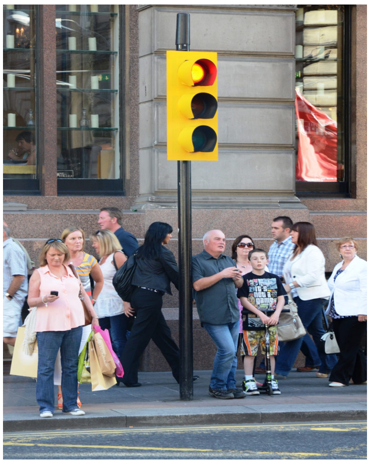

Table 2: Predicted scores from the best relevance scoring model, given example sentences from the learner dataset and the included photo as a prompt. The first 4 sentences were written in response to this image, whereas the last 4 were written about a different photo.

can be seen in Table 3. The baseline LSTM-COS system is based on the framework by Kiros et al. (2014) – it uses an LSTM for composing a sentence into a vector, calculates the relevance score by finding the cosine similarity between the sentence vector and the image vector, and optimises the model using the hinge loss function. This model already performs relatively well and is able to distinguish between relevant and random image-text pairs with 68.2% accuracy.

On top of this model we incrementally add 3 modifications and measure their impact on the performance. First, we augment the model with the gating architecture described in Section 2.2. The vector representation of the text is used to calculate a dynamic mask, which is then applied to the image vector. This allows the model to first read the sentence, and then decide which parts of the image are more important for the similarity calculation. The inclusion of the gating component improves accuracy by 1.4% and average precision by 3%.

Next, we change the scoring and optimisation functions as described in Section 2.3. Cosine similarity measure is substituted with a dot product between the vectors, removing useful bounds on the score, but allowing more flexibility in the model. In addition, the hinge loss function is exchanged for calculating the negative cross-entropy over a softmax. While the hinge loss performs only pairwise comparisons and applies a sharp cut-off, softmax ties all the examples into a probability distribution and provides a more gradual prioritisation for the parameter optimisation. By introducing these changes, the accuracy is again increased by 1.5% and average precision by 4.4%.

	DEV		TEST	
	ACC	POS	NEG	ACC
Random	16.7	0.5	0.5	16.7
LSTM-COS	70.8	0.7	0.0	72.6
+ gating	75.6	0.5	-0.6	76.5
+ cross-ent	82.8	5.8	-5.2	83.8
+ dropout	**87.0**	5.6	-3.7	**87.4**

Table 4: Results for different system configurations on the Flickr30k development and test sets. We report accuracy and the average predicted scores for positive and negative examples.

Finally, we apply dropout with probability 0.5 to both the 300-dimensional word embeddings in the input sentence and the 1024-dimensional image representation produced by the BVLC GoogLeNet. By randomly setting half of the values to 0 during training, additional variance is introduced to the available data and the model is becomes more robust for handling noisy learner-generated text. Integrating dropout improves the performance further by 4.3% and average precision by 2.9%.

Table 2 contains examples of the predicted scores from the final model, given example sentences written by language learners. For most sentences, the model successfully distinguishes between relevant and irrelevant topics, assigning

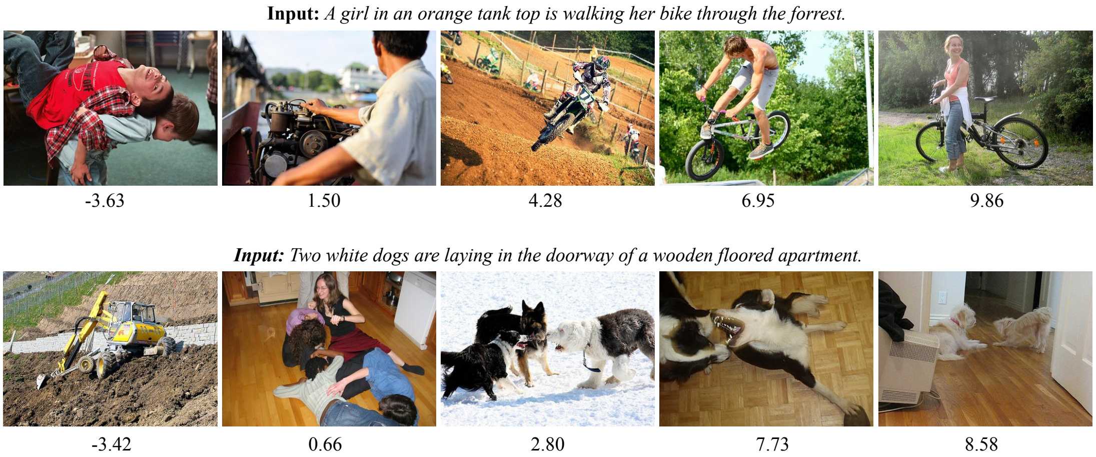
Input: *A girl in an orange tank top is walking her bike through the forrest.*
-3.63
1.50
4.28
6.95
9.86
Input: *Two white dogs are laying in the doorway of a wooden floored apartment.*
-3.42
0.66
2.80
7.73
8.58

