SemEval-2020 Task 10:
Emphasis Selection for Written Text in Visual Media

Amirreza Shirani[†], **Franck Dernoncourt**[‡], **Nedim Lipka**[‡],
Paul Asente[‡], **Jose Echevarria**[‡], **and Thamar Solorio**[†]
[†]University of Houston [‡]Adobe Research
[†]`{ashirani,tsolorio}@uh.edu`
[‡]`{franck.dernoncourt,lipka,asente,echevarr}@adobe.com`

Abstract

In this paper, we present the main findings and compare the results of SemEval-2020 Task 10, Emphasis Selection for Written Text in Visual Media. The goal of this shared task is to design automatic methods for emphasis selection, i.e. choosing candidates for emphasis in textual content to enable automated design assistance in authoring. The main focus is on short text instances for social media, with a variety of examples, from social media posts to inspirational quotes. Participants were asked to model emphasis using plain text with no additional context from the user or other design considerations. SemEval-2020 Emphasis Selection shared task attracted 197 participants in the early phase and a total of 31 teams made submissions to this task. The highest-ranked submission achieved 0.823 Match$_m$ score. The analysis of systems submitted to the task indicates that BERT and RoBERTa were the most common choice of pre-trained models used, and part of speech tag (POS) was the most useful feature. Full results can be found on the task's website[1].

1 Introduction

In visual communication, emphasis is an intentional focus on one or more components to create a main focal point or center of interest with the composition. It has been found that it only takes the human eye 50 milliseconds to form an opinion on a visual composition (Lindgaard et al., 2006). Therefore, it is important for a visual element to deliver a clear message by calling attention to specific information. Whether on flyers, posters, ads, social media posts or motivational messages, emphasis is usually designed to grab a viewer's attention by being distinct from the rest of the design elements.

Thanks to various online platforms, a massive amount of digital text is being generated by users every day. These media are filled with content competing for attention and are usually highly designed to engage viewers' attention to convey their message in the most efficient way. For textual content, word emphasis is used as a powerful tool to better convey the desired meaning of the written text to the audience. Utilizing emphasis techniques can potentially add another dimension to the text through visualization. Emphasis on textual content can be done with colors, backgrounds, or fonts, or with styles like italic and boldface to clarify or even change the meaning of a sentence by drawing attention to some specific information. Figure 1a shows an example that is aesthetically appealing but fails to effectively communicate its intent. Understanding the text would allow the system to propose a different layout that emphasizes words that contribute more to the communication of the intent, as shown in Figure 1b.

In the last few years, we have observed many significant improvements in various platforms for generating, formatting, and editing digital text. For example, some graphic design applications such as Adobe Spark[2] perform automatic text emphasis using templates that include images and text with different design effects. However, the used layout algorithms are often inflexible in that they rigidly emphasize words based on the visual attributes (e.g., word length) of those words, rather than the semantics of the text. As a result, the outcome may fail to accurately communicate the meaning of the written text, resulting in

[1] `https://competitions.codalab.org/competitions/20815`

[2] `https://spark.adobe.com`

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1360–1370
Barcelona, Spain (Online), December 12, 2020.

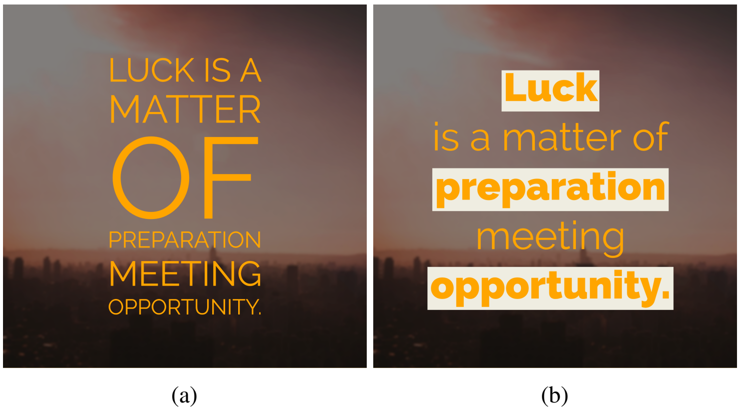


Figure 1: Two different text layouts emphasizing different parts of the sentence.

unintended emphasis and the wrong message to the audience. However, an emphasis selection model can potentially make better suggestions by having a better understanding of the input text.

Task Characteristics This emphasis selection task poses new challenges associated with the nature of the task: (1) No additional context from the user or the rest of the design such as background image is provided. Therefore the proposed task requires a computational understanding of the written text. (2) The dataset contains very short texts, usually fewer than ten words. Generally, working with short text instances is challenging since the decision needs to be made by only considering a few words. (3) Word emphasis patterns are author- and domain-specific, therefore, without knowing the author's intent and only considering the input text, multiple emphasis selections are valid. However, a good model should be able to capture the inter-subjectivity or common sense within given annotations and finally label words according to higher agreements.

