Leveraging BERT for Multi-Dimensional Sentiment Analysis of Employee Reviews

Fiodar Ryzhykau
ryzhikovfy@gmail.com

Abstract

There is still a relatively small amount of NLU research in the area of Human Social Science in the work environment. One of the main reasons is a high sensitivity of the personal information, and as a result - lack of publicly available data for analysis and experimentation.
The general definition of the "9-box" performance and potential analysis task, which is explored in this paper, maps to the text classification field, and can be characterized as a multi-dimensional sentiment analysis of employee reviews.
Conducted experiments validate the benefits of transfer-learning on a new domain with multi-class sentiment space. The main challenge in this task is a very little distinction between the close classes and sensitivity to the actual meaning of the words and phrases in the reviews. Instead of 2 (Positive/Negative) or 3 (Positive/Neutral/Negative) more commonly tested classes, this task is being focused on 9.
Results confirmed that sentiment analysis based on BERT (Devlin et al., 2018) significantly outperforms the well-known classification methods.
The code and results are available at

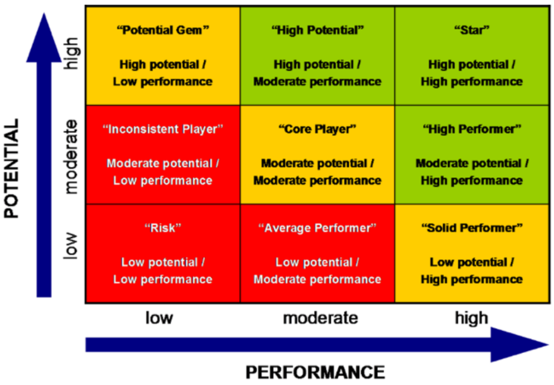
https://github.com/fryzhykau/BERT-employee-reviews-analysis

1 Introduction

For the recent past transfer-learning classification methods with state-of-the-art models like BERT (and the ones based on the similar design) or GPT2 have taken a leading place in NLP and started to overtake more classic approaches like Naive Bayes, Logistic Regression, SVM, Random Forest, etc.

The central hypothesis of this paper is that transfer-learning approach can provide significant benefits in the sentiment analysis on a new, unseen domain, with relatively-low amount of training data (around 100 examples per class). The key differentiator of this task is that sentiment should be characterized in 2 dimensions at once (making 9 classes instead of classic 2 or 3), which makes the task a bit more complex.

The general task is related to the employee assessment in modern companies, that is usually conducted on a quarterly or annual basis. "9-box model" (also know as "9 grid model" or "Performance and Potential Model") has been developed by McKinsey (2008) and helps to grade employee Performance and Potential in order to objectify the value of that employee to the company. This approach is used in many companies as a method to identify, support and promote the talent.
These "9-box grid" look as follows [1] :



	low	moderate	high
high	"Potential Gem" High potential / Low performance	"High Potential" High potential / Moderate performance	"Star" High potential / High performance
moderate	"Inconsistent Player" Moderate potential / Low performance	"Core Player" Moderate potential / Moderate performance	"High Performer" Moderate potential / High performance
low	"Risk" Low potential / Low performance	"Average Performer" Low potential / Moderate performance	"Solid Performer" Low potential / High performance

Given the review that corresponds to one of the categories, model should recognize the text sentiment and properly map the person to that category, considering 3 grades (Low, Medium, High) for both Performance and Potential dimensions.
BERT has been selected as a target deep-learning model to validate this hypothesis.

Considering the existing "knowledge" within the transfer-learning models and ability to quickly "learn" the meanings in the new field, based on the

[1] Image taken from article (Barnhill, 2017)

free-formed text, this experiment confirms the potential of provided approach in a broader variety of similar use cases, e.g. multi-class request classification for emails or understanding complex intents in the human descriptions.

Additional insight from result analysis shows that models are more precise to define extreme classes, which is probably a result of more vivid emotions and expressions provided by humans in those cases. However, comparing to the classic models, BERT did much better job on classifying the non-extremes proving the value of added knowledge and deep-learning approach.

One more relatively obvious, but worth mentioning hypothesis has been validated and confirmed, which assumes that Large BERT should outperform its Base configuration on the given task, due to the broader set of features and greater number of layers.

2 Related work

There is a great variety of papers available in the field of NLU sentiment analysis. The invention of the transfer-learning approach and pretrained deep-learning models like BERT has started to quickly drive the state-of-the-art performance in this field. However most of the reviewed works are focused on a relatively simple sentiment class configurations.

Focus of the papers "Sentiment Analysis of Twitter Data" (Agarwal et al., 2011) and "UDLAP: Sentiment Analysis Using a Graph Based Representation" (Castillo et al., 2015) was set on finding better word/text representations with a high degree of data pre-processing. However in most of the cases the performance boost in the new configurations of those classic models was marginal, considering the state-of-the-art baselines of that time. This led to a conclusion that classic methods were reaching their limits and NLP area required a significant breakthrough in order to get to the next level.

Reviewed papers dated 2019 and onwards (e.g. papers "Aspect-Based Sentiment Analysis Using BERT" (Hoang et al., 2019) and "FinBERT: Financial Sentiment Analysis with Pre-trained Language Models" (Araci, 2019)) have shifted focus on the construction of encoders/decoders as well as the selection of the proper set of layers and their configurations for the pretrained deep-learning models. Such context-based models have also shown a good transfer-learning capability for the cross-domain data. These methods have completely changed the field of research and allowed to even beat the human accuracy in many areas.

Another important conclusion from paper "FinBERT: Financial Sentiment Analysis with Pretrained Language Models" (Araci, 2019) states that deep-learning techniques for NLP, that previously have been known to be "data-hungry" should apparently no longer be considered as such. It has also shown very good results of BERT in the new domain.

Paper "Text Summarization with Pretrained Encoders" (Liu and Lapata, 2019) provided a good overview of the text summarization approaches, leveraging pretrained BERT encoders. The aim of the summarization is close in some aspects to the sentiment analysis and is set to condense a document into a shorter version while preserving most of its meaning. One important finding from this paper have shown that the most important content from the News dataset was located within approximately 5 first sentences (according Human-validated criteria of Informativeness, Fluency, and Succinctness). This finding helped to define the recommended size of the employee reviews for the data collection as well as with the further selection of the training set.

3 Data

3.1 Data Collection

Most of the publicly available datasets for the Social Science-related tasks contain mainly statistical data (numeric values or Boolean-type gradation for a particular characteristic).
Formal employee reviews usually contain sensitive data that can only be shared with either direct manager or senior members of the company. Such data cannot leave the premises of the company, and thus had to be generated or collected explicitly.

A popular crowd-sourcing platform - Amazon Mechanical Turk (MTurk) was used to collect reviews for this experiment. A custom task was created in order to make sure that there is a good level of variability and quality of data.
The task instructions looked as follows:

> *In this task you're asked to generate a free-form review for your imaginary colleague. The review should assess employee's performance for the last quarter by one of "9-box" categories below.*

> *Note: Review should be in English and not shorter than 4 sentences. Avoid using word combinations from the given Category as is (i.e. phrases containing words "performance" and "potential").*

Given "9-box" categories:

- Category 1: "Risk" (Low performance, Low potential)
- Category 2: "Average performer" (Moderate performance, Low potential)
- Category 3: "Solid Performer" (High performance, Low potential)
- Category 4: "Inconsistent Player" (Low performance, Moderate potential)
- Category 5: "Core Player" (Moderate performance, Moderate potential)
- Category 6: "High Performer" (High performance, Moderate potential)
- Category 7: "Potential Gem" (Low performance, High potential)
- Category 8: "High Potential" (Moderate performance, High potential)
- Category 9: "Star" (High performance, High potential)

Employee names were randomly generated from the vocabulary for a better variability. And only English-speaking Workers were qualified to perform these MTurk tasks.

As a result, the dataset of about 1000 reviews has been collected with initial validation for the main guidelines.

3.2 Data Analysis

Further data review and analysis of the collected data provided the following insights:

- There is a wide range of the review descriptions.
- Reviews are provided in multiple domains (e.g. music, sport, IT, etc.), which is beneficial for variability as well.
- People still tend to use word combinations from a given Category (i.e. low performance, high potential, "Core Player", etc.), which can make some reviews biased to the description.
- People are not always syntactically and grammatically correct (e.g. periods with wrong indents, sentences start with lower-case letters, etc.).
- People do not always make very precise match of the review to the given category (needs additional judgment).
- Sometimes by not stating/expressing something people may implicitly mean something as well (e.g. by not saying how great the performance is, they might mean that it is not "High", but rather "Medium").
- For extreme values (i.e. "High" and "Low" grade for Performance and Potential at the same time) MTurk workers tend to express their sentiment more vividly. In general - expecting