Figure 2: Relevance scores for two example sentences, using the best model from Section 4. Higher values indicate higher confidence in the text being relevant to the image.

lower scores to the last 4 sentences that describe a different image. However, the model also makes a mistake and incorrectly assigns a low score to the third sentence – this likely happens due to the sentence being much longer and more convoluted than most examples in the training data, leading the LSTM to lose some important information in the sentence representation.

For comparison, we also evaluate the system architectures on the Flickr30k dataset in Table 4. In this setting, we present the model with a sentence and 6 images from the Flickr30k test set, one of which is known to be relevant while the others are selected randomly. Accuracy is then measured as the proportion of test cases where the model chooses the correct image as the most relevant one. A random baseline has a 1 in 6 chance of finding the correct image for an input sentence, as there are 5 negative examples for every positive example. We also report the average scores assigned by the models to positive (relevant) and negative (not relevant) pairs of images and sentences. As can be seen by the averaged predicted scores in Table 4, the final system is free to push positive and negative examples apart by a larger margin, increasing the average score difference by an order of magnitude.

5 Analysis

Figure 2 contains predicted scores for different images, given example sentences as input. As can be seen, the system returns high scores when the sentences are paired with very relevant images, and also offers an intuitive grading of relevance. For the first sentence describing an orange shirt and a bicycle, the model has assigned reasonably high scores to other images containing bikes and orange objects. Similarly, for the second sentence the system has found alternative images containing dogs and wooden floors.

In order to analyse the possible weaknesses of the model, we manually examined cases that are difficult for the system. Figure 3 contains 4 examples from the Flickr30k development set where a valid image-description pair received a negative score from the relevance model. While a negative score does not necessarily mean an error, as that depends on the chosen threshold, it indicates that the model has low confidence in this being a correct pairing. The use of rare terms is a source of confusion for the model – if a word was not used in the training data sufficiently, it will make the relevance calculation more difficult. For example, ”unicycle” and ”fire lit batons” are relatively rare terms that can cause confusion in example A. In addition, the description mentions only the man, while most of the photo depicts a crowd and a building.

An alternative source of confusion comes from the visual component, with GoogLeNet having more trouble with certain images. Out of 5,070 image-sentence pairs in the development data, the best model assigned negative scores to 222. Out of those, only 140 had a unique image, indicating that the visual component has more trouble detecting the content of certain unusual images, such as examples C and D, regardless of the textual composition. Both of these issues represent cases where the model is faced with input that is substantially different from the training examples, and therefore

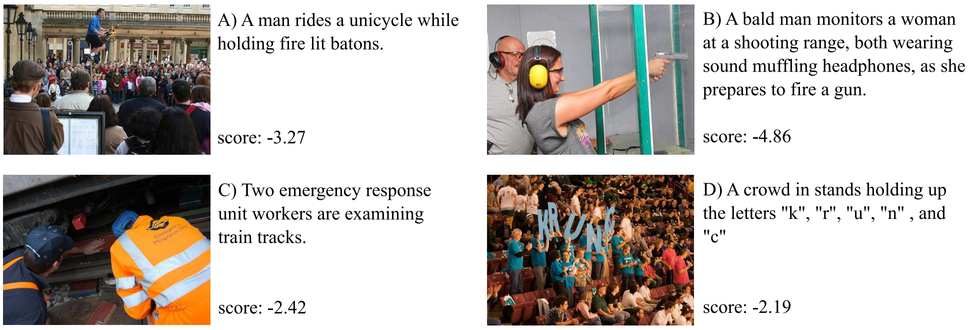
A) A man rides a unicycle while holding fire lit batons.
score: -3.27
B) A bald man monitors a woman at a shooting range, both wearing sound muffling headphones, as she prepares to fire a gun.
score: -4.86
C) Two emergency response unit workers are examining train tracks.
score: -2.42
D) A crowd in stands holding up the letters "k", "r", "u", "n" , and "c"
score: -2.19

Figure 3: Example valid pairs of images and sentences from the Flickr30k development set where the system incorrectly predicts a low relevance score.

