UIC-NLP at SemEval-2020 Task 10: Exploring an Alternate Perspective on Evaluation

Philip Hossu
University of Illinois at Chicago
Department of Mathematics,
Statistics, and Computer Science
phossu2@uic.edu

Natalie Parde
University of Illinois at Chicago
Department of Computer Science
parde@uic.edu

Abstract

In this work we describe and analyze a supervised learning system for word emphasis selection in phrases drawn from visual media as a part of the Semeval 2020 Shared Task 10. More specifically, we begin by briefly introducing the shared task problem and provide an analysis of interesting and relevant features present in the training dataset. We then introduce our LSTM-based model and describe its structure, input features, and limitations. Our model ultimately failed to beat the benchmark score, achieving an average $match()$ score of 0.704 on the validation data (0.659 on the test data) but predicted 84.8% of words correctly considering a 0.5 threshold. We conclude with a thorough analysis and discussion of erroneous predictions with many examples and visualizations.

1 Introduction and Task Background

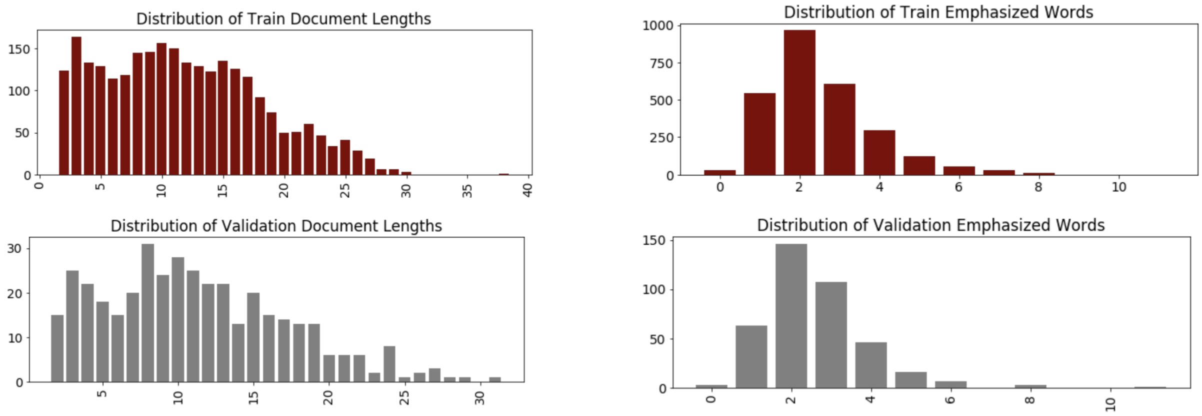
Short phrases are a frequent form of communication, particularly in visual media (e.g., advertisements and motivational quotes). It is common to see one or more of these words emphasized, whether it be through bolding, italicizing, or changing font size or color. Carefully selecting word emphasis can be key for effective advertising and clear messaging; thus, automating this task offers intriguing real-world potential.

In this shared task, researchers sought to develop systems which accurately emphasize words in variable length text phrases, drawn originally from various forms of visual media. While there have been similar past investigations into domain-specific emphasis selection models, this task was based on a diverse set of phrases, ranging from inspirational quotes (e.g., *Only in the* ***darkness*** *can you* ***see*** *the* ***stars***) to advertisements (e.g., ***Bastille*** *Day Block* ***Party***) and beyond. In all cases these phrases originated from some form of visual media, such as online graphics or printed posters/advertisements. The task is highly subjective—for the aforementioned "block party," one could argue that any number of emphasis options are appropriate, from "**Bastille Day** Block Party," to "**Bastille** Day **Block Party**," or maybe even "**Bastille Day Block Party**." This challenging, inherent subjectivity limits the performance of any model created for the task. This task is further described by Shirani et al. (2020).

2 Data Analysis

This task made use of a dataset obtained from Adobe Spark (https://spark.adobe.com). The provided training set contained 2742 unique phrases, comprising a total of 32400 words. These phrases were annotated manually by nine individuals who decided, for each word, whether the word should be emphasized. The individual labels were aggregated by the task organizers, with the final gold standard being provided as ratios of votes for emphasized over votes for not emphasized, ranging from $\{0.0, 0.11, ..., 1.0\}$. A similarly-labeled validation set containing 392 phrases (a total of 4385 words) was also provided. Before creating models, we analyzed the data to better understand the underlying trends and how they might impact model training and performance. Below we detail three interesting observations that directly impact model learnability.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1704–1709
Barcelona, Spain (Online), December 12, 2020.



