WMD at SemEval-2020 Tasks 7 and 11: Assessing humor and propaganda using Unsupervised Data Augmentation

Guillaume Daval-Frerot
Paris, France
gdavalfrerot@gmail.com

Yannick Weis
Karlsruhe, Germany
yannick.weis@freenet.de

Abstract

In this work, we combine the state-of-the-art BERT architecture with the semi-supervised learning technique UDA in order to exploit unlabeled raw data to assess humor and detect propaganda in the tasks 7 and 11 of the SemEval-2020 competition. The use of UDA shows promising results with a systematic improvement of the performances over the four different subtasks, and even outperforms supervised learning with the additional labels of the Funlines dataset.

1 Introduction

In times when information is able to circulate freely via the internet with limited to non-existing quality control, organized groups and governments are using various tools of propaganda to spread miss-information and push their own agenda to achieve financial and political gains (Miller, 1939). With millions of press articles written every month, it became unrealistic for humans to perform any form of regulation. Machine Learning (ML) can help with the automatic processing of large volumes of data, but paradoxically also requires a significant amount of examples in order to apprehend high-level concepts such as humor or propaganda. More and more solutions to this problem are emerging, for example Transfer Learning (Pan and Yang, 2010) which allows the use of already existing resources to increase the performance of a given task. It can be achieved by using a larger and closely related dataset, or simply by fine-tuning a pre-trained model.

A recent neural architecture proposed for natural language processing in (Devlin et al., 2018) and achieving state-of-the-art results in numerous tasks is called Bidirectional Encoder Representations from Transformers (BERT). One of the reasons for its popularity is the availability of models pre-trained with large resources in different languages. It is based on the combination of bidirectional Long Short-Term Memory (LSTM) networks, described in (Hochreiter and Schmidhuber, 1997) and Attention Mechanisms, described in (Bahdanau et al., 2015). Both techniques have been discussed and considered as promising subjects of study in previous works (Daval-Frerot et al., 2018) on sentiment analysis during the SemEval-2018 challenge.

Another solution to deal with the data collection is to exploit raw data with semi-supervised approaches. Both humor assessment in Task 7 (Hossain et al., 2020a) and propaganda detection in Task 11 (Da San Martino et al., 2020) have the advantage of a wide availability of samples in online press articles. By avoiding the use of labels, a lot more resources can be used through techniques like Unsupervised Data Augmentation (UDA) proposed in (Xie et al., 2019). The idea is to draw similarities between labeled and unlabeled data during a semi-supervised training, making the overwhelming millions of un-regulated articles into a strength as an exploitable resource.

In this paper, we propose to combine the two solutions along with other practices with the objective of using the bare minimum of human-prepared resources while simultaneously improving the performances.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1865–1874
Barcelona, Spain (Online), December 12, 2020.

2 Task descriptions

2.1 Task 7: Assessing Humor in Edited News Headlines

The purpose of the task, described in (Hossain et al., 2020a) is to estimate the funniness of micro-edited headlines, where a micro-edit is defined as one of the following word-wise replacements: entity to noun, noun to noun, and verb to verb. Each member of a jury, consisting of at least five people, assigns a grade to the headlines. The grade is an integer from 0 to 3 where 0 means ”not funny”, 1 means ”slightly funny”, 2 means ”moderately funny” and 3 means ”funny”. All the scores for a given modified headline are then averaged to obtain a unique real value. However, each of the individual scores are also available in the official dataset described in (Hossain et al., 2019). The task itself is split into two subtasks to which we refer as 7-1 and 7-2.

The first subtask is a regression based approach with the aim of directly predicting the averaged funniness of the edited headlines. Two alternatives exist using classification, since the prediction should only take a small, finite number of values: the system could be trained to only predict the few possible scores, or even simply to learn the preferences of each judge. However, the number of judges can vary and the individual scores are ordered by value and not by person, making these two approaches less straightforward. We decided to stick to the regression as the task was thought.