Expected Impact of the Task The ultimate goal is to enable design assistance for authors by suggesting words that are good candidates to emphasize. The typical applications of this task include, but are not limited to, creating flyers, posters, drawings, advertisements and other visual material one may find online and across social media platforms such as Pinterest, Instagram and Snapchat. Moreover, emphasis selection models have applications in many design programs such as Adobe Spark, Apache OpenOffice Impress, GIMP, or Microsoft PowerPoint.

SemEval Emphasis Selection Task In this shared task, we invited research on a novel Natural Language Processing (NLP) task that represents unique algorithmic and modeling challenges due to its nature. We observed a diverse and interesting set of solutions to tackle the existing challenges from a large number of participants, both from academia and industry. As part of this shared task, we released a dataset annotated with word emphases, which served as a benchmark to compare various techniques. Furthermore, we expect the task to be interesting for researchers studying relevant tasks such as machine-human interaction, reading comprehension, graphic design and user experience. In the following sections, we describe the setup, participation, results, and more importantly, the insights gained from the task.

1.1 Task Definition

Given a sequence of tokens $C = \{x_1, ..., x_n\}$, a real number $y_i \in [0, 1]$ needs to be assigned for each token in the sequence, indicating the degree to which the token needs to be emphasized. In other words, we define the emphasis score y_i as the probability or weight of the i^{th} token in the sequence. Finally, during the evaluation, the final set of emphases are generated by selecting tokens with the highest values (described in Section 5).

2 Related Work

We firstly introduced and formulated the task of emphasis selection in (Shirani et al., 2019) in which an end-to-end label distribution learning (LDL) model in a sequence tagging architecture is proposed to model emphasis. We evaluated the model against different baselines on the Spark dataset (introduced in Section 3).

Keyword or key-phrase detection may be the closest topic to emphasis selection. Keywords can capture the main topics described in a given document (Turney, 2002). Modeling keywords or key-phrases has been widely addressed in different domains such as news articles (Wan et al., 2007), scientific publications (Nguyen and Kan, 2007) and Twitter data (Zhang et al., 2016; Bellaachia and Al-Dhelaan, 2012). Keyword detection mainly focuses on finding important nouns or noun phrases (Augenstein et al., 2017). In contrast, emphasis could be applied to a subset of words with different roles in a sentence. Generally, word emphasis may use to express emotions, show contrast, capture a reader's interest or clarify a message. Moreover, emphasis selection in social media posts deals with very short texts and the prediction needs to be made based on a single instance.

In the context of expressive prosody generation, emphasis has been addressed based on acoustic and prosodic features that exist in spoken data. For example, (Nakajima et al., 2014) predicted emphasized accent phrases from advertisement text information and (Mass et al., 2018) modeled word emphasis on audience-addressed speeches.

3 Data Collection

The data used for this shared task is the integration of two datasets from different sources, which are created from scratch based on texts collected from the Adobe Spark and Wisdom Quotes website. The dataset used for this task can be found in the task's data repository[3]. The following are the descriptions of the two datasets. The **Spark dataset** is collection of 1,195 instances from Adobe Spark[4]. It contains a variety of subjects featured in flyers, posters, advertisements or motivational memes on social media. The **Quotes dataset** is a collection of quotes from well-known authors collected from Wisdom Quotes[5] with 2,681 instances.

Table 1 provides details about the length of instances in the datasets. The Emphasis dataset with 3,876 instances, consists of 44,976 words and 4,886 unique words. We used Amazon Mechanical Turk and asked nine annotators to label each piece of text. More precisely, we asked annotators to select word(s) in the given text that should be emphasized. Having nine annotators gives us this ability to capture different viewpoints, each focusing on different parts of the sentence. Figure 2 shows an example of text annotated with nine annotations. In this example, there is more consensus in emphasizing words like "inspiration" and "Genius". On the other hand, words like "is" and "percent" are not good candidates based on general agreement. To ensure high-quality annotation, we included carefully-designed quality questions in 10 percent of the hits. Moreover, we only allowed master annotators to participate.

The data is split up randomly between training, development and test sets. A training data set of 2,741 instances, development set of 392 instances, and test set of 743 instances were released to the participants. Fleiss' Kappa score (Shrout and Fleiss, 1979) of 24.60 was observed on the data set. Such a Kappa score indicates the existence of multiple perspectives about emphasis in the dataset.