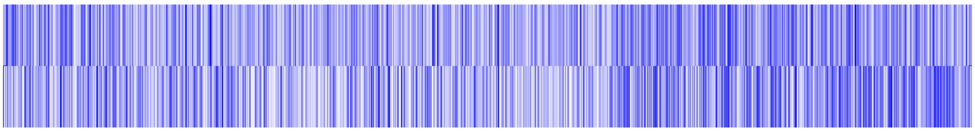

Figure 4: Visualisation of the 1,024 visual gating weights for two example sentences. Lighter areas indicate features where the model chooses to discard the visual information.

fails to perform as well as possible. This can be remedied by either creating models that are able to generalise better to unseen examples, or by expanding the sources of available training data.

We also analysed the gating component, which is conditioned on the text vector and applied to the image vector. The calculation of the gating weights includes a bias term and a logistic function, which means it could easily adapt to always predicting a vector of 1-s, effectively leaving the image vector unmodified. Instead, we found that the model actively makes use of this additional architecture, choosing to switch off many features in the image vector. Figure 4 shows a visualisation of the 1024 gating weights for the two example sentences used in Figure 2. Values close to 0 are represented by white, and values close to 1 are shown in blue. As can be seen, quite a few features receive weights close to zero, therefore effectively being turned off. In addition, the two sentences have fairly different gating signatures, demonstrating that weights are being calculated dynamically based on the input sentence.

6 Conclusion

We presented a system for mapping images and sentences into a shared distributed vector space and evaluating their semantic similarity. The task is motivated by applications in automated language assessment, where scoring systems focusing on grammaticality are otherwise vulnerable to memorised off-topic answers.

The model starts by learning embeddings for words in the input sentence, then composing them to a vector representation using an LSTM. In parallel, the image is first passed through a pre-trained image detection model to extract visual features, and then a further supervised layer to transform the representation to a suitable space. We found that applying dropout on both word embeddings and visual features allowed the model to generalise better, providing consistent improvements in accuracy.

Next, we introduced a novel gating component which first reads the input sentence and then decides which visual features from the image pipeline are important for that specific sentence. We found that the model actively makes use of this component, predicting different gating patterns depending on the input sentence, and substantially improves the overall performance in the evaluations. Finally, we moved from a pairwise hinge loss to optimising a probability distribution over the possible candidates, and found that this further improved relevance accuracy.

The experiments were performed on two different datasets – a collection of short answers written by language learners in response to visual prompts, and an image captioning dataset which pairs single sentences to photos. The relevance assessment model was able to distinguish unsuitable image-sentence pairs on both datasets, and the model modifications showed consistent improvements on both tasks. We conclude that automated relevance detection of short textual answers to visual prompts can be performed by mapping images and sentences into the same distributed vector space, and it is a potentially useful addition for preventing off-topic responses in automated assessment systems.

References

Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. 2016. Automatic Text Scoring Using Neural Networks. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* .

James Bergstra, Olivier Breuleux, Frederic Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. 2010. Theano: a CPU and GPU math compiler in Python. *Proceedings of the Python for Scientific Computing Conference (SciPy)* http://www-etud.iro.umontreal.ca/ wardefar/publications/theano_scipy2010.pdf.

Ted Briscoe, Ben Medlock, and Øistein Andersen. 2010. Automated Assessment of ESOL Free Text Examinations. Technical report.

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A Thorough Examination of the CNN / Daily Mail Reading Comprehension Task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Ronan Cummins, Helen Yannakoudakis, and Ted Briscoe. 2015. Unsupervised Modeling of Topical Relevance in L2 Learner Text. In *In Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition* https://doi.org/10.1109/CVPR.2009.5206848.