The second subtask is a ternary comparison between two modifications of the same headline in order to find the funniest. One micro-edit can be considered more fun, less fun or equally fun to another, but the last possibility was not considered for the evaluation. Once again there are a few ways to tackle the problem:

- use the system previously trained for task 7-1 to predict a value for the two edited headlines as independent samples and then compare the predictions.
- train a similar system to predict a value different from the given scores. As the grades are the average of at least five different values, the distribution tends to be denser around the mean score (as shown in Figure 2), in addition to the naturally unbalanced observed density. A more uniform distribution could be obtained for example by using an isotonic regression, making the comparison more robust for originally close values.
- train a new system taking the two modified headlines as input and predicting explicitly which sample is funnier. The system should be constrained to be symmetrical, with an architecture inspired for example by siamese networks (Bromley et al., 1993).

We chose the first proposition in order to focus on subtask 7-1, as it was advantageous for both subtasks. In addition, it allowed to considerably reduce the number of trainings, and so the overall computational resources required.

2.2 Task 11: Detection of Propaganda Techniques in News Articles

The task described in (Da San Martino et al., 2020) deals with the automatic detection of propaganda in news articles, and was proposed in a slightly different version in a previous workshop named NLP4IF (Da San Martino et al., 2019a). It is composed of two complementary subtasks, namely Span Identification and Technique Classification. Although they are called SI and TC by the organisers, we kept the format defined with task 7 and refer to them as 11-1 for SI, and 11-2 for TC.

The first subtask is a segmentation of news articles to detect propaganda. The fact that the labels are sub-sequences inside sentences was constraining for the pipeline, for example with the choice of augmentations. For the same reason, it was also the most computationally expensive subtask. The system was trained on a word-wise level as it allowed more freedom, such as replacing words or simply modifying them without altering the ground truth. It also reduced the time required for trainings. More details about the nature of the samples and labels are given in (Da San Martino et al., 2019b).

The second subtask is a classification of the previously detected segments into one of the 18 propaganda techniques, regrouped into 14 classes listed in Table 2. As several segments with different labels can be present in some sentences, the inputs given to our system were strictly composed of the tokens belonging to one specific segment, and not including for example the surrounding words or sentences.

3 System overview

The submission is based on the Unsupervised Data Augmentation (UDA) technique described in (Xie et al., 2019) applied to a pre-trained BERT architecture (Devlin et al., 2018) in order to use external, unlabeled resources in addition to the official datasets. The pipeline is composed of the following steps, with more details for most of them in specific subsections:

1. **Pre-processing**: the samples are tokenized using the `nltk` package described in (Loper and Bird, 2002). In the case of subtask 11-1, the articles are also split into sentences.
2. **Augmentation**: more samples are generated using various kinds of modifications over both labeled and unlabeled datasets.
3. **Feature extraction**: the submitted system works on subword-level embeddings used with the BERT architecture, but other experiments were also conducted on word-level embeddings with GloVe (Pennington et al., 2014).
4. **Semi-supervised training**: the augmentations are used following the UDA technique to re-train an existing BERT model where the last layer was adapted to the specific subtasks.
5. **Ensemble**: the training step is repeated several times with small variations to produce different instances to aggregate them into an ensemble.
6. **Post-processing**: the predictions are processed to match the required submission format. In the case of subtask 11-1, the predictions are smoothed to avoid micro gaps between words or other tokens.

The aim of the overall pipeline was to keep out manually labeled data as much as possible, in opposition to previous works described in (Daval-Frerot et al., 2018), although it is a source of quality features. It concerns mostly the use of lexicons, as they are often specific to languages or tasks and require a lot of human time. It was preferred to rely only on unsupervised feature extraction techniques such as GloVe.