Table 1: Dataset Statistics

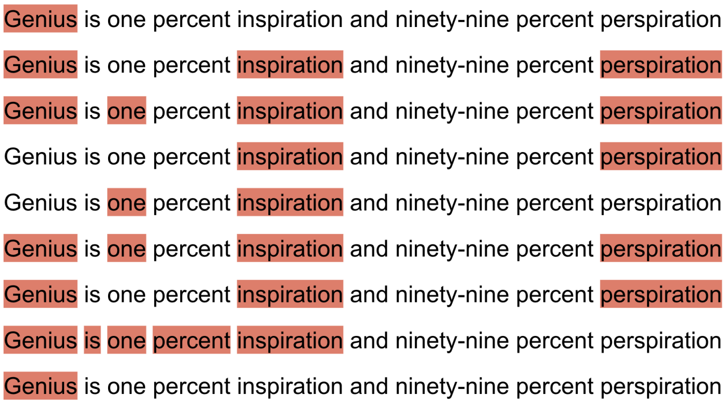
Dataset/Text length	Max	Avg.	SD
Spark dataset	29	6.24	4.64
Quotes dataset	38	13.99	5.47
Emphasis dataset (combined)	38	11.60	6.33

Table 2 shows an example of a short text annotated with the BIO annotations. As it is shown, words such as "Best" are selected more often for emphasis than other words in the sequence. First, we compute the label distribution for each instance, which corresponds to the count per label normalized by the total number of annotations (shown in "Norm. Freq. column"). Then we compute *Emphasis Probabilities*

[3] https://github.com/RiTUAL-UH/SemEval2020_Task10_Emphasis_Selection

[4] https://spark.adobe.com

[5] http://wisdomquotes.com



Genius is one percent inspiration and ninety-nine percent perspiration

Genius is one percent inspiration and ninety-nine percent perspiration

Genius is one percent inspiration and ninety-nine percent perspiration

Genius is one percent inspiration and ninety-nine percent perspiration

Genius is one percent inspiration and ninety-nine percent perspiration

Genius is one percent inspiration and ninety-nine percent perspiration

Genius is one percent inspiration and ninety-nine percent perspiration

Genius is one percent inspiration and ninety-nine percent perspiration

Genius is one percent inspiration and ninety-nine percent perspiration

Figure 2: A short text example from the Quotes dataset along with its nine annotations.

for all the words in the sequence. The final evaluation is against ground truth emphasis probabilities (explained in Section 5).

Table 2: An example from the Spark dataset along with its nine annotations. In this table,"B/I"s and "O"s represent emphasis and non-emphasis words respectively. "B"s indicate the beginning and "I"s indicate the inside of emphasis. "Freq." and "Norm. Freq." columns show the normal and normalized values for label frequencies respectively.

Words	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**A8**	**A9**	**Freq. [B,I,O]**	**Norm. Freq. [B,I,O]**	**Emphasis Probs [B+I]**
Tag	B	O	B	O	B	O	O	O	B	[4,0,5]	[0.44,0,0.55]	[0.44]
Your	O	O	O	O	O	O	O	O	O	[0,0,9]	[0,0,1]	[0]
Best	O	B	B	B	B	B	O	B	B	[7,0,2]	[0.77,0,0.22]	[0.77]
Friends	O	I	O	O	O	I	B	O	I	[1,3,5]	[0.11,0.33,0.55]	[0.44]

4 Data Analysis

Many systems reported performance gain by using Part of Speech Tags (POS) tags in their models. In this section, we analyze the effectiveness of this feature by closely examining the top 20 POS tags in our dataset. We used the Stanford Part-Of-Speech Tagger (Toutanova et al., 2003) to obtain POS tags for all tokens in our dataset. We divide the emphasis probabilities to four intervals (0-0.25, 0.25-0.50, 0.50-0.75 and 0.75-1.00) and compute how the POS tags are distributed in these four intervals.

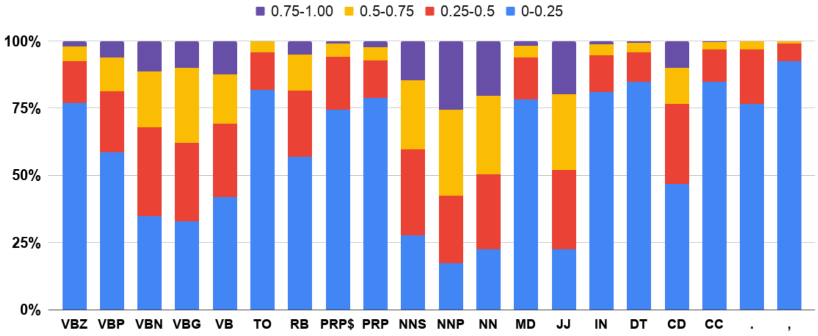


Figure 3: Frequencies of the top 20 POS tags in 0-0.25, 0.25-0.5, 0.5-0.75, 0.75-1.00 intervals of emphasis probabilities. The vertical values correspond to the percentage of tag counts over the total number of words in training set.