(a) Document lengths.

(b) Emphasized words.

Figure 1: Distributions of training and validation set document lengths and number of emphasized words.

First, despite there being an average length of approximately 11.7 words per phrase[1] in the training set, there is only an average of 2.5 words emphasized per phrase.[2] We show the full distributions of (a) document, or phrase, lengths, and (b) emphasized words per phrase, in Figure 1. It can be observed that the distribution of phrase length is very different from that of the number of words emphasized, and the correlation coefficient between the two is weak (r=0.24). Therefore, we may expect models to struggle with outlier validation phrases containing many words to be emphasized, and perhaps long phrases in general. Distributions for the validation set, also shown in Figure 1, echo the trend seen with the training data (11.19 average phrase length vs. 2.59 average emphasis length).

Second, we noticed that the training set contains 85 duplicate phrases. This presented the opportunity for some interesting additional analyses; we focused specifically on those repeated exactly twice, which was the vast majority of the duplicates. We sought to find how often the annotators agreed with themselves between pairs of repeated phrases to provide further evidence regarding the subjectivity of the task. Absolute word-wise agreement – cases when all annotators voted each word the same way in a duplicate phrase – was very low, around 27%. We illustrate this in Table 1, showing an example of a duplicated phrase with only 2 of 11 labels matching. We ensured that all 85 duplicate phrases were removed from the training set before fitting our model parameters, reducing our training set to a relatively small size of 2657 phrases.

A third and final interesting observation lies in the distribution of the aggregated labels themselves. The mean of *nonzero* training labels is 0.366, and the overall mean of training labels is 0.284. The full distribution of gold standard labels across all words in the training and validation sets is shown in Table 2. From this table, we highlight a number of factors which may impede successful learning. Labels of 0.44, 0.55, and 0.66 can be fairly considered low confidence (approximately equal proportions of annotators did and did not think the word in question should be emphasized), and have a sum of 7342 – approx. 22% of the training and validation sets. Furthermore, we see a notable difference between the number of high-confidence positive (labels of 0.77 or above) training labels, 2597 (8%), versus high-confidence negative (labels of 0.33 or below) labels, 22461 (69%). Given this, we may expect that when our model disagrees with the validation labels, we would see more predictions that were too low than too high.

3 Model Design and Training Procedures

We developed and trained several models using Keras in Python 3. Generally, our goal was to create a stacked bidirectional LSTM structure and feed it a descriptive word embedding and additional information

[1] The task organizers tokenized the phrases such that punctuation marks were considered as separate words, as were individual constituents of contractions (e.g., “don’t” was separated into “do” and “’nt”).

[2] Here we denote a word to be emphasized if a simple majority of voters picked the positive class.

Table 1: Example of inconsistent labels on duplicate phrase.

WORD	PHRASE 135 LABEL	PHRASE 2425 LABEL
If	**0.11**	**0.00**
you	**0.22**	**0.11**
can	**0.33**	**0.22**
dream	**1.00**	**0.66**
it	**0.66**	**0.33**
,	**0.22**	**0.11**
you	0.22	0.22
can	**0.44**	**0.33**
do	0.66	0.66
it	**0.44**	**0.33**
.	**0.33**	**0.11**

Table 2: Distribution of true labels in training and validation sets.

LABEL	COUNT (TRAIN)	COUNT (VALIDATION)
0.0	7281	849
0.11	6600	952
0.22	4875	683
0.33	3705	468
0.44	2987	416
0.55	2442	335
0.66	1913	303
0.77	1436	199
0.88	836	142
1.0	325	38

for every input word to aid its learning capability. Our input features are described in detail below.

While we experimented with many different word embeddings, including Word2Vec (Mikolov et al., 2013) and ELMo (Peters et al., 2018), we observed the most consistently positive results using pre-trained 100-length GloVe (Pennington et al., 2014) embeddings. We introduced this information into the network using a Keras Embedding layer, where every missing word was given the average embedding of any word present in the three datasets (training, validation, and test).