3.1 Data augmentations

A common practice in supervised learning is to apply augmentations to the collected samples in order to increase their number and their diversity, resulting in better performances as shown in (Simard et al., 1998). An efficient augmentation is a modification which should at least respect the two following purposes:

- introduce more information: the new sample should be different enough from the original sample in order to carry additional knowledge about the task. It can be done by discarding useless information, or drawing similarities with other samples.
- keep the data realistic: the new sample should follow the same format as the rest of the data, and be plausible under the scope of the task. The label should preferably stay the same, or at least change in a predictable way.

	Examples
Originals	1. *"The larger-than-usual outbreak had helped spread the bacteria that causes the plague."* 2. *"The icy cigar was probably knocked askew by a violent collision with another object."* 3. *"The big frigid blue ball is the farthest planet from the Sun and it hasn't been studied as much."*
Back-translations (French)	1. *"The epidemic, larger than usual, has contributed to the spread of the bacteria responsible for the plague."* 2. *"The frozen cigar was probably knocked over by a violent collision with another object."* 3. *"The large icy blue ball is the most distant planet from the Sun and it has not been studied as much."*
Back-translations (German)	1. *"The above-average outbreak had helped spread the bacteria that caused the plague."* 2. *"The icy cigar was probably lopsided by a violent collision with another object."* 3. *"The big cold blue ball is the most distant planet from the Sun and has not been examined as much."*
Synonym replacements	1. *"The larger-than-usual irruption had facilitated diffuse the bacteria that stimulate the plague."* 2. *"The cold cigar was credibly bumped askew by a vehement hit with another object."* 3. *"The big frigid bluing glob is the farthest satellite from the Sun and it hasn't been analysed as much."*
TF-IDF replacements	1. *"An larger-than-usual outbreak will helped spread as bacteria in causes spin plague."* 2. *"Barbecuing icy cigar was as knocked askew to herald insides collision the another object."* 3. *"No big frigid blue mustache is by farthest planet from and Sun or airhorn hasn't from studied grand much."*

Table 1: Three sentences selected from task 11 dataset with examples of samples generated using augmentations.

As the two objectives are partially opposed, it is important to find the right balance. A relevant approach used in (Xie et al., 2019) is to mix several techniques with different degrees of respect for one or the other. By following this idea we decided to explore three different augmentations:

- **Back-translation**: used for example in (Sennrich et al., 2015), it consists in translating a sentence into another language before translating it back to the source language. The semantic is generally preserved and the new sentence is still grammatically correct, while adding diverse formulations and expressions to the dataset.
- **Synonym replacement**: another way to reformulate a sentence is to swap the nouns and verbs with their synonyms. The effect is similar to the back-translation, with the disadvantage that synonyms are applied to words while back-translation can change the whole structure of the sentence. However, this technique is more accessible for automatic large-scale processing, both in term of resources and computation.
- **TF-IDF replacement**: the idea is to replace the least important words of the sentence with other unimportant words, with regard to their use in the corpus through their TF-IDF score. The generated sentences are no longer grammatically correct, resulting in noisy sentences and forcing the system to filter out non-essential information.

The use of augmentations can improve the system performance, as mentioned previously, but it is important to point out that the new samples are close to their originals. This redundancy becomes limiting when increasing the amount of generated samples (Simard et al., 1998). The point is often to avoid collecting more data, which can be consuming both in time and resources. In the present case, the raw data consisting of press articles and titles is widely available in a large amount. The expensive part is to manually annotate the samples, resulting in restricted datasets. In that specific case, the approach described in (Xie et al., 2019) called UDA allows to make use of that unlabelled raw data through semi-supervised learning. Their proposition is based on consistency training methods, used to produce systems more robust to noisy inputs (Bachman et al., 2014). Augmentations can by definition be used to produce small variations in the samples, and actually proved to be efficient to enforce consistency. One method in particular is to not consider the original and augmented samples as independent: while the prediction obtained from the original sample should correspond to its label, the prediction from the augmented sample should simply be close to the prediction from the original sample. The model M is trained with a loss function L which can be formalized as:

$$L(M|\mathbb{X},\mathbb{Y}) = \sum_{x,y\in\mathbb{X}\mathbb{Y}} L_1(M(x),y) + \sum_{x,\tilde{x}\in\mathbb{X}\tilde{\mathbb{X}}} L_2(M(\tilde{x}),M(x)) \tag{1}$$