Figure 3 shows the occurrence of the top 20 POS tags in four emphasis probability intervals for all token labels in our training set. POS tags like ",", ".", "DT" and "PRP" are more favored to have low emphasis probabilities (0-0.25). Interestingly, words with the highest probabilities (0.75-1.00) are usually from “NNP”, “NN” and “JJ” word types. As we expected, there are some general trends for emphasized words with respect to the type of words in sentences, which make POS tags a useful feature for modeling emphasis.

5 Evaluation Metric

The evaluation was performed on the test set. Participants were asked to provide a real value (greater and equal to zero) for each token in the test set that indicates the probability of the token being emphasized. All models were evaluated with Match$_m$ metric and ranked based on the averaged values of scores for m=$1, 2, 3, 4$.

Match$_m$ For each instance x in the test set D_{test}, we selected a set $S_m^{(x)}$ of $m \in \{1 \ldots 4\}$ words with the top m probabilities according to the ground truth. Analogously, we selected a prediction set $\hat{S}_m^{(x)}$ for each m, based on the predicted probabilities. We defined the metric Match$_m$ as follows:

$$\text{Match}_m := \frac{\sum_{x \in D_{test}} |S_m^{(x)} \cap \hat{S}_m^{(x)}| / m}{|D_{test}|}$$

Finally, we computed the average value of Match$_m$ for all $m \in \{1 \ldots 4\}$ and ranked the submitted systems based on this averaged value (*RANK*). To better handle word duplicates, the computation is based on the position of words in a sentence rather than the actual words. Note that there were many cases where two or more tokens have the exact same probability. In this case, if the model predicts either one of the labels, we considered it as a correct answer. Table 5 shows some examples form the dataset, illustrating how the metric is computed.

6 Baseline Model

We provided a baseline model for this task. This model (DL-BiLSTM-ELMo) is a sequence-labeling model that essentially utilizes ELMo contextualized embeddings (Peters et al., 2018) as well as two BiLSTM layers to label emphasis. During the training phase, the Kullback-Leibler Divergence (KL-DIV) (Kullback and Leibler, 1951) is used as the loss function. More analysis and the complete description of this model is provided in (Shirani et al., 2019).

7 Systems and Results

This task attracted 197 participants and a total of 31 teams made submissions to this task. The teams that submitted papers for the SemEval-2020 proceedings are listed in Table 3. In total, 25 teams performed higher than the baseline and six teams performed lower. 13 of the 31 teams also submitted their system description papers.

The base models used in the task submissions ranged from ELMo (Peters et al., 2018), BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019), to state-of-the-art pre-trained models such as ERNIE 2.0 (Sun et al., 2019), XLM-RoBERTa (Conneau et al., 2019), T5 (Raffel et al., 2019) and ALBERT (Lan et al., 2019). Figure 4 shows different pre-trained models used in this task. Among them, BERT and RoBERTa were used most often. Ensemble transformer-based models were one of the most popular approaches (26% of submissions). All submissions applied deep neural network techniques to model emphasis. Moreover, some teams did explore hand-crafted features, such as part-of-speech tags, named entities, valence, arousal, dominance scores to enhance the performance of their models.

7.1 Top Systems

The results for each of four scores, as well as the *RANK* score, are shown in Table 3. The top-3 teams based on the *RANK* score are *ERNIE* (Huang et al., 2020), *Hitachi* (Morio et al., 2020), *IITK* (Singhal et al., 2020). The top-performing team, *ERNIE*, achieved the highest Match$_m$ score of 0.823, 0.009 points higher than the second team and 0.013 points higher than the third team. *ERNIE*, achieved the highest

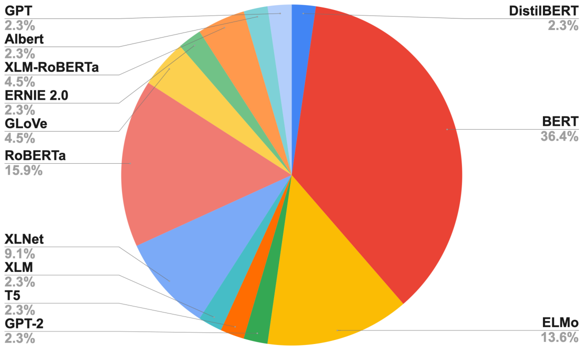


Figure 4: Pie chart showing the pre-trained models used in this task.

score not only in *RANK* score but across all four scores. The next system on our leader board is *Hitachi*, with a score of 0.814. And finally, *IITK*, by achieving 0.810 *RANK*, stands in third place.