In addition to the word embedding, a one-hot encoded part of speech (POS) tag was generated for each word using the StanfordPOSTagger (Toutanvoa and Manning, 2000) version from 2018-10-16. While the training and validation datasets came with provided POS tags, these tags were not provided for the test dataset. Therefore, we chose to overwrite all POS tags so there would be no issues with consistency.

The second supplemental feature was a vector of three scores corresponding to valence, arousal, and dominance (VAD) scores for each input word, to attempt to capture emotion-related attributes. We extracted this information from the sizeable NRC VAD Lexicon (Mohammad, 2018); we refer readers to the original paper for a detailed explanation of how these scores are determined. In short, *valence* corresponds to the positivity or negativity of a word, *arousal* corresponds to the word's intensity, and *dominance* refers to the amount of affective control the word exerts. Any words which were missing VAD scores were given an array of zeroes.

These three pieces of information (word embedding, POS vector, and VAD vector) were fed into a neural network utilizing the following structure: Concatenation Layer → Bidirectional LSTM Layer (out size 1024) → Bidirectional LSTM Layer (out size 1024) → Dense Layer (output size 38). The output layer contains 38 nodes corresponding with the maximum document length across the three datasets. We optimized the network using $logcosh$ (log cosine similarity) loss, linear activations, and a batch size of 1.

This structure was not difficult to implement, but the nature of Keras concatenation layers forced us to pad input sequences to a constant length (max phrase length) and predict 38 outputs for every phrase. This is likely one of the contributing factors limiting model performance. We also experimented with many other architectural variations, including introducing dropout layers, dimension and number of stacked LSTM layers, more dense layers at the end of the network, and as previously noted, modifying the word embedding type. Ultimately, the described setup yielded the most consistent, albeit low-ranking in the context of the shared task, results. We found that higher-dimensional word embeddings increased network training time and caused significant issues with overfitting.

4 Results and Error Analysis

The shared task evaluation metric was the $match(n)$ statistic, defined by Shirani et al. (2019). This metric considers the relative ranking of prediction labels, rather than the values themselves. Here n refers to the number of top predicted values to be compared – in other words, for $n = 2$, the metric looks at the two words having the highest gold standard scores for an instance, and computes their overlap with the two words having the highest predicted scores. Our model achieved an average $match$ score of 0.704 on the validation set, which shows that some patterns have been learned, while others remain unknown.

For our error analysis, we chose to consider an accuracy measurement that takes into account how these models would be practically deployed in a downstream application. We used a 0.5 threshold to signify a positive (> 0.5) or negative (< 0.5) emphasis label. Considering predictions for each of the 4385 individual words in the validation set, 3718 are a *0.5 match*; in other words, the validation label and prediction agree at a 0.5 threshold that the word should/shouldn't be emphasized. This 84.8% measurement is highly promising,[3] and leaves us with 667 errors to analyze. As we are treating the analysis from the scope of binary classification, we also computed other standard classification metrics, finding that in this context precision is 0.527, recall is 0.742, and F1 is 0.616. After analyzing approximately one-third of the errors manually, repetitive categories became quickly apparent. These categories are explained, with frequency percentages and examples, below.

Close misses on low-confidence labels made up a significant proportion of the errors present. We define this category to consist of incorrect predictions ranging between 0.35 and 0.65 where the true label was of low confidence (0.44, 0.55, or 0.66). As we expected from the analysis of the training data, in a vast majority of these misses, the model predicted too low rather than too high. This category accounts for approximately 33% of all misses. We chose to filter these errors from the remainder of our analysis so that subsequent error categories only include more descriptive misses (scaling the future categories up to a fraction of the 667 total).

Late-phrase misses were very common and are likely fueled by the wide distribution of phrase lengths and relatively low average number of emphasized words per phrase. These errors are defined to be a *0.5 match* threshold disagreement between the predicted label and validation label, where the word with the error is at index ≥ 10. This category accounted for around 17% of the total number of errors.

Abnormal sentence structure seemed to cause some difficulty for our model. While this category is somewhat hard to define, around 8% of our model's errors can be attributed to phrases which are poetic or sound like advertisements – not necessarily following the typical grammar rules or flow. An example of such a sentence is shown in Table 3.