- $\mathbb{X}$, $\tilde{\mathbb{X}}$, $\mathbb{Y}$ the sets containing the original samples, the augmented samples and the labels.
- $\mathbb{X}\mathbb{Y}$ the set containing the relevant pairs of original samples and labels.
- $\mathbb{X}\tilde{\mathbb{X}}$ the set containing the relevant pairs of original and augmented samples.
- L_1 the loss applied over a prediction from an original sample and its label.
- L_2 the loss applied over predictions from an original sample and one of its augmented versions.

The important point in (Xie et al., 2019) approach is that the second term in equation (1) does not depend on the labels, meaning it can be applied over unlabeled data. The difference is that without the first term to guide the model, no relevant convergence is guaranteed over the unlabeled part of the dataset. However, a significant gain in performances has been observed, which is interpreted as a ”spread” of the labels: by similarity with labeled samples, some unlabeled samples obtained labels by induction which then spread even further, with sometimes a surprising low label-to-sample ratio.

3.2 Model ensembles

The idea of ensembles is widely used in machine learning, and not only with neural networks (Opitz and Maclin, 1999). One of the methods consists in using different instances of a given architecture by combining the predicted output. It can lead to scores better than any of the single instances mainly due to the following reasons.

The first reason is the variability between instances. For neural networks, the non-convexity of the problem implies different solutions depending on the initial state of the weights. Several other factors can introduce even more variability, such as data augmentation since most techniques rely on random modifications. Also in our context, semi-supervised learning through UDA is prone to more diverse solutions.

The second reason comes from the practice of selecting the best epoch during the training by splitting the dataset into training and validation sets. The latter should be large enough to estimate the general performance of the system, which means sacrificing a significant part of the data. In addition it also leads to more variability. Using several instances allows to cover the entire dataset while still selecting the best epochs with enough data.

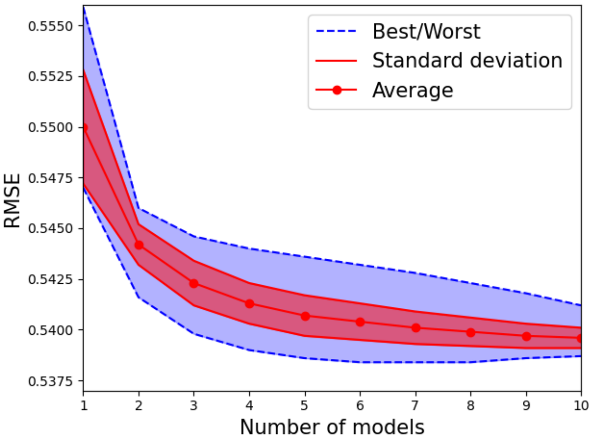

Figure 1: Scores for task 7-1 development set using UDA with different sizes of LSTM ensembles. All of the $\binom{n}{k}$ combinations were tested with size $k \in [\![1, 10]\!]$ and $n = 15$ different instances.

3.3 Metric optimizations

In task 11 the metric chosen for the evaluation is the F_1 score, and more precisely micro-average F_1 score in the case of the 11-2 subtask. This choice is common for language processing even though it can be questioned based on several criteria (Ferri et al., 2009). This is not the aim of this part, although some points require a special attention.

First of all, the F_1 score is defined as a harmonic mean of the precision and recall, so in the end defined by the output probabilities through true positives tp, false positives fp and false negatives fn as:

$$F_1 = \frac{2tp}{2tp + fp + fn} \tag{2}$$

Since all of those parameters are dependent on the chosen classification threshold, it is possible to find different scores from the same predicted probabilities. The two approaches to solve this problem according to Lipton (2014) are:

- directly use the F_1 score as a loss when fitting the model to the dataset.
- estimate the optimal threshold during training with various techniques.