Andrea Frome, Greg S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc' Aurelio Ranzato, and Tomas Mikolov. 2013. Devise: A deep visual-semantic embedding model. *Advances in neural information processing systems* http://papers.nips.cc/paper/5204-devise-a-deep-visual-semantic-embedding-model.

Derrick Higgins, Jill Burstein, and Yigal Attali. 2006. Identifying Off-topic Student Essays Without Topic-specific Training Data. *Natural Language Engineering* 12. https://doi.org/10.1017/S1351324906004189.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-term Memory. *Neural Computation* 9. https://doi.org/10.1.1.56.7752.

Youmna Hussein, Marek Rei, and Ted Briscoe. 2017. An Error-Oriented Approach to Word Embedding Pre-Training. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*.

Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional Architecture for Fast Feature Embedding. In *ACM International Conference on Multimedia*. https://doi.org/10.1145/2647868.2654889.

Nal Kalchbrenner, Ivo Danihelka, and Alex Graves. 2015. Grid Long Short-Term Memory. *arXiv preprint arXiv:1507.01526* http://arxiv.org/abs/1507.01526.

Andrej Karpathy and Fei Fei Li. 2015. Deep visual-semantic alignments for generating image descriptions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* https://doi.org/10.1109/CVPR.2015.7298932.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-Aware Neural Language Models. *In Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16)* http://arxiv.org/abs/1508.06615.

Levi King and Markus Dickinson. 2016. Shallow Semantic Reasoning from an Incomplete Gold Standard for Learner Language. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*.

Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: a Method for Stochastic Optimization. In *International Conference on Learning Representations*.

Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. 2014. Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models. *arXiv preprint arXiv:1411.2539* http://arxiv.org/abs/1411.2539v1.

Benjamin Klein, Guy Lev, Gil Sadeh, and Lior Wolf. 2015. Associating neural word embeddings with deep image representations using Fisher Vectors. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* pages 4437–4446. https://doi.org/10.1109/CVPR.2015.7299073.

Lin Ma, Zhengdong Lu, Lifeng Shang, and Hang Li. 2015. Multimodal convolutional neural networks for matching image and sentence. In *Proceedings of the IEEE International Conference on Computer Vision*. pages 2623–2631. https://doi.org/10.1109/ICCV.2015.301.

Tomáš Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the International Conference on Learning Representations (ICLR 2013)*. https://doi.org/10.1162/153244303322533223.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 Shared Task on Grammatical Error Correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*. http://www.aclweb.org/anthology/W/W14/W14-1701.

Isaac Persing and Vincent Ng. 2014. Modeling prompt adherence in student essays. In *52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*.

Marek Rei, Gamal K. O. Crichton, and Sampo Pyysalo. 2016. Attending to Characters in Neural Sequence Labeling Models. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING-2016)*. http://arxiv.org/abs/1611.04361.

Marek Rei and Ronan Cummins. 2016. Sentence Similarity Measures for Fine-Grained Estimation of Topical Relevance in Learner Essays. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*. http://ir.dcs.gla.ac.uk/ ronanc/papers/reiBEA2016.pdf.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, and Phil Blunsom. 2016. Reasoning about Entailment with Neural Attention. *International Conference on Learning Representations* .

Keisuke Sakaguchi, Michael Heilman, and Nitin Madnani. 2015. Effective Feature Integration for Automated Short Answer Scoring. *HLT-NAACL 2015 - Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings of the Main Conference* http://www.scopus.com/inward/record.url?eid=2-s2.0-84960119751&partnerID=tZOtx3y1.

Swapna Somasundaran, Chong Min Lee, Martin Chodorow, and Xinhao Wang. 2015. Automated Scoring of Picture-based Story Narration. *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications* .

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout : A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research (JMLR)* 15. https://doi.org/10.1214/12-AOS1000.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. *Computer Vision and Pattern Recognition (CVPR)* https://doi.org/10.1109/CVPR.2015.7298594.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing* .

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Charagram: Embedding Words and Sentences via Character n-grams. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. http://arxiv.org/abs/1607.02789.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A New Dataset and Method for Automatically Grading ESOL Texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. http://www.aclweb.org/anthology/P11-1019.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. In *Transactions of the Association for Computational Linguistics*. http://web.engr.illinois.edu/ aylai2/publications/TACL2014DenotationGraph.pdf.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. In *Advances in Neural Information Processing Systems*. http://arxiv.org/abs/1509.01626#.