Two final categories with a significant number of errors are *early-phrase* (word in position 0 through 9) and *late-phrase* (word in position ≥ 10) *sequence miscategorization.* These errors occurred when the

[3] As a brief aside, when we increase the threshold number for a word being emphasized or not, our model's performance continues to increase; however, we opted to use 0.5 as the most fair and logical cutoff.

Table 3: Errors were often seen in phrases displaying abnormal sentence structure.

WORD	TRUE LABEL	PREDICTION
Back	0.11	0.378
to	0.11	0.241
Life	**0.88**	**0.306**
•	0.00	0.102
•	0.00	0.166
•	0.00	0.340
Back	0.22	0.163
to	0.22	0.335
Reality	**1.0**	**0.483**

Table 4: Late-phrase sequences of emphasized words were often missed by the model.

POSITION	WORD	TRUE LABEL	PREDICTION
19	you	0.44	0.130
20	**do**	**0.55**	**0.111**
21	**not**	**0.66**	**0.262**
22	**want**	**0.77**	**0.215**
23	**to**	**0.66**	**0.302**
24	**know**	**0.66**	**0.111**
25	.	0.11	0.064

Table 5: Approximate percentages of errors per category.

CATEGORY	PERCENT
Close misses on low confidence labels	33%
Late-phrase misses	17%
Abnormal sentence structure	8%
Early-phrase sequence miscategorization	5%
Late-phrase sequence miscategorization	12%
Other	25%

model failed to predict multiple words correctly in a row. An interesting distinction is that the model struggled far more with these sequences late-phrase than early-phrase. An example is shown in Table 4.

The types defined above account for approximately 75% of the 667 errors, leaving 25% for the "other" category. There were several interesting, but rare, types of misses in this category. One such instance was the occasional out-of-vocabulary word like *TIC-TAC-TOE*, which was surely missing an embedding and VAD scores. We also noticed some errors for which the model struggled on words with many senses (like *light*), or periods in the middle of sentences, underscoring the utility of learned semantic and grammatical structure patterns for this task. The full break-down of approximate percentages for each error is shown in Table 5.

We end this section by reiterating some optimism regarding the model's performance. At this 0.5

cutoff threshold, the model predicted 84.8% of the words correctly. Increasing the threshold to 0.6 brings the accuracy up to 87.8%, and increasing it to 0.7 results in 91.3% correct predictions. Although the model fell short of baseline performance when measured using the $match(n)$ statistic, when considering the downstream task of actually selecting words to be emphasized, the model has clearly learned many accurate patterns. The task also remains subjective, leaving many "incorrect" predictions to not necessarily seem nonsensical (e.g., "**Jump** into **retirement**" (model) vs. "Jump into **retirement**" (true label)).

5 Conclusion

For this shared task, we sought to develop a system which accurately selects words to be emphasized in short phrases of varying length and subject matter. We used Keras to develop a stacked bidirectional LSTM network and trained it using pre-trained GloVe embeddings and additional, supplemental information including POS tags and VAD scores for every input word. Unfortunately, despite many iterations on design and inputs, our model fell short when compared to the baseline using the $match(n)$ metrics, ranking 22nd. Through our error analysis, we have identified a number of common patterns with which the model struggled, including late-phrase words, sequences of words, and abnormal/poetic sentence structure. We plan to introduce additional strategies for addressing those limitations in follow-up work, and hope that our analysis provides useful guidance for others pursuing this task as well.

Acknowledgements

We thank the anonymous reviewers for their helpful feedback.

References

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Saif Mohammad. 2018. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 English words. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 174–184, Melbourne, Australia, July. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June. Association for Computational Linguistics.

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, Florence, Italy, July. Association for Computational Linguistics.

Amirreza Shirani, Franck Dernoncourt, Nedim Lipka, Paul Asente, Jose Echevarria, and Thamar Solorio. 2020. Semeval-2020 task 10: Emphasis selection for written text in visual media. In *Proceedings of the 14th International Workshop on Semantic Evaluation.*

Kristina Toutanvoa and Christopher D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 63–70, Hong Kong, China, October. Association for Computational Linguistics.