Technique	n	%	Σ%
Loaded language	2123	34.6	34.6
Name calling, Labeling	1058	17.3	51.9
Repetition	621	10.1	62.0
Doubt	493	8.1	70.1
Exaggeration, Minimisation	466	7.6	77.7
Appeal to fear/prejudice	294	4.8	82.5
Flag-waving	229	3.7	86.2
Causal oversimplification	209	3.4	89.6
Appeal to authority	144	2.4	92.0
Slogans	129	2.1	94.1
Whataboutism, Straw men, Red herring	108	1.8	95.9
Black-and-White fallacy	107	1.7	97.6
Thought-terminating cliches	76	1.2	98.8
Bandwagon, Reductio ad hitlerum	72	1.2	100.0

Table 2: Number of occurrences per technique in the task 11 training dataset in decreasing order with percentage and cumulative percentage.

The same paper demonstrates that the optimal threshold is half the maximum F_1 under the assumption that the model output is a well-calibrated distribution. Once again different methods exist to either enforce this condition when training the model, or transform the output into such a distribution (Zadrozny and Elkan, 2002). However, to keep the implementation simple we chose to do a grid search of the best threshold during the validation step to both learn the threshold and avoid selecting the wrong best epoch because of the mentioned F_1 variability.

Secondly, the micro-average F_1 score is dominated by common labels in the case of unbalanced classes (Lipton et al., 2014). The distribution in the training dataset is shown in Table 2 in the case of task 11-2. The most represented class is *Loaded language* with 34.6%, which is more than the 10 least represented propaganda techniques. It could be advantageous to simply ignore a few techniques.

4 Experimental setup

4.1 Datasets

The datasets provided for the task 7 and 11 are respectively described in (Hossain et al., 2019) and (Da San Martino et al., 2019b), and will not be detailed much. One dataset is provided for each subtask, but note that dataset 7-2 only consists of ternary comparisons of some samples from dataset 7-1, which is composed of 9652 original and modified headlines. The dataset 11-1 consists of 371 articles associated with 5468 segments as labels, corresponding to the input segments of dataset 11-2 to classify into one of the 14 classes.

An additional 8248 samples with labels were provided by the organizers of task 7, released as the Funlines[1] dataset described in (Hossain et al., 2020b). This is technically an external source, as the methods and people in charge may slightly differ from the official dataset. We conducted a few experiments with it, both as additional data with label and using UDA without labels.

Another external dataset was used for task 11 without labels through semi-supervised learning. Originally available on Kaggle, we refer to it as All-the-news-articles[2] dataset. It consists of around 204k articles with headlines, but only up to 1000 randomly chosen articles were used per training for technical reasons. For task 7 we preferred to apply UDA with the Funlines dataset only, as it would require additional steps to generate fun modifications of the titles from All-the-news-articles dataset.

4.2 Trainings

The implementation used for BERT is available in the `transformers`[3] package described in (Wolf et al., 2019), with different pre-trained versions. We used fine-tuning on the basic version and refer it as BERT$_{fine}$. We also proposed to train one from scratch, but to avoid overfitting we empirically chose to reduce the size to 2 layers of 64 neurons each. We refer to it as BERT$_{small}$, and added a similar bidirectional LSTM network for comparison. The three architectures can be conveniently adapted to regression and sentence/token classification.

All of the networks were trained using Adam with a learning rate of 1×10^{-3} during 15 epochs from scratch, and 1×10^{-5} during 5 epochs for fine-tuning. The main losses are MSE for task 7 and BCE for task 11. The secondary losses used with UDA are MSE for task 7 and the Kullback-Leibler divergence for task 11. The validation sets were composed of a random $1/5^{th}$ of the available data, and consequently we chose to use ensembles of 5 instances.

