Automated paper sheet evaluation using Machine Learning

Koffi Agbenya[1], Yu Bai Hao[1], Franck Gechter[2], and Fabrice Lauri[3]

[1]*Computer Science department, University of Technology of Belfort Montbéliard (UTBM)*
[2]*IRTES, University of Technology of Belfort Montbéliard(UTBM). Email: franck.gechter@utbm.fr*
[3]*IRTES, University of Technology of Belfort Montbéliard (UTBM). Email: fabrice.lauri@utbm.fr*

Abstract

Evaluating both subjective and objective answers is a complex task. The main difference between human and machine when evaluating anything is that human judgement can vary because of emotions for instance while machine judgement will remain the same. We proposed in this work a system capable of evaluates students paper sheet using machine learning algorithm such as Natural Language Processing (NLP), Computer Vision (CV).

Keywords: NLP, Stemming, LSTM, FCN, Semantic similarity, Siamese network, Manhattan metric, CNN, Text detection

1 Introduction

Measuring the semantic relatedness of two pieces of text is a fundamental problem in natural language processing tasks like question answering, plagiarism detection or query ranking. In this paper we address the sentence similarity measurement problem as a proof of concept in order to adress the paper sheet evaluation problem. Given a paper sheet P1 and a ground truth sheet P2, the task is to extract first the exercices, then the answers per question and compute their similarity in term of score. This score is approximately the mark given to the answer.

Measuring text similarity is challenging because of the variability of linguistic expression and the limited amount of annotated training data. One of the solution is by modeling the underlying semantic similarity between sentences/phrases. Particularly, a good model should not be susceptible to variations of wording/syntax used to express the same idea. However, it remains a big issue since labeled data is scarce and sentences have both complex structure and variable length.

Our task is more complexed than the classic task of measuring similarity. The space defined by the questions we deal with is composed of two types of question: Objective questions and subjective questions. The way we deal with the objective questions is not the same as the way we deal with the subjective question. Objective question is the question which require students to select the correct response from several alternatives or to supply a word or short phrase to answer a question or complete a statement. As example we can cite: multiple choice, true-false, matching, completion. Subjective or essay permit the student to organize and present an original answer. As example we have: short-answer essay, extended-response essay, problem solving, performance test items. In addition, we are facing the task of extract with computer vision and text extraction technics the content of the paper sheet.

Recent successes in sentence similarity and text extraction from an image have been obtained by using neural network (Tai et al. 2015; Mueller et al. 2016). Our approach is also based on neural networks: we propose a modular functionnal architecture with different pipeline. In input there are a text detection system and text extraction system which extract text from student's scanned paper. The text extracted from the previous system will be pre-processed with NLP technics and

Artificial intelligence (2020)

DOI: 10.1017/pan.xxxx.xx

Corresponding author
Franck Gechter

Edited by
John Doe

then provides to machine learning algorithm which task will be to compare the students answers to correction model and gives mark.

In the other hand, Text detection is a branch of target detection of deep learning. With the popularity of deep learning, text detection business is also playing a role in more and more fields for example in autopilot field.

Because of the complexity of the task, we studied the pipelines separately. The first part consist of using the models EAST(Efficient and Accurate Scene Text Detector) (X. Zhou *et al.* 2017), then complete text recognition by using C-RNN (Ren *et al.* 2015) and CTC (Graves *et al.* 2006) framework. The second part consist of using a siamese LSTM (Long Short Term Memory) for learning sentence similarity based on Manhattan metric. We demonstrate state-of-art performance in both tasks. Our EAST + RCNN model is applied to a dataset composed of student sheets from an assessment. Our Manhattan based Siamese LSTM network is applied to address the problem of Quora Question Pairs Kaggle competition task wich is similar to our problem.

The rest of the paper is organised as follows: first of all, we present the related work, secondly we describe the datasets, then we describe the models we used to perform the tasks. Finally we discuss our results and propose a possible solution to address our original task.

2 Related Work

Several studies have been carried out to deal with this subject. Thus, Piyush Patil et al. proposed in (Patil *et al.* 2018) a system uses machine learning and NLP to solve the problem. Their algorithm performs tasks like Tokenizing words and sentences, part of speech tagging, chunking, chinking, lemmatizing words and wordnetting to evaluate the subjective answer. The algorithm provides the semantic meaning of context. The proposed system is divided into two modules. The first one is extracting the data from the scanned images and organizing it in the proper manner and the second is applying ML and NLP to the text retrieved from the above step and giving marks to them.

Bhuvaneswari et al. on the other hand proposes in paper titled Semantic similarity based answer sheet evaluation using NLP (Bhuvaneswari *et al.* 2017), a system that automatically evaluates the answer sheet in the form of text document for both objective and subjective type questions using NLP. Authors based their system on different NLP techniques such as token separation, parts of speech tagging, stop words removal, stemming words. They have used the popular NLTK (Natural Langauge Toolkit) for processing. It's important to note that their system is based on digital documents. After processing step, the system of evaluation uses semantic similarity technic. The similarity computing is based on cosine similarity.

In 2017, Lakshmi et al. proposed in their work Evaluating Students' descriptive answers using NLP and ANN (Lakshmi and Ramesh 2017) a slightly different way to deal with the problem. They create answer sheet and keyword dataset for examination process. These dataset are stored in data storage and the student enters their answers in the examination page. The system automatically calculates result using two algorithms of NLP and ANN. The system consist of three steps. Pre-processing step where pre-processing technique is applied on the answers entered by the students. Comparison step where they used an ANN algorithm for the normal answer comparison and stores marks in database. Checking step where NLP techniques are used to evaluate the same answer with previous to check grammar mistakes and stores the marls in database and finally compares both marks and provides final result. They proposed in the paper for text mining process to use tools like NLP and WordNet. They group the English words into some of sets of synonyms called synsets which provides short definitions and usage examples, and records a number of relations among these synonym sets. NLP technics they have used are Part of Speech Taggin uses here to extract the important keywords in the answer given by staff before assessment is done., mispelling words, stop words removal, WordNet tool is used to give the related synonyms to literal word in the subordinate terms. Evaluation step is performed using ANN. It role is normal answer comparison

and stores marks in database. Here the answer will be evaluated only by normal comparison of text using the keywords. Each and every word of student answer is compared with correct answer. If student answer is match with correct answer increase scores are assigned using ANN algorithm. After score assignment final scores are divided by summation of assigned scores of all words.

Although the measure of similarity is rather satisfactory in the previous cited works, it is important to note that new measurement methods are current and more effective than that used by the previous authors. Thus, Jonas Mueller et al. in their work titled Siamese Recurrent Architectures for Learning Sentence Similarity (Mueller and Thyagarajan 2016) presents a siamese adaptation of the long short-term memory (LSTM) network (Hochreiter and Schmidhuber 1997) for labeled data comprised of pairs of variable-length sequences. Their model is applied to assess semantic similarity between sentences. They provide word-embedding vectors supplemented with synonymic information to the LSTMs, which use a fixed size vector to encode the underlying meaning expressed in a sentence. The proposed model called Manhattan LSTM Model (MaLSTM). There are two networks LSTMa and LSTMb which each process one of the sentences in a given pair. It's important to note that LSTMa = LSTMb here. Like many top performing semantic similarity systems, their model takes as input word-vectors which have been pre-trained on an external corpus. They use the 300-dimensional word2vec embeddings which Mikolov et al demonstrate can capture intricate inter-word relationships. The MaLSTM predicts relatedness for a given pair of sentences and they train the siamese network using backpropagation-through-time under the Mean Squared Error (MSE) loss function. Their model uses 50-dimensional hidden representations ht and memory cells ct. Optimization of the parameters is done using Adadelta method of Zeiler along with gradient clipping to avoid the exploding gradients problem. It's well known that the success of LSTMs depends crucially on their initialization, and often parameters transferred from neural networks trained for a different task can serve as a strong starting point for the optimization. They first initialize their LSTM weights with small random Gaussian entries.

In the same way, Yassine Benajiba et al studied semantic pattern similarity in (Benajiba *et al.* 2019). They utilize siamese networks to model semantic pattern similarity task to determine SQL patterns for unseen questions in a database-backed question answering scenario. They have used the WikiSQL data set, which contains over 87000 natural language questions aligned with SQL queries that produce the answer. To model the similarity of two questions the authors use recurrent siamese neural networks. Their approach is given an unseen question, find in a pool of questions, another question with the same semantic pattern. To perform this, they employ siamese LSTM regression model to predict the similarity of the SQL templates of two questions. One of the engineering things they do is instead of comparing the unseen question to the entire training set, they cluster the training set ahead of time using lexical representation of the questions and compare a new question only to the members of its nearest cluster.

One of the most complicated tasks for this type of system is detecting and extracting the text from the scanned paper sheet. Xinyu Zhou et al. in proposes a simple yet powerful pipeline that yields fast and accurate text detection in natural scenes. The pipeline directly predicts words or text lines of arbitrary orientations and quadrilateral shapes in full images, eliminating unnecessary intermediate steps, with a single neural network. The pipeline consist of two stages FCN and NMS merge state. The pipeline utilizes a fully convolutional network (FCN) model that directly produces word or text-line level predictions, excluding redundant and slow intermediate steps. The produced text predictions, which can be either rotated rectangles or quadrangles, are sent to Non-maximum suppression to yield final results.

Text detection is similar to detection for general objects, therefore, the text detection framework is based on the improvement of the traditional target detection framework (X. Zhou *et al.* 2017), (Ren *et al.* 2015), (Graves *et al.* 2006). Due to issues such as text shape, aspect ratio, and direction, traditional target detection frameworks do not perform well on text detection tasks. In the literature,

the text detection tasks can be classified into two approaches: Top-down and bottom-up. In (Tian *et al.* 2016), the authors improved the faster rcnn framework, detected each character, and applied the B-LSTM network to connect characters into text lines. But this method will accumulate errors, resulting in poor final results. In (Shi, Bai, and Yao 2017), they proposed Seglink, another method of bottom up. Seglink is based on the SSD framework. It can simultaneously return the text area and its connection relationship, and can more accurately return the coordinates and angle information of the bounding box. In (Deng *et al.* 2018), the authors used image segmentation technology to achieve text detection. The top-down method has inherent advantages over the bottom-up method. In (Ma *et al.* 2018), the authors have improved the Faster-RCNN network and added an attribute angle to the bounding box, which allows the RRPN network to detect slanted text. In (Y. Zhou *et al.* 2017), the authors used arbitrary quadrilaterals instead of rectangles to express the text area boundaries more compactly.

3 Presentation of the datasets

3.1 ICDAR 2015 dataset and Synth 90k dataset

ICDAR 2015 dataset is owned by International Conference on Document Analysis and Recognition, which includes 1000 training images and 500 testing images. The annotations in this dataset are as follows. For each picture, there is at least one text box. The annotation file uses 8 integer values to describe the coordinates of the four vertices of the text box, and a string to describe the text information in the text box. For each text box in this picture, there are a total of 9 descriptions. We use the ICDAR dataset to train the EAST network. Synth 90k is a synthetic, very large dataset. Each picture is an independent text box. The label information of the file is the text in the text box. We use this data set to train our CRNN and CTC networks. We only need to write a script, we can use the label file to generate a training label file that meets the format.

3.2 UTBM PS22 class students sheet

This dataset is provided by one of our supervisors. It consists of a PDF document which gathered 41 copies of an assessment of analog electronic, the lecture is taught at UTBM as PS22. The dataset is used for testing purpose of our model.

3.3 Quora Questions Pairs

The dataset was provided at a kaggle competition organized by Quora. Quora is a place to gain and share knowledge about anything. It's a platform to ask questions and connect with people who contribute unique insights and quality answers. This empowers people to learn from each other and to better understand the world.

Over 100 million people visit Quora every month, so it's no surprise that many people ask similarly worded questions. Multiple questions with the same intent can cause seekers to spend more time finding the best answer to their question, and make writers feel they need to answer multiple versions of the same question. Quora values canonical questions because they provide a better experience to active seekers and writers, and offer more value to both of these groups in the long term.

The goal of the competition is to predict which of the provided pairs of questions contain two questions with the same meaning. The ground truth is the set of labels that have been supplied by human experts. The ground truth labels are inherently subjective, as the true meaning of sentences can never be known with certainty. Human labeling is also a 'noisy' process, and reasonable people will disagree. As a result, the ground truth labels on this dataset should be taken to be 'informed' but not 100% accurate, and may include incorrect labeling. We believe the labels, on the whole, to represent a reasonable consensus, but this may often not be true on a case by case basis for

individual items in the dataset.

4 Model Architectures

4.1 EAST Framework

The first pipeline of the framework consists of detecting the text on the paper sheet to extract the student's response. The EAST framework has achieved good results both in terms of detection accuracy and speed. The training of ordinary text recognition networks may be divided into several parts, and there may be some complicated and time-consuming operations in this process, such as the extraction of candidate frames, etc. The EAST framework simplifies the text detection pipeline, and the text box extraction is completed in only two steps: the text box is directly filtered by NMS after being detected by the FCN. Simplified steps can reduce the accumulation of errors.

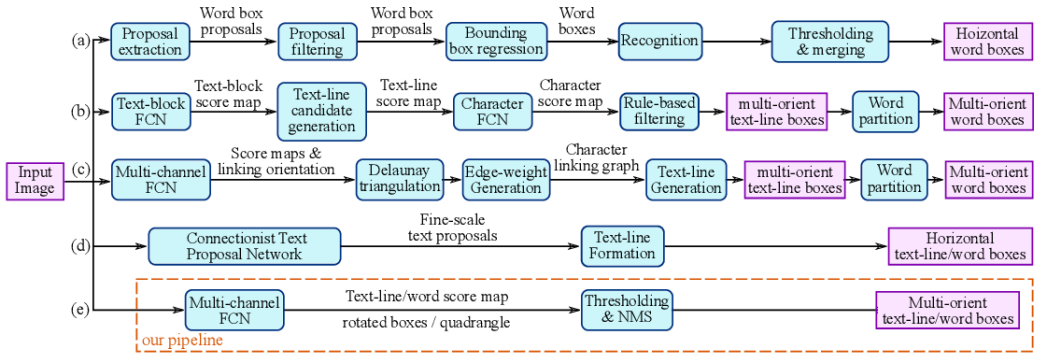

Figure 1. (e) EAST pipeline, which eliminates most intermediate steps, consists of only two stages and is much simpler than previous solutions.

The key component of the algorithm is a fully convolutional network (FCN). This network is trained to directly predict the existence of text instances and their geometry. The following is the neural network structure of this algorithm:

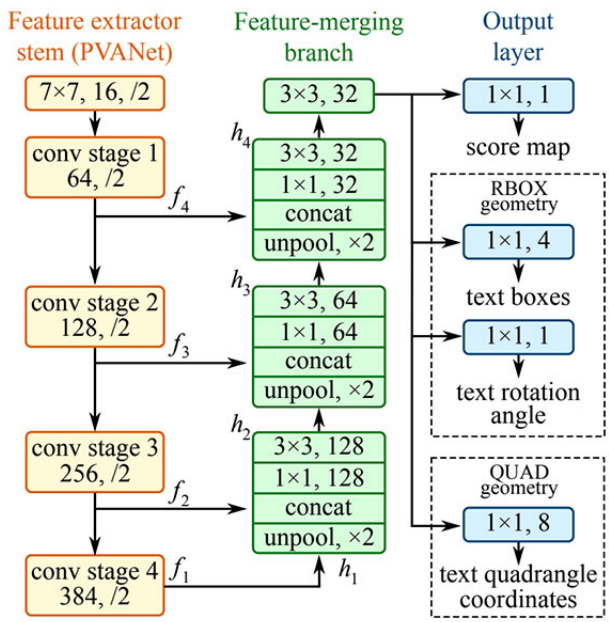

Figure 2. Structure of the EAST text detection FCN

The feature extraction backbone uses the pre-trained PVANet network on the ImageNet dataset

to generate 4 levels of feature maps, defined as f(i), and the size is 1/32, 1/16, 1/8, 1/4 of the origin image.

Geometry	Channels	Description
AABB	4	$G = R = \{d_i \mid i \in \{1, 2, 3, 4\}\}$
RBOX	5	$G = \{R, \theta\}$
QUAD	8	$G = Q = \{(\Delta x_i, \Delta y_i) \mid i \in \{1, 2, 3, 4\}\}$

Table 1. Output geometry design

The EAST network supports three output results. AABB represents a normal rectangular frame, the output is the horizontal and vertical coordinates of the vertex of the upper left corner of the rectangular frame and the height and width of the rectangular frame, RBOX represents the rectangular frame with a rotation angle, the output is the horizontal and vertical coordinates of the vertex of the upper left corner of the rectangular frame , the height and width of the rectangular frame and the rotation angle of the rectangular frame, and QUAD represents a quadrilateral with any shape, and the output is the horizontal and vertical coordinates of the four vertices of the quadrilateral. Taking into account the actual situation of our project, we believe that the use of RBOX can be well adapted to the data set and easy to train.

EAST uses both score map loss and geometric loss as loss functions. It is worth mentioning that we used class-balanced cross entropy (Cui *et al.* 2019) as the score map loss.

$$L = L_s + \lambda_g L_g$$

For the specific calculation process of the loss function, this paper does not make too much introduction.

Algorithm 1 Locality-Aware NMS

```
function NMSLocality(geometries)
    S ← ∅, p ← ∅
    for g ∈ geometries in row first order do
        if p ≠ ∅ ∧ ShouldMerge(g, p) then
            p ← WeightedMerge(g, p)
        else
            if p ≠ ∅ then
                S ← S ∪ {p}
            end if
            p ← g
        end if
    end for
    if p ≠ ∅ then
        S ← S ∪ {p}
    end if
    return StandardNMS(S)
end function
```

Figure 3. The algorithm Locality-Aware NMS

EAST uses the LANMS algorithm instead of the NMS algorithm to reduce the time complexity from $O(n^2) to O(n)$.

4.2 CRNN + CTC

In the text recognition task, we use the CRNN + CTC framework. Here, these two parts are inseparable. The CRNN network is a combination of CNN and RNN. The overall framework of this part is as follows:

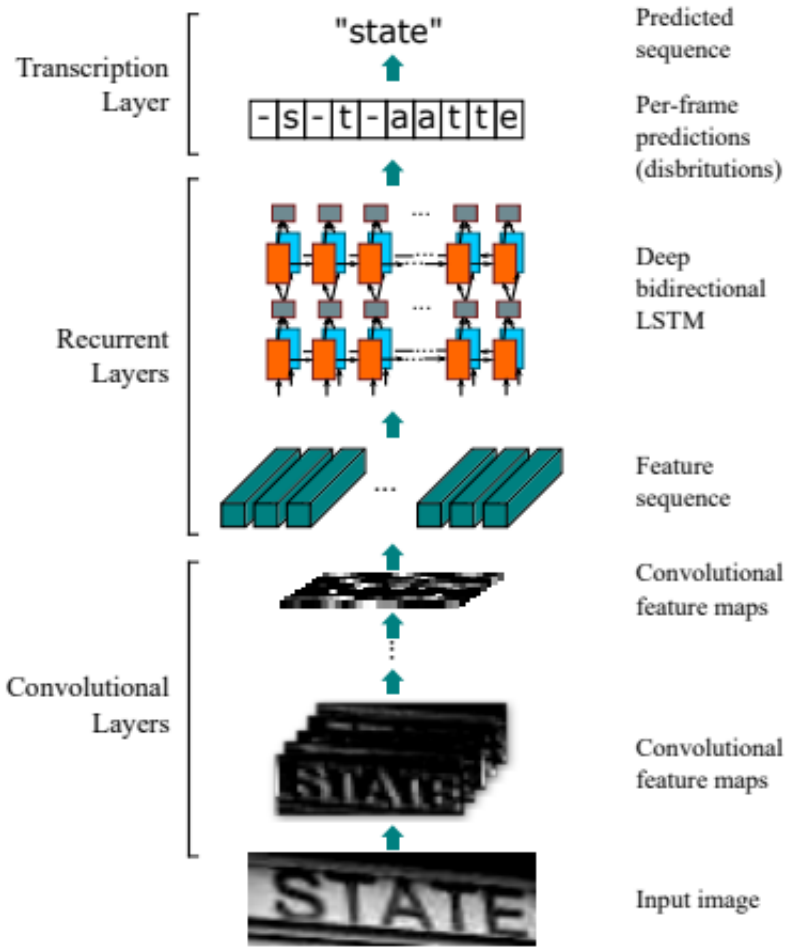

Figure 4. Structure of CRNN & CTC

We first use a sliding window to slide along the image from left to right. Convolutional neural networks extract image features in each sliding window image. After the sliding window has scanned the entire image completely, we obtain a sequence of image features.

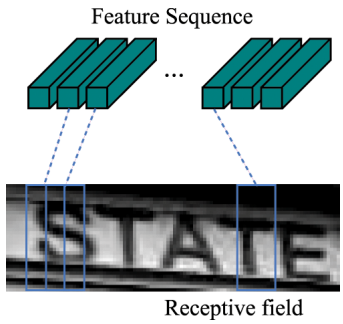

Figure 5. Convolutional layers

Next, we input the obtained feature sequence into a recurrent neural network. In this paper, we use a bidirectional-LSTM network.

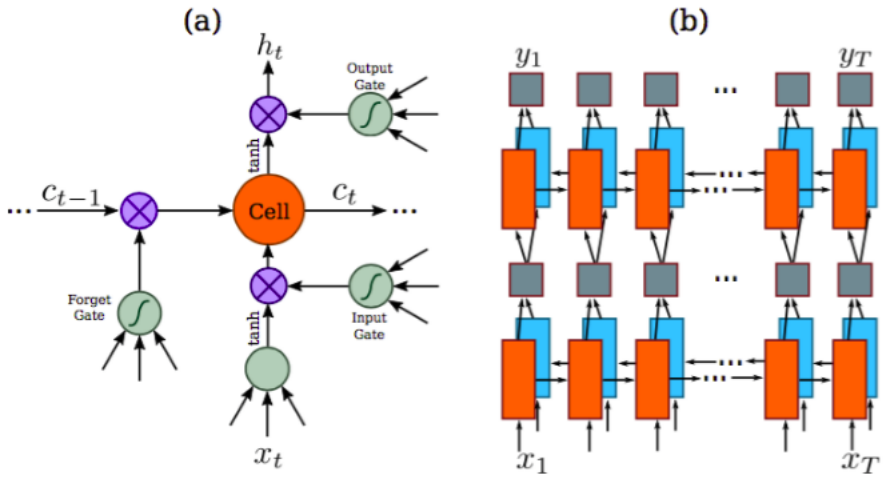

Figure 6. (a) The structure of a basic LSTM unit. An LSTM consists of a cell module and three gates, namely the input gate, the output gate and the forget gate. (b) The structure of deep bidirectional LSTM we use in our paper. Combining a forward (left to right) and a backward (right to left) LSTMs results in a bidirectional LSTM. Stacking multiple bidirectional LSTM results in a deep bidirectional LSTM.

Eventually, the bidirectional-LSTM network may output the predicted letters. However, because each letter will occupy multiple sliding windows in the original image, we need to filter the repeated letters to get the final output result.

4.3 MaLSTM

MaLSTM stands for Manhatann LSTM and is proposed by (Mueller and Thyagarajan 2016). The Manhatann LSTM is a model using a siamese neural network of two LSTMs to measure similarity between a pair of sentences. The two LSTMs convert the variable length sequence into a fixed dimensional vector embedding. A similarity function is then applied on top of these vectors to compute a similarity measure. The last hidden state is used as the vector embedding for the sequence.

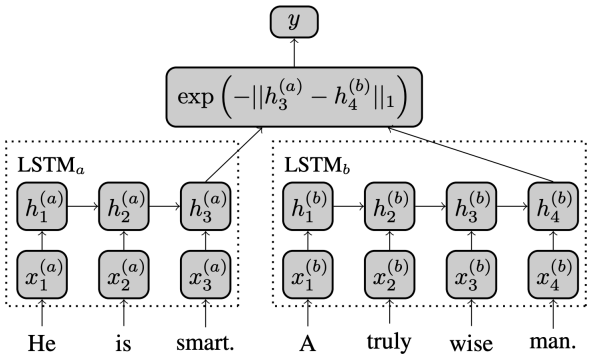

Figure 7. MaLSTM model structure

The LSTM (Long-Short Term Memory) is a variant of recurrent neural network proposed by (Hochreiter and Schmidhuber 1997) and is well-known for fixing the vanishing gradients problems that suffer the standard RNN. The LSTM sequentially updates a hidden-state representation, but these steps also rely on a memory cell containing four components (which are real-valued vectors): a memory state c_t, an output gate o_t that determines how the memory state affects other units, as