7.2 Best Paper Awards

Our shared task awarded several best paper distinctions to complement the top performing systems. Here are the categories of best papers and the winners for each:

- Best system description paper: *IDS* (Shin et al., 2020), this paper, with interesting analysis components, advances our understanding regarding the effectiveness of pre-trained language models for this specific task.

- Best result interpretation paper: *MIDAS* (Anand et al., 2020), the authors go the extra mile to analyze the results in this paper.

- Best negative results paper: *UIC-NLP* (Hossu and Parde, 2020), the authors performed extensive experiments with non-contextualized pre-trained models as well as a variety of hand-crafted features. Through the error analysis, the authors identified a number of common challenging patterns for the model, including late-phrase words, sequences of words, and abnormal/poetic sentence structure.

7.3 Top Performing Systems and Novel Architectures

In this section, we provide a brief description of the best performing and novel approaches. Table 4 shows a high level summary of these systems.

ERNIE achieved the highest score by fine-tuning ERNIE 2.0 as the base model. They also reported high performance by using other pre-trained models like XLM-RoBERTa, RoBERTa and ALBERT. They further boosted the model by utilizing data augmentation and hand-crafted features like word capitalization and the occurrence of hashtags in instances.

Hitachi tackled the task by combining rich contextualized embeddings and fine-tuning seven Pre-trained Language Models (PLMs) on the task such as BERT, GPT-2 (Radford et al., 2019), RoBERTa, XLM-RoBERTa, XLNet (Yang et al., 2019), XLM (Lample and Conneau, 2019), and T5. In addition, they added POS tags and token embeddings from a character-level LSTM layer. They introduced a distribution fusion system to fuse the output distributions of the fine-tuned models and find the optimal hyperparameter set. They showed the performance gain of the fusion model over average ensemble as well as individual PLMs. Among all PLMs, BERT and XLNet models were more successful in predicting emphasis individually.

IITK, the team ranking in third place, proposed an ensemble model where the base models were BERT, RoBERTa, and XLNet. In order to aggregate the outputs, they computed the average of the scores predicted by these models. The authors also provided different baselines from the character-level BiLSTM model with attention to transformer-based models like XLM-RoBERTa, ALBERT and GPT-2. When comparing all individual models, XLNet-Large performed the best.

Table 3: List of teams that participated in SemEval-2020 Task 10 with their ranks and scores.

#	Rank	Team Name	*RANK* Score	Score 1	Score 2	Score 3	Score 4
1	1	ERNIE	**0.823**	**0.724** (1)	**0.819** (1)	**0.862** (1)	**0.887** (1)
2	2	Hitachi	0.814	0.715 (2)	0.811 (3)	0.851 (6)	0.880 (3)
3	3	IITK	0.81	0.694 (5)	0.812 (2)	0.854 (3)	0.879 (4)
4	4	Randomseed19	0.805	0.677 (8)	0.803 (4)	0.858 (2)	0.881 (2)
5	5	Sherry	0.799	0.677 (8)	0.799 (5)	0.850 (7)	0.870 (7)
6	5	Sattiy	0.799	0.676 (9)	0.797 (6)	0.850 (7)	0.871 (6)
7	6	Unixlong	0.798	0.674 (10)	0.797 (6)	0.852 (5)	0.868 (8)
8	6	Giftai	0.798	0.674 (10)	0.793 (7)	0.853 (4)	0.870 (7)
9	7	FPAI	0.796	0.690 (7)	0.780 (10)	0.840 (9)	0.873 (5)
10	7	PAER	0.796	0.699 (3)	0.782 (8)	0.833 (12)	0.870 (7)
11	8	YYYY	0.795	0.694 (5)	0.780 (10)	0.835 (10)	0.871 (6)
12	9	L2020	0.794	0.696 (4)	0.781 (9)	0.831 (13)	0.868 (8)
13	10	BugHunter	0.785	0.654 (13)	0.775 (12)	0.845 (8)	0.867 (9)
14	11	Amobee	0.783	0.692 (6)	0.766 (16)	0.822 (17)	0.853 (17)
15	11	MIDAS	0.783	0.650 (16)	0.780 (10)	0.834 (11)	0.868 (8)
16	12	procyon	0.781	0.661 (11)	0.777 (11)	0.830 (14)	0.859 (12)
17	12	new_bill__	0.781	0.655 (12)	0.774 (13)	0.834 (11)	0.862 (11)
18	12	Jupyter	0.781	0.653 (14)	0.772 (14)	0.835 (10)	0.863 (10)
19	13	CrazyRock	0.774	0.651 (15)	0.766 (16)	0.824 (15)	0.857 (14)
20	14	CLP	0.772	0.642 (17)	0.763 (17)	0.823 (16)	0.858 (13)
21	15	TextLearner	0.767	0.627 (18)	0.769 (15)	0.823 (16)	0.850 (19)
22	16	Bright	0.758	0.627 (18)	0.749 (19)	0.809 (20)	0.848 (21)
23	17	LAST	0.756	0.610 (20)	0.749 (19)	0.812 (19)	0.853 (17)
24	18	AP	0.754	0.612 (19)	0.741 (20)	0.806 (22)	0.855 (16)
25	18	Tëxtmarkers	0.754	0.596 (22)	0.752 (18)	0.815 (18)	0.851 (18)
26	19	Baseline	0.75	0.608 (21)	0.737 (21)	0.807 (21)	0.849 (20)
27	20	EL-BERT	0.745	0.591 (23)	0.726 (22)	0.807 (21)	0.856 (15)
28	21	David3	0.705	0.514 (24)	0.693 (23)	0.784 (23)	0.830 (22)
29	22	SRIB_OnDeviceAI	0.659	0.479 (26)	0.640 (24)	0.732 (24)	0.785 (23)
30	22	UIC-NLP	0.659	0.490 (25)	0.638 (25)	0.731 (25)	0.775 (24)
31	23	IDS	0.629	0.425 (27)	0.618 (26)	0.716 (26)	0.756 (25)
32	24	YNU-HPCC	0.401	0.231 (28)	0.373 (27)	0.467 (27)	0.534 (26)

