Project#14: Emphasis Selection For Written Text in Visual Media

Vipul Singhal[1], **Sahil Dhull**[2], **Rishabh Agarwal**[3]
[1]160798, [2]160607, [3]160567
[1]EE, [2]CSE, [3]EE
{vipulsn,sahild,rish}@iitk.ac.in

Abstract

Text emphasis makes the text more powerful and is helpful in conveying the author's intent. We are studying the problem of selecting the words to emphasize in a short written text in an automated fashion.

We propose an end-to-end model that takes as input the text and corresponding to each word, gives a score describing the chances of the word to be emphasized. The dataset used is provided in SemEval 2020 Task 10, which consists of short text instances. Our goal will be to perform well in the competition.

1 Introduction

Visual communication relies on images and short texts like in flyers, posters, etc. The purpose of these is to convey the message effectively and without ambiguity. Moreover, they should be able to attract the reader's attention within the first few seconds. For text, this can be achieved by laying emphasis on particular words to convey the intent better. The task is to design automatic methods for emphasis selection, i.e., choosing candidates for emphasis in short written text, to enable automated design assistance in authoring.

Since the problem is a new one, not much has been tried on this task. The baseline paper (Shirani et al., 2019) employs an end-to-end label distribution learning (LDL) and predicts a selection distribution. The model consists of an embedding layer, followed by a BiLSTM layer and, in the end, a fully connected layer.

We plan on using other layers also instead of just BiLSTM like GRU, and using character embeddings along with word embeddings. Another approach will be using BERT embedding model with BRNN, attention, and CRF layer on top of it.

In section 2, we'll formally define the problem statement. Section 3 contains our proposed approach followed by Data Description, and in the end, our goals and timeline is mentioned.

2 Problem Definition

Given a sequence of words or tokens C = $\{x_1, x_2, ..., x_n\}$, we want to compute a score S_i for each x_i which indicates the degree of emphasis to be laid on the word.

The problem requires minimizing the loss function of neural network, which is an optimization problem.

3 Proposed Approach

We propose an end-to-end model which takes as input the words in the text and corresponding to each word, gives a score describing the degree of emphasis to be laid on the word. We plan to try at least two different types of sequence labeling model to learn emphasis patterns.

The first approach (Akhundov et al., 2018) involves character-level or byte-level embeddings of each word of a sentence computed using a BiLSTM layer, concatenated with word embeddings computed using pre-trained embeddings which is further passed through a pair of BiLSTM Layers. We will finally add a fully connected layer to predict a score for each word. The byte-level (or the character-level) embeddings will capture the morphological information of the words in the sentence, whereas word embeddings capture the semantic information of the words. Another variant of the above approach can be using GRU layer instead of BiLSTM layer (Yang et al., 2016).

The second approach (Emelyanov and Artemova, 2019) uses the BERT Language Model as embeddings with bidirectional recurrent network, attention, and CRF layer on the top. If the time and machines permit, we can even fine-tune the BERT Language Model.

4 Corpus/Data Description

The dataset taken is the dataset provided in SemEval-2020 Task-10: Emphasis Selection for Written Text in Visual Media. The dataset consists of 1,206 short text instances obtained from Adobe Spark. The dataset contains 7,550 tokens, and the average number of tokens per instance is 6.16, ranging from 2 to 25 tokens. On average, each instance contains 2.38 emphases, and the ratio of non-emphasis to emphasis tokens is 1.61.

Amazon Mechanical Turk was used, and nine annotators were asked to label each piece of text. The label distribution for each instance, which corresponds to the count per label normalized by the total number of annotations, was also computed. Finally, the data was split randomly into training(60%), development (10%), and test (30%) sets for further analysis.

5 Other applications (if applicable)

Applications like Adobe Spark perform automatic text layout in text and images, based on visual attributes like word length rather than semantics. The problem that we solve can be used to generate applications for automating the emphasis task based on meaning, to enable automated design assistance in authoring.

6 Goals and Timeline

We plan to make a website which, when given a plain text as input, generates the emphasized text as output.

1. First week - Reading related papers on sequence labeling and NLP models - All Group Members

2. By 15 Feb (before mid sem) - baseline paper implementation with tweaks (Sahil), 1st approach (Vipul) and 2nd approach (Rishabh)

3. Tweaking and improving the models in the above approaches (before mid sem break) - all group members (their corresponding models)

4. Upto 10 March - Participating in SemEval 2020 Task 10

5. March end - Making the website/web-app - all group members

6. Upto 25 April - Writing a research paper - all group members

References

Adnan Akhundov, Dietrich Trautmann, and Georg Groh. 2018. Sequence labeling: A practical approach. *arXiv preprint arXiv:1808.03926*.

Anton A Emelyanov and Ekaterina Artemova. 2019. Multilingual named entity recognition using pretrained embeddings, attention mechanism and ncrf. *arXiv preprint arXiv:1906.09978*.

Amirreza Shirani, Franck Dernoncourt, Paul Asente, Nedim Lipka, Seokhwan Kim, Jose Echevarria, and Thamar Solorio. 2019. Learning emphasis selection for written text in visual media from crowd-sourced label distributions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1167–1172.

Zhilin Yang, Ruslan Salakhutdinov, and William Cohen. 2016. Multi-task cross-lingual sequence tagging from scratch. *arXiv preprint arXiv:1603.06270*.