well as an input (and forget) gate i_t (and f_t) that controls what gets stored in (and omitted from) memory based on each new input and the current state. Below are the updates performed at each $t \in \{1, ..., T\}$ in an LSTM parameterized by weight matrices $W_i, W_f, W_c, W_o, U_i, U_f, U_c, U_o$ and bias-vector b_i, b_f, b_c, b_o:

$$i_t = sigmoid(W_i x_t + U_i h_{t-1} + b_i) \tag{1}$$

$$f_t = sigmoid(W_f x_t + U_f h_{t-1} + b_f) \tag{2}$$

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{3}$$

$$c_t = i_t \odot \tilde{c}_t + f_t \odot c_{t-1} \tag{4}$$

$$o_t = sigmoid(W_o x_t + U_o h_t - 1 + b_o) \tag{5}$$

$$h_t = o_t \odot \tanh c_t \tag{6}$$

In the original paper, for the given pair of sentences, the authors applies a pre-defined similarity function $g : \mathbb{R}^{d_{rep}} \times \mathbb{R}^{d_{rep}} \rightarrow \mathbb{R}$ to their LSTM-representations. Similarities in the representation space are subsequently used to infer the sentences' underlying semantic similarity. The LSTM here plays the role of encoder. The sole error signal backpropagated during training stems from the similarity between sentence epresentations $h^{(a)}_{T_a}, h^{(b)}_{T_b}$, and how this predicted similarity deviates from the human annoted ground truth relatedness. We restrict ourselves to the simple similarity function $g(h^{(a)}_{T_a}, h^{(b)}_{T_b}) = \exp(-\left\|h^{(a)}_{T_a} - h^{(b)}_{T_b})\right\|_1) \in [0, 1]$. This forces the LSTM to entirely capture the semantic differences during training, rather than supplementing the RNN with a more complex learner that can help resolve shortcomings in the learned representations.

In (Chopra, Hadsell, and LeCun 2005), the authors recommend to use ℓ_1 rather than ℓ_2 norm in similarity function. The reason is ℓ_2 can lead to undesirable plateaus in the overall objective function. This is because during early stages of training, a ℓ_2-based model is unable to correct errors where it erroneously believes semantically different sentences to be nearly identical due to vanishing gradients of the Euclidean distance.

5 Experiments and results

5.1 Text Detection and Text recognition

When we test after training, to get the best output, we need to find the best parameters of the network. Three of these parameters are critical. They are score map thresh, box thresh and nms_thres. If we also use the test set of the ICDAR data set for testing, when the model is optimized for output, the values of these three parameters are:

$$\text{score\_map\_thresh} = 0.8$$
$$\text{box\_thresh} = 0.1$$
$$\text{nms\_thresh} = 0.2$$

But since our actual data set is not the ICDAR data set, this optimization parameter does not apply to our own data set. Since we do not have a clear quantitative method to evaluate the output of the model, we obtain the optimized parameters by manual evaluation.

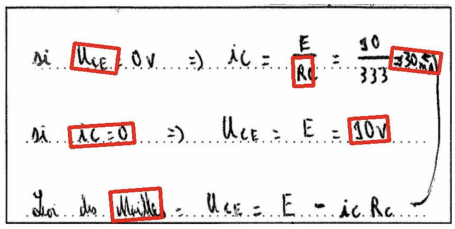

Figure 8. EAST output score map thresh = 0.8

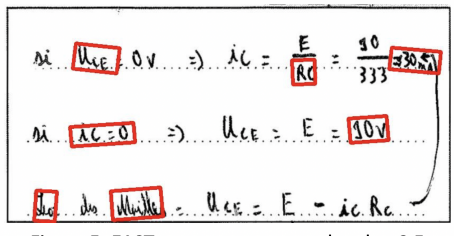

Figure 9. EAST output score map thresh = 0.5

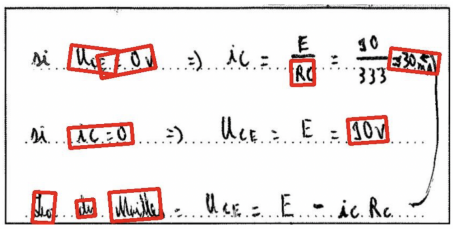

Figure 10. EAST output score map thresh = 0.25

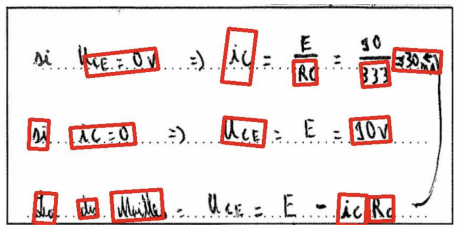

Figure 11. EAST output score map thresh = 0.05

We found that when the threshold is lower, the more text boxes can be detected, the better the model is. Finally, our model parameters are set to:

score_map_thresh = 0.0.5

box_thresh = 0.001

nms_thresh = 0.001

In addition, input pictures of different sizes also have a significant effect on the final output result. Since the training data set we use is ICDAR, we need to try to make the average size of the text box of our own input image and the average size of the text box of the ICDAR training dataset be same. After experiments, we found that directly inputting the answer box at the original size (the size of the machine-readable card is 2480 * 3506) will make the model have the best output.