A wide range of novel methods were used to model emphasis. For example, *FPAI* (Guo et al., 2020) converted the task of emphasis selection to a simplified query-based machine reading comprehension (MRC) task, where the goal was to answer the fixed query, "Find candidates for emphasis".

To tackle the low inter-annotator agreement in the dataset, *Tëxtmarkers* (Glocker and Markianos Wright, 2020) attempted to model multiple annotators jointly by adapting a crowd layer architecture (Rodrigues and Pereira, 2018), introducing initialization with agreement dependent noise. The crowd layer is intended to help the model to outperform a baseline trained with token level majority voting.

IDS (Shin et al., 2020) performed an interesting analysis of pre-trained models to investigate whether PLMs contain enough knowledge to select proper words for emphasis. They compared different zero-shot models in which self-attention distributions of PLMs were used to emphasize words. More precisely, the authors investigated individual attention heads of different models like BERT, DistilBERT, GPT-2, RoBERTa, XLNet, and XLM to probe their ability to identify emphasis without any fine-tuning. Their interesting findings indicate that DistilBERT is more successful in predicting emphasis while XLNet and GPT-2 perform poorly when there is no training for this task.

The top non-transformer-based model, *Procyon* (ranked 12th), successfully proposed an ELMo-based

Table 4: Main features in participating systems. An 'N/A' in the **Ref.** column means that we did not receive a system description paper for that entry.

Rank	Team Name	Ref.	Best performing Methods
1	ERNIE	(Huang et al., 2020)	ERNIE 2.0 + data augmentation + hand-crafted features
2	Hitachi	(Morio et al., 2020)	Distribution fusion of BERT, GPT-2, RoBERTa, XLM-RoBERTa, XLNet, XLM, T5 + POS tags and token embeddings
3	IITK	(Singhal et al., 2020)	Ensemble of BERT, RoBERTa and XLNet
4	Randomseed19	(Shatilov et al., 2020)	Ensemble of BERT, RoBERTa and XLNet
5	Sherry	N/A	Ensemble of Bert, XLM-RoBERTa and RoBERTa + POS tags for adjusting final outputs
5	Sattiy	N/A	Aggregation of BERT, XLNet and GPT (formulated the task as a 10-class classification task)
7	FPAI	(Guo et al., 2020)	RoBERTa-base (converted the task to a query-based machine reading comprehension task)
10	BugHunter	N/A	LSTM model using BERT, POS tags and tf-idf features (multi-task framework with classification and distribution training objectives)
11	Amobee	N/A	Combination of BERT and RoBERTa
11	MIDAS	(Anand et al., 2020)	Ensemble of RoBERTa, BERT and XLNet
12	Procyon	N/A	ELMo (multi-modal model based on word representations and POS tags)
12	Jupyter	N/A	BERT + sentiment values such as pleasantness, attention, sensitivity, aptitude and polarity
13	CrazyRock	N/A	BERT + 4 parallel Bi-GRU + 2 layers of self attention
14	CLP	N/A	BERT + BiLSTM + hand-crafted features (listwise ranking model)
15	TextLearner	(Yang et al., 2020)	RoBERTa (two-staged ranking model)
16	Bright	N/A	BERT and ELMo (Graph neural networks)
17	LAST	(Bestgen, 2020)	BERT and ELMo with LightGBM
18	AP	N/A	ELMo with BiLSTM
18	Tëxtmarkers	(Glocker and Markianos Wright, 2020)	BERT, GRU, attention + crowd layer
20	EL-BERT	(Kanani et al., 2020)	ELMo + POS + SentBERT + WordBERT
22	UIC-NLP	(Hossu and Parde, 2020)	GloVe, LSTM, POS and valence, arousal, dominance (VAD) scores
23	IDS	(Shin et al., 2020)	BERT, DistilBERT, GPT-2, RoBERTa, XLNet, and XLM (using self-attention distributions of PLMs in a zero-shot setting)
24	YNU-HPCC	(Liao et al., 2020)	ELMo and BERT (multi-granularity ordinal classification model)

multi-modal model with two sub-networks to learn emphasis scores based on word representations and POS tags separately.

7.4 Discussion

To have a better understanding of the challenges of this task, we perform an error analysis to examine where the models succeed and in what situations they face difficulties in selecting emphasis words. More specifically, we compute the average Match$_m$ score over all 31 submissions for every example in the test set and examine the challenging cases for all models. Table 5 shows some interesting examples from the test set with three Match$_m$ scores ($m1$–$m3$) from all submissions, where $m1$ stands for the average score for system predictions obtained by selecting the top word, and $m3$ stands for results from selecting the top 3. In many cases, selecting emphasis words was unchallenging for most of the systems (e.g., S1 in Table 5 with "Imagination" as the top word and "rules" and "world" with same emphasis probability.). In some examples, there is no single token standing out in the sentence, so it was not easy to select one single word with certainty. S2 is a good example with low $m1$ and high $m2$ and $m3$, indicating disagreement between models and annotators for choosing the first word with the highest probability. We also observed many cases where one word clearly stands out of the sentence but it is not clear which words should be selected next. S3 is an example of this where most systems were able to select the top word "talked" correctly, but faced difficulties in predicting other words for that sentence.

There are some cases where prediction is easy for humans but still poses challenges for models. For example, most annotators agreed on selecting "basketball" with the highest probability in S4; however, many models failed to select this word in the top position, probably due to the unusual structure of the sentence. In this example, "East", "Sleep" and "Watch" have equal probabilities in the annotation.

Table 5: Examples from the test set with averaged Match$_m$ scores across all submitted systems. Words with high emphasis probability labels are shown in bold.

Num	Sentence	Match$_m$
S1-1	**Imagination** rules the world.	$m1$ = 0.9354
S1-2	**Imagination rules** the **world**.	$m2$ = 1.0
S1-3	**Imagination rules** the **world**.	$m3$ = 1.0
S2-1	All successes begin with self-discipline. It starts with **you**.	$m1$ = 0.0322
S2-2	All successes begin with **self-discipline**. It starts with **you**.	$m2$ = 0.8225
S2-3	All **successes** begin with **self-discipline**. It **starts** with **you**.	$m3$ = 0.9677
S3-1	I learned most, not from those who taught me but from those who **talked** with me.	$m1$ = 0.8387
S3-2	I learned most, not from those who taught me but from those who **talked with** me.	$m2$ = 0.5645
S3-3	I learned most, not from those who taught me but from those who **talked with me**.	$m3$ = 0.4516
S4-1	Eat . Sleep . Watch **Basketball** . Repeat .	$m1$ = 0.1290
S4-2	**Eat** . **Sleep** . **Watch Basketball** . Repeat .	$m2$ = 0.6935
S4-3	**Eat** . **Sleep** . **Watch Basketball** . Repeat .	$m3$ = 0.7419

8 Conclusion

This paper summarizes the insights gained from organizing Task 10 at SemEval-2020. Given a short piece of text, the task consists of selecting candidate words to emphasize. We received a good number of system submissions, with 13 teams submitting a system description paper. While there were many differences between individual systems, we observed a strong trend favoring the use of transformer based models as key ingredient in the proposed architectures. Many description papers present valuable analyses of the data and task. We encourage readers interested in this task to take a careful look at these papers for additional inspiration on how to improve results further.

References

Sarthak Anand, Pradyumna Gupta, Hemant Yadav, Debanjan Mahata, Rakesh Gosangi, Raymond Zhang, and Rajiv Ratn Shah. 2020. MIDAS at SemEval-2020 task 10: Emphasis selection using label distribution learning and contextual embeddings. In *Proceedings of the 14th International Workshop on Semantic Evaluation*.

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications. *arXiv preprint arXiv:1704.02853*.

Abdelghani Bellaachia and Mohammed Al-Dhelaan. 2012. NE-rank: A novel graph-based keyphrase extraction in Twitter. In *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 372–379. IEEE.

Yves Bestgen. 2020. LAST at SemEval-2020 Task 10: Finding tokens to emphasise in short written texts with precomputed embedding models and LightGBM. In *Proceedings of the 14th International Workshop on Semantic Evaluation*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Kevin Glocker and Stefanos Andreas Markianos Wright. 2020. Tëxtmarkers at SemEval-2020 Task 10: Emphasis selection with agreement dependent crowd layers. In *Proceedings of the 14th International Workshop on Semantic Evaluation*.