Although the back-translation is recommended in (Xie et al., 2019), we chose to avoid it since we could not find any official open-source package for proper translations, and because task 11-1 required to keep the same number of tokens in a similar order to keep the labels relevant. Thus for both regular augmentations and UDA we augmented all samples twice with TF-IDF and synonym replacements.

4.3 Other models

More classic techniques have been tested to assess the quality of the propositions, namely Support-Vector Machine (SVM), Gradient Boosting (GB) and Random Forest (RF).

The input features were obtained by averaging GloVe (word-level) embeddings over either the samples for tasks 7 and 11-2 or a 5-word long sliding window for task 11-1. The use of ensembles was not considered relevant since the first one solves a convex problem, and the other two already are aggregates of models.

The `scikit-learn`[4] library, described in (Pedregosa et al., 2011), has been used for all those models using default arguments for both instantiations and trainings, as the goal is to propose a reproducible baseline.

[1] `https://www.funlines.co/humor`

[2] `https://components.one/datasets/all-the-news-articles-dataset/`

[3] `https://huggingface.co`

[4] `https://scikit-learn.org`

5 Results

5.1 Analysis

The curves in Figure 1 are showing the evolution of the scores over the subtask 7-1 development set with an increasing size of the ensembles. The average RMSE score of all the possible combinations is monotonously decreasing with a cumulated improvement of 0.01 going from $k = 1$ (single models) to $k = 10$ (10 models per ensemble). Around 60% of the gain is made between $k = 1$ and $k = 2$ with a better score than the best single model alone. However, the best score represented by the lower dotted curve is flat from $k = 6$ to $k = 8$ and starts to increase after that, suggesting that the average performances may be reduced beyond $k = 10$.

Each dot in Figure 2 represents a prediction, and is partially transparent to give an idea of the local density. The perfect ideal system would result in dots perfectly centered around $f(x) = x$, but here the linear regression is $f(x) = 0.23x + 0.76$ and the dots are widely spread around it. No prediction is made outside of the range $[0.3, 2.0]$. Note that the performance associated to those predictions is an RMSE of 0.524 while the best candidate achieved a score of 0.513 over the development set.

The experiments with scores shown in Table 3 aim to determine the best use of the Funlines dataset in task 7. Note when reading the table that the RMSE should be minimized while accuracy should be maximized. The results obtained using the samples without the labels with UDA (B) are systematically better than the two other cases. Also the results when using both samples and labels through a regular training (C) are worse than the case where the Funlines dataset is not used at all (A) in a quite significant way in subtask 7-1 while only slightly worse in subtask 7-2.

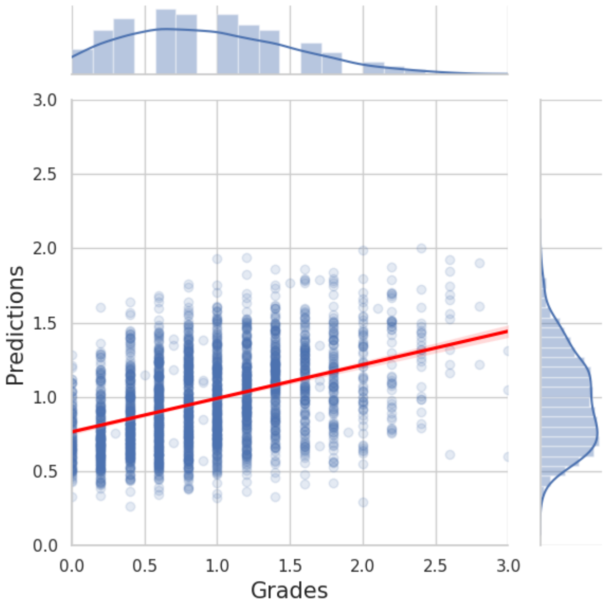

Figure 2: Distribution of the predictions against the expected grades for task 7-1 development set, and the fitted linear regression. The data density for the predictions and expected grades are respectively shown to the right and on top.