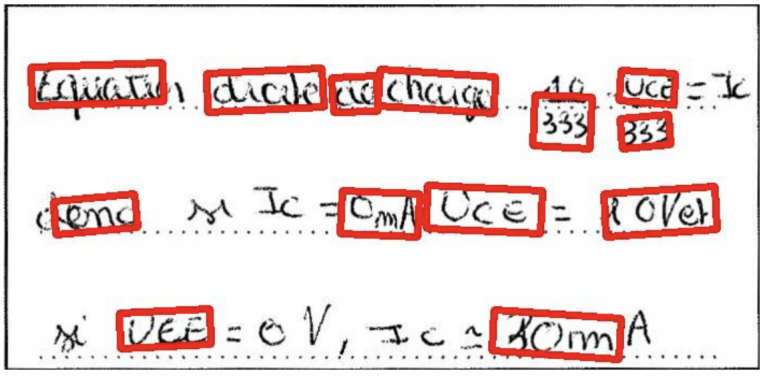

Figure 12. EAST Output example

After making a simple affine transformation of the input text box of the EAST model, we can use it as the next input.

Once the detection has been made with the EAST framework, the CRNN+CTC couple is used to

recognise the detected text.

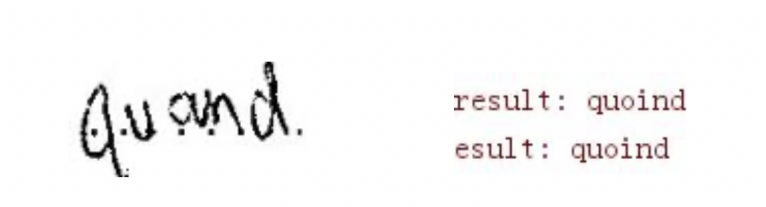

Figure 13. Result of CTC

As shown above, the CTC algorithm doesn't perform well. The specific reason is because the training data set is all generated by the algorithm, with a specific font and format, and handwriting is more difficult to recognize. Our next step is to use the handwriting data set to continue training the model.

5.2 Sentence similarity measurement

The Quora dataset contains 404350 sentence pairs with a 323480/80870 training/test split. Each pair is annotated with 0 or 1 corresponding to the ground truth set by a human reader. To enable our model to generalize beyond the vocabulary present in the Quora dataset, we provide the LSTM with inputs that reflect relationships between words beyond what can be inferred from the small number of training sentences. Like Mueller and Thyagarajan 2016 in their paper, our LSTM takes as input word-vectors which have been pre-trained on an external corpus. We use the 300-dimensional word2vec embeddings which (Mikolov, Chen, *et al.* 2013; Mikolov, Sutskever, *et al.* 2013) demonstrate can capture intricate inter-word relationships such as $vec(brother) - vec(man) + vec(woman) \approx vec(sister)$.

The MaLSTM predicts relatedness for a given pair of sentences via $g(h_{T_a}^{(a)}, h_{T_b}^{(b)})$, and we train the siamese network using backpropagation-through-time under root mean squarred-error (RMSE) loss function. Like Mueller and Thyagarajan 2016 our LSTM uses 50-dimensional hidden representations h_t and memory cells c_t. Optimization of the parameters is done using the Adam optimizer (Kingma and Ba 2015) along with with gradient clipping to avoid the exploding gradients problem (Pascanu, Mikolov, and Bengio 2013). We first initialize our LSTM weights with small random Gaussian entries. We train the MaLSTM model for seven (7) epochs. We obtain as measurement values on the test set the following values in the table 2:

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score
MaLSTM	77.953145	69.719073	71.140242	0.70

Table 2. Model validation

Our model realizes the state-of-the art accuracy and F1-score for semantic similarity task. For instance in validation phase, we obtain the following results in the table 3:

In the table 3, GT stands for Ground Truth. As we see in the table above, our model can easily find dissimilarities but stuck with similar sentences. In the next section, we will discuss further our results.

6 Discussion

This work demonstrate that text detection and text recognition is not a simple task mostly for handwritten text. In addition it demonstrates that a simple siamese LSTM is able of modeling complex semantics if the representations are explicitly guided. Despite the encouraging results we

Sentence 1	Sentence 2	GT	MaLSTM
How do I check who viewing your Facebook profile?	Can I see who looks at my Facebook profile?	1	0
What are the asymptotes of y=cotx?	What are the asymptotes of y=Tanx?	0	0
What is the difference between elastic constant and modulus of elasticity?	Is there any difference between the modulus of elasticity and the modulus of rigidity?	0	0
What do professional boxers and MMA fighters do one hour before a fight?	Who is stronger, a MMA fighter or a bodybuilder?	0	0
I'm a chubby girl and my face looks swollen. How can I slim my face?	My face is fat. How do I slim down my moon face?	1	0

Table 3. Sentences classification

have obtained, there is much to improve in our different pipelines for better results. As a result, we should try another methods for word-embedding such as those of (Li *et al.* 2015), especially as these word-vectors more comprehensively capture synonymity and entity-relationships. Moreover, it will be interesting to exploit the word2vec skip-gram model to build our word-embedding which will suits the best to our ask. The next step of our work should contains an in-depth study on other models for text detection.