Chenyang Guo, Xiaolong Hou, Junsong Ren, Lianxin Jiang, Yang Mo, Haiqin Yang, and Jianping Shen. 2020. FPAI at SemEval-2020 Task 10: A query enhanced model with RoBERTa for emphasis selection. In *Proceedings of the 14th International Workshop on Semantic Evaluation*.

Philip Hossu and Natalie Parde. 2020. UIC-NLP at SemEval-2020 task 10: Exploring an alternate perspective on evaluation. In *Proceedings of the 14th International Workshop on Semantic Evaluation*.

Zhengjie Huang, Shikun Feng, Weiyue Su, Xuyi Chen, Shuohuan Wang, Jiaxiang Liu, Xuan Ouyang, and Yu Sun. 2020. ERNIE at SemEval-2020 Task 10: Learning word emphasis selection by pre-trained language model. In *Proceedings of the 14th International Workshop on Semantic Evaluation*.

Chandresh S. Kanani, Sriparna Saha, and Pushpak Bhattacharyya. 2020. EL-BERT at SemEval-2020 Task 10: A multi-embedding ensemble based approach for emphasis selection in visual media. In *Proceedings of the 14th International Workshop on Semantic Evaluation*.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Dawei Liao, Jin Wang, and Xuejie Zhang. 2020. YNU-HPCC at SemEval-2020 task 10: Using a multi-granularity ordinal classification of the bilstm model for emphasis selection. In *Proceedings of the 14th International Workshop on Semantic Evaluation*.

Gitte Lindgaard, Gary Fernandes, Cathy Dudek, and Judith Brown. 2006. Attention web designers: You have 50 milliseconds to make a good first impression! *Behaviour & Information Technology*, 25(2):115–126.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yosi Mass, Slava Shechtman, Moran Mordechay, Ron Hoory, Oren Sar Shalom, Guy Lev, and David Konopnicki. 2018. Word emphasis prediction for expressive text to speech. In *Interspeech*, pages 2868–2872.

Gaku Morio, Terufumi Morishita, Hiroaki Ozaki, and Toshinori Miyoshi. 2020. Hitachi at SemEval-2020 Task 10: Emphasis distribution fusion on fine-tuned language models. In *Proceedings of the 14th International Workshop on Semantic Evaluation*.

Hideharu Nakajima, Hideyuki Mizuno, and Sumitaka Sakauchi. 2014. Emphasized accent phrase prediction from text for advertisement text-to-speech synthesis. In *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computing*, pages 170–177.

Thuy Dung Nguyen and Min-Yen Kan. 2007. Keyphrase extraction in scientific publications. In *International Conference on Asian Digital Libraries*, pages 317–326. Springer.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Filipe Rodrigues and Francisco C Pereira. 2018. Deep learning from crowds. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Aleksandr Shatilov, Denis Gordeev, and Alexey Rey. 2020. randomseed19 at SemEval-2020 Task 10: Emphasis selection for written text in visual media. In *Proceedings of the 14th International Workshop on Semantic Evaluation*.

Jaeyoul Shin, Taeuk Kim, and Sang-goo Lee. 2020. IDS at SemEval-2020 task 10: Does pre-trained language model know what to emphasize? In *Proceedings of the 14th International Workshop on Semantic Evaluation*.

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.

Patrick E Shrout and Joseph L Fleiss. 1979. Intraclass correlations: uses in assessing rater reliability. *Psychological Bulletin*, 86(2):420.

Vipul Singhal, Sahil Dhull, Rishabh Agarwal, and Ashutosh Modi. 2020. IITK at SemEval-2020 Task 10: Transformers for emphasis selection. In *Proceedings of the 14th International Workshop on Semantic Evaluation*.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2019. ERNIE 2.0: A continual pre-training framework for language understanding. *arXiv preprint arXiv:1907.12412*.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology–Volume 1*, pages 173–180. Association for Computational Linguistics.

Peter D Turney. 2002. Learning to extract keyphrases from text. *arXiv preprint cs/0212013*.

Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. 2007. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 552–559.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, pages 5754–5764.

Zhishen Yang, Lars Wolfsteller, and Naoaki Okazaki. 2020. TextLearner at SemEval-2020 Task 10: A contextualized ranking system in solving emphasis selection in text. In *Proceedings of the 14th International Workshop on Semantic Evaluation*.

Qi Zhang, Yang Wang, Yeyun Gong, and Xuanjing Huang. 2016. Keyphrase extraction using deep recurrent neural networks on Twitter. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 836–845.