The main scores over all subtasks are gathered in Table 4. Note once again that the RMSE should be minimized while all other metrics should be maximized. Among the proposed techniques, UDA is systematically better than the common use of augmentations (Aug.) and no data augmentation (Base) for a given model. The common augmentation even appear to be detrimental in the case of subtask 7-2 with BERT$_{small}$. Among the proposed systems, BERT$_{fine}$ is better in most cases, followed by LSTM, then BERT$_{small}$ and finally the baselines.

5.2 Discussion

The previous observations allow to draw a few conclusions in the present case. The most interesting one in regard to our personal objectives concerns the Funlines dataset. The use of semi-supervised learning allowed to achieve better scores by using quality samples, even when the associated labels were not relevant to detrimental. However, the poor distribution of the predictions shown in Figure 2, close to the best results obtained during the competition, suggests that the task objective is far from being achieved.

	Funlines		7-1 (RMSE)			7-2 (Acc)		
	Samples	Labels	LSTM	BERT$_{small}$	BERT$_{fine}$	LSTM	BERT$_{small}$	BERT$_{fine}$
A)	No	No	0.548	0.554	0.538	0.589	0.579	0.604
B)	Yes	No	0.541	0.552	0.524	0.606	0.599	0.620
C)	Yes	Yes	0.562	0.566	0.549	0.588	0.576	0.602

Table 3: Results with different uses of the Funlines dataset in subtasks 7-1 and 7-2. A) corresponds to trainings using only the official dataset, B) to trainings additionally using the Funlines dataset without labels through UDA, and C) to trainings using the union of the official and Funlines datasets as regular data.

		RF	GB	SVM	LSTM			BERT$_{small}$			BERT$_{fine}$		
					Base	Aug.	UDA	Base	Aug.	UDA	Base	Aug.	UDA
7-1	(RMSE)	0.581	0.559	0.553	0.548	0.545	0.541	0.554	0.562	0.552	0.538	0.528	0.524
7-2	(Acc)	0.559	0.589	0.605	0.589	0.596	0.606	0.588	0.579	0.599	0.604	0.618	0.620
11-1	(F_1)	0.218	0.259	0.261	0.304	0.320	0.341	0.307	0.318	0.336	0.319	0.349	0.363
11-2	(μF_1)	0.415	0.472	0.448	0.501	0.510	0.513	0.481	0.486	0.508	0.502	0.516	0.523

Table 4: Scores of the different systems over the development sets of the four subtasks, using the official evaluation metrics: Root Mean Square Error (RMSE), Accuracy (Acc), F_1 score (F_1), micro-average F_1 score (μF_1).

Moreover, the labels from the Funlines dataset are gathered using methods similar to the official dataset, raising questions about the observed difference and suggesting perhaps subjectivity of humor is quite impacting. The approach proposed in subsection 2.1, consisting in estimating the humor of each judge, may have been interesting to study. On the same page, the subtask 7-2 could have benefited from one of the more specific solutions proposed, as it was shown in Figure 2 solving a regression problem seems quite inefficient to establish comparisons even though the 0.620% accuracy is close to the 0.649% best candidate accuracy.

On the contrary, the results obtained for subtask 11-1 are significantly lower than the best achieved with an F_1 score of 0.363 compared to 0.534. The consistency between our proposed baseline and the multiple tests suggests some missing processing steps in our pipeline. In that case our different propositions would likely still be relevant to enhance stronger systems if no similar solutions were already used. Indeed, the largest gains in performances using UDA are observed on the submissions for subtasks 11-1 and 11-2, the latter achieving correct results in regard to the competition. The impact of the amount of external resources was not studied because of technical limitations, but is an objective in the near future.

6 Conclusion

In this competition we decided to highlight a few practices that could benefit many domains at lower cost. The performances in the various tasks covered by this article have all been improved by the use of semi-supervised learning with UDA, and in some cases even more than supervised learning with additional, closely related, external resources. The availability of systems pre-trained with large resources such as BERT also enhanced the performances through transfer learning. The common point between these two techniques is to turn incomplete or unrelated resources into knowledge for the neural network architectures.

The different propositions were discussed and most of them have room for improvements. The limited computational resources available constrained the experiments, as for example just 2.4% of the external datasets found were actually used. Moreover, in an effort to keep the pipeline simple, a few steps discussed in (Xie et al., 2019) were avoided but could be used. In general, more specific approaches of each task will be subject to further research.

References

Philip Bachman, Ouais Alsharif, and Doina Precup. 2014. Learning with pseudo-ensembles. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3365–3373. Curran Associates, Inc.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Jane Bromley, J.W. Bentz, Leon Bottou, I. Guyon, Yann Lecun, C. Moore, Eduard Sackinger, and R. Shah. 1993. Signature verification using a siamese time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(4), August.

Giovanni Da San Martino, Alberto Barrón-Cedeño, and Preslav Nakov. 2019a. Findings of the NLP4IF-2019 shared task on fine-grained propaganda detection. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 162–170, Hong Kong, China, 10. Association for Computational Linguistics.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019b. Fine-grained analysis of propaganda in news articles. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*, EMNLP-IJCNLP 2019, Hong Kong, China, November.

Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. SemEval-2020 task 11: Detection of propaganda techniques in news articles. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, SemEval 2020, Barcelona, Spain.

Guillaume Daval-Frerot, Abdessalam Bouchekif, and Anatole Moreau. 2018. Epita at semeval-2018 task 1: Sentiment analysis using transfer learning approach. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 151–155.

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*.

Cèsar Ferri, Jose Hernandez-Orallo, and R. Modroiu. 2009. An experimental comparison of performance measures for classification. *Pattern Recognition Letters*, 30:27–38, 01.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Nabil Hossain, John Krumm, and Michael Gamon. 2019. "president vows to cut <taxes> hair": Dataset and analysis of creative text editing for humorous headlines. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 133–142, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Nabil Hossain, John Krumm, Michael Gamon, and Henry Kautz. 2020a. Semeval-2020 Task 7: Assessing humor in edited news headlines. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2020)*, Barcelona, Spain.

Nabil Hossain, John Krumm, Tanvir Sajed, and Henry Kautz. 2020b. Stimulating creativity with funlines: A case study of humor generation in headlines. In *Proceedings of ACL 2020, System Demonstrations*, Seattle, Washington, July. Association for Computational Linguistics.

Z Chase Lipton, Charles Elkan, and Balakrishnan Narayanaswamy. 2014. Thresholding classifiers to maximize F1 score. *Machine Learning and Knowledge Discovery in Databases*, 8725:225–239.

Edward Loper and Steven Bird. 2002. Nltk: the natural language toolkit. *arXiv preprint cs/0205028*.

C.R. Miller. 1939. *How to detect and analyze propaganda ...: An address delivered at Town hall, Monday, February 20, 1939*. Town Hall pamphlet. Town Hall, Inc.

David Opitz and Richard Maclin. 1999. Popular ensemble methods: An empirical study. *Journal of artificial intelligence research*, 11:169–198.

S. J. Pan and Q. Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *CoRR*, abs/1511.06709.

Patrice Y. Simard, Yann A. LeCun, John S. Denker, and Bernard Victorri, 1998. *Transformation Invariance in Pattern Recognition — Tangent Distance and Tangent Propagation*, pages 239–274. Springer Berlin Heidelberg, Berlin, Heidelberg.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. 2019. Unsupervised data augmentation for consistency training.

Bianca Zadrozny and Charles Elkan. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, page 694–699, New York, NY, USA. Association for Computing Machinery.