References

Benajiba, Y., J. Sun, Y. Zhang, L. Jiang, Z. Weng, and O. Biran. 2019. "Siamese Networks for Semantic Pattern Similarity." *International Conference on Semantic Computing:* 191–194. doi:10.1109. ICSC.2019.00044.

Bhuvaneswari, M. S., S. Esakkiammal, J. A. vinisha, and S. U. Sankari. 2017. "Semantic Similarity Based Answer Sheet Evaluation Using NLP." *INTERNATIONAL JOURNAL FOR TRENDS IN ENGINEERING & TECHNOLOGY* 23:39–42. ISSN: 2349 – 9303. http://www.acadpubl.eu/hub/.

Chopra, S., R. Hadsell, and Y. LeCun. 2005. "Learning a similarity metric discriminatively, with application to face verification." In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on,* 1:539–546. IEEE.

Cui, Y., M. Jia, T. Lin, Y. Song, and S. Belongie. 2019. "Class-Balanced Loss Based on Effective Number of Samples." In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* 9260–9269.

Deng, D., H. Liu, X. Li, and D. Cai. 2018. "PixelLink: Detecting Scene Text via Instance Segmentation." *CoRR* abs/1801.01315. arXiv: 1801.01315. http://arxiv.org/abs/1801.01315.

Graves, A., S. Fernández, F. Gomez, and J. Schmidhuber. 2006. "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," (Pittsburgh, Pennsylvania, USA), ICML '06: 369–376. doi:10.1145/1143844.1143891. https://doi.org/10.1145/1143844.1143891.

Hochreiter, S., and J. Schmidhuber. 1997. "Long short-term memory." *Neural computation* 9 (8): 1735–1780.

Kingma, D. P., and J. Ba. 2015. "Adam: A Method for Stochastic Optimization." *CoRR* abs/1412.6980.

Lakshmi, V., and D. V. Ramesh. 2017. "EVALUATING STUDENTS' DESCRIPTIVE ANSWERS USING NATURAL LANGUAGE PROCESSING AND ARTIFICIAL NEURAL NETWORKS." *International Journal of Creative Research Thoughts* 5 (4): 3168–3173. ISSN: 2320-2882. http://www.ijcrt.org.

Li, Y., L. Xu, F. Tian, L. Jiang, X. Zhong, and E. Chen. 2015. "Word Embedding Revisited: A New Representation Learning and Explicit Matrix Factorization Perspective." In *Proceedings of the 24th International Conference on Artificial Intelligence,* 3650–3656. IJCAI'15. Buenos Aires, Argentina: AAAI Press. ISBN: 9781577357384.

Ma, J., W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue. 2018. "Arbitrary-Oriented Scene Text Detection via Rotation Proposals." *IEEE Transactions on Multimedia* 20 (11): 3111–3122.

Mikolov, T., K. Chen, G. S. Corrado, and J. Dean. 2013. "Efficient Estimation of Word Representations in Vector Space." http://arxiv.org/abs/1301.3781.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. 2013. "Distributed Representations of Words and Phrases and their Compositionality." Edited by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger: 3111–3119. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf.

Mueller, J., and A. Thyagarajan. 2016. "Siamese Recurrent Architectures for Learning Sentence Similarity." *Conference on Artificial Intelligence:* 2786–2792. http://www.aaai.org.

Pascanu, R., T. Mikolov, and Y. Bengio. 2013. "On the difficulty of training recurrent neural networks." In *Proceedings of the 30th International Conference on Machine Learning,* edited by S. Dasgupta and D. McAllester, 28:1310–1318. Proceedings of Machine Learning Research 3. Atlanta, Georgia, USA: PMLR, 17–19 Jun. http://proceedings.mlr.press/v28/pascanu13.html.

Patil, P., S. Patil, V. Miniyar, and A. Bandal. 2018. "Subjective Answer Evaluation Using Machine Learning." *International Journal of Pure and Applied Mathematics* 118 (24): 1–13. ISSN: 1314-3395. http://www.acadpubl.eu/hub/.

Ren, S., K. He, R. Girshick, and J. Sun. 2015. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.” Edited by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett: 91–99. http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf.

Shi, B., X. Bai, and C. Yao. 2017. “An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (11): 2298–2304.

Tian, Z., W. Huang, T. He, P. He, and Y. Qiao. 2016. “Detecting Text in Natural Image with Connectionist Text Proposal Network.” *CoRR* abs/1609.03605. arXiv: 1609.03605. http://arxiv.org/abs/1609.03605.

Zhou, X., C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang. 2017. “EAST: An Efficient and Accurate Scene Text Detector.” *Conference on Computer Vision and Pattern Recognition:* 5551–5560. ISSN: 1063-6919. doi:10.1109.CVPR.2017.283.

Zhou, Y., Q. Ye, Q. Qiu, and J. Jiao. 2017. “Oriented Response Networks.” In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 4961–4970.