What a face says

– sentiment classification on facial and linguistic expressions

Embodied and Situated Language Processing Project
Gothenburg University 2017
Ida Rørmann Olsen, gusrrmid@student.gu.se

This project is on sentiment classification: sentences are mapped with facial expressions to explore whether a combination would improve classification performance. The idea is that people have certain facial expressions when expressing certain content. Such a model would benefit situated language systems by providing knowledge on sentiment - what faces say. Three classification experiments are done: on facial expressions, on linguistic expressions, and combined. The image data features are extracted from a pre-trained CNN, plugged-in as last layer. A LSTM is trained on the linguistic expressions, and on a combined input of the sentences and images. The results show that combining the input in this setup does not help. This could be due to the data chosen. Further work on more and other data could continue the initial steps that this project has made towards a situated language system able to classify sentiment from visual and linguistic input.

1. Introduction

Sentiment analysis is a hot topic in NLP, and lots have been done to develop systems to decide a sentiment score on natural language expressions. An exploration of whether (or how) visual data can improve this process is not only interesting for wondering language technologists: knowledge of e.g. what faces goes with certain linguistic expressions can 1) hopefully smoothen human-machine interaction in robotics, (situated language systems) 2) possibly help e.g. autistic patients to map between linguistic expressions, faces and feelings, and 3) probably contribute to the work on detecting sarcasm (or lies or ..), if a face doesn't match an expression.

This project combines categories of facial expressions with sentiment scores for natural language sentences. More specifically, it is an implementation to explore the possibility of predicting a sentiment category learned on a bunch of sentences associated with a group of images with certain facial expressions.

The question is whether adding visual data to the language data will improve the sentiment prediction.

The steps in the present work is to:

1) (manually) pair the sentences with the chosen facial expression – building a synthetic dataset
2) Save the sentiment score of the applied sentences. Convert from nominal to categorical
3) Build 3 models (LSTM): language only, faces only, and fit the pairs of sentences and faces with the sentiment category (several experiments)
4) Evaluate

In section 2, the toolkits, methods and decisions throughout the experiment is described. In section 3 the results are presented, and will be discussed in section 4. Finally, section 5 gives concluding remarks

and ideas for future improvement and work with the data.

2. Materials and methods

The visual data (D.L. Goodfellow et al. 2013) is 48x48 pixel grayscale images of faces labelled with the seven universal facial expressions: angry, disgust, fear, happy, sad, surprise, and neutral. The dataset consists of around 33.000 images.

Images from the happy and angry category

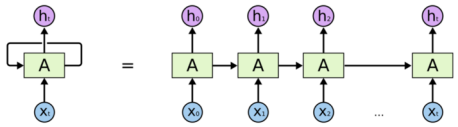
The features from this data is extracted with the VGG16 architecture [1] (with several convolutional blocks) that is pre-trained on ImageNet dataset. Weights from the data is used at the final fully-connected classification layer when building the image-only model. [1]

The linguistic data (Richard Socher et al. 2013) is the classic set of movie review tweets from *Stanford Sentiment Treebank*, labelled with a sentiment score from 0 to 1 (from negative to positive). It includes 215,154 phrases, but I included only the longest 150,000.[2] The sentences from the linguistic dataset was tokenized, lemmatized, lowercased. Keras was used to embed the sentences.

The neural networks are build with the Keras package (Francois Chollet et al. 2015) in Python3. See attached Jupyter notebook.

Pandas (Wes McKinney 2010) and NumPy (Stéfan van der Walt 2011) was applied for pre-processing and data handling.

The LSTM (Hochreiter & Schmidhuber, 1997) is a kind of a recurrent network, that can "remember" and keep information previously seen in internal "long-term memory" states. This stored contextual information influence the current predictions, which results in a mechanism that *"[…] allows RNNs to exploit a dynamically changing contextual window over the input sequence history".*(Hassim Sak, et al., 2014)



Firstly, I build a language-only LSTM model. Secondly, an image-only LSTM model. The models are set to predict the classes 'negative' or 'positive' (angry=negative, happy=positive).

Finally, a combination model: I mapped the negative and positive sentences with the angry and happy faces, respectively. The image data sets the upper limit (in size), so every image was paired several sentences. The resulting data is 57184 triplets which consists of sentences, the images and sentiment category.

Visual and language data as inputs to a system is not rare, but this is a special case: The technique is not image captioning, (since the goal is not describing the input images), nor is it a typical Q&A, since the LSTM model is not trained on question-answer pairs, but pairs of sentences and images. There is a right answer though, a sentiment score (categorical), which is to be

[1] As here https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html

[2] Higher chance of getting a sentence, rather than punctuations or smileys

classified.[3] The data can be used for several purposes, see discussion in section 5.

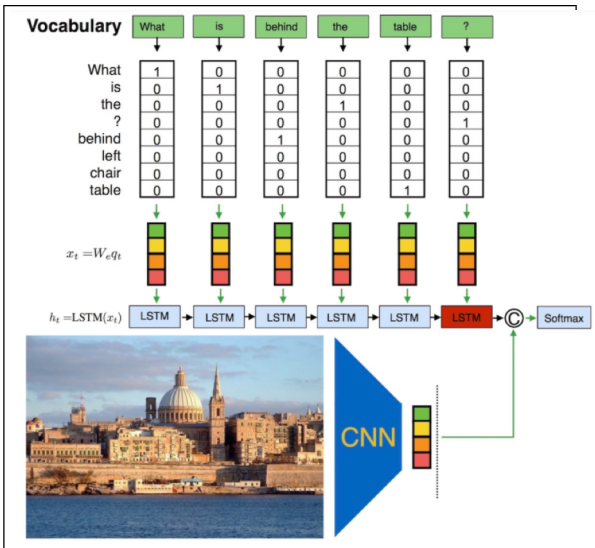


Figure 1: Example from visual Q&A model. Input question (green) is represented in one-hot vectors with the vocabulary (left). Input context is a picture. Then through a LSTM layer. A pretrained CNN model is used on the input context at the end before softmax

The hypothesis is that pairing the sentences and their sentiment scores with the images of different categories of facial expressions will confuse the sentiment prediction (experience show). Will a model perform better by combining this visual data and this language data?

3. Results

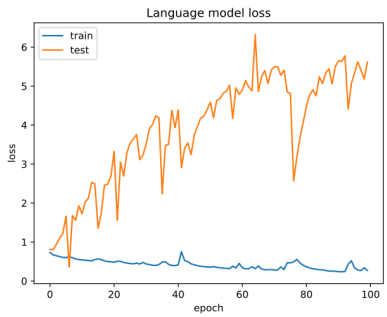


Figure 2: Language model loss

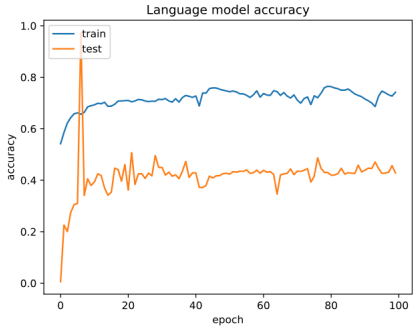


Figure 3: Language model accuracy

The language-only model performs with an average around 40% in the test set, and about 70% for training, with an increasing test loss, decreasing train loss. This indicates that it is getting better and better at predicting while training (overfitting) but worse at predicting the test set: it is memorizing the data. A cleaner dataset and change of parameters (as dropout) might help. See discussion below.

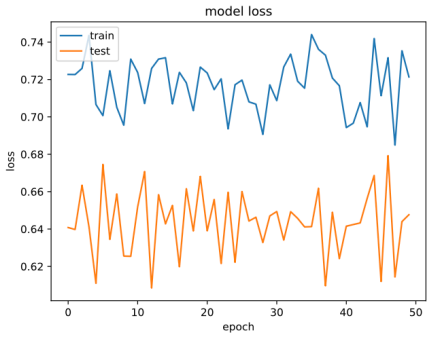


Figure 4: Image model loss

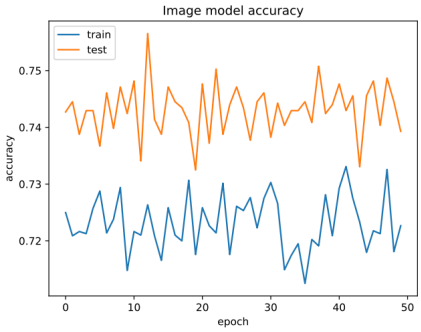


Figure 5: Image model accuracy

[3] Changed from regression for simplicity.

The image-only model performs better (and better than chance), at a bit over 70% accuracy, but the loss is very high and not decreasing over training time. This indicates again that the model does not get better given more data, but is confused about what function to generate.

An improvement of the accuracy with a combination of this visual and language data would be possible if the model found a pattern within the two categories, *positive* and *negative*, in the sentences, as well as in the images. This was not the case:

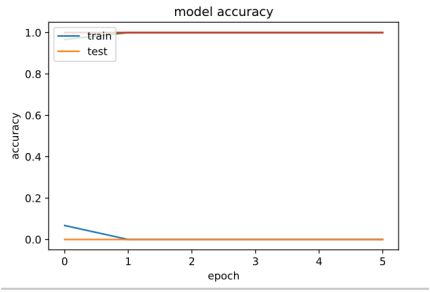


Figure 6 Combination model

The model predicts with 100% accuracy (red), and minimal constant loss (yellow and blue) is very low and constant. This was not what I expected, considering the nature of the data and the previous two models. The model history indicates that the input to the models is either seen before, or that the input data reveals the answer.

The hypothesis of a worse performance with combining language data with visual data d oes hold, in this exact setup. The model did not learn, given the training history. This c an be due to the several reasons, that I will discuss below.

4. Discussion

The language learning is not impressing, indicating that the model did not find a significant pattern in the tweets. A deeper semantic analysis of the sentences might reveal informative clues for the model to distinguish the classes. Training a character-based language model on another English corpus, and then use the encoded knowledge for a sentiment analysis of my data – and for the visual data too, as suggested above. This would give the model more "background knowledge" to base predictions on. The data is in itself noisy with many empty sentences and repetitions across classes.

The image-only model used a pre-trained CNN for extracting features from given pixels. The CNN was not trained on facial expression though (but object recognition), which probably would benefit this task. Classifying facial expressions is a complex task even for humans, and would need more data and computational power than possible here.

The combination model performed suspiciously well. Given the data and previous models (and experience) I expect lower precision. Further work on data and model parameters might help: increasing dropout size in a neural network helps avoid overfitting.
Finally, if we assume to have enough and appropriate data, the way to construct the synthetic dataset can be questioned too. I chose to select the part of the corpus with the longest sentences, and then map to facial expression categories. Also, we do not always look the same when saying certain things, and the choice of mapped categories could be changed as well.

5. Conclusions and further work

Three experiments were done: one only on images, one only on language, one where a

LSTM trained jointly on images and sentences. Neither of the three models performed promising. Better performance by joining sentiment bearing sentences with facial expressions was not achieved in this setup. Further work would be to train language and image models on more and cleaner data. A bigger language dataset with deeper linguistic analysis would probably be beneficial. The improvements would hopefully confirm the idea of this project: My claim is still what we express generally comes with certain facial expressions, and we can use this in situated language systems. Despite insignificant results, this project was a step towards this purpose.

6.. References

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher Manning, Andrew Ng and Christopher Potts: *Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank*, Conference on Empirical Methods in Natural Language Processing (EMNLP 2013) : https://nlp.stanford.edu/sentiment/treebank.html

I. Goodfellow, D. Erhan, P.L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D.H. Lee, Y Zhou, C Ramaiah, F. Feng, R. Li, X. Wang, D. Athanasakis, J. Shawe-Taylor, M. Milakov, J. Park, R. Ionescu, M. Popescu, C. Grozea, J. Bergstra, J. Xie, L. Romaszko, B. Xu, Z. Chuang, and Y. Bengio: "*Challenges in Representation Learning: A report on three machine learning contests."* arXiv 2013.

Francois Chollet et al.: *Keras,* on GitHub, 2015 - https://github.com/keras-team/keras

McKinney, Wes. (2010) *Pandas: a Foundational Python Library for DataAnalysis and Statistics* presented at SciPy 2010

Stéfan van der Walt, S. Chris Colbert and Gaël Varoquaux. *The NumPy Array: A Structure for Efficient Numerical Computation*, Computing in Science & Engineering, **13**, 22-30 (2011),

Sak, Haşim & Senior, Andrew & Beaufays, Françoise. (2014). Long Short-Term Memory Based Recurrent Neural Network
Architectures for Large Vocabulary Speech Recognition.

Hochreiter, S., Schmidhuber, J.: *Long Short-Term Memory* in Neural Computation Journal, vol. 9 Issue 8, pp. 1735-1780, MIT Press Cambridge, USA, 1997

[1] : VGG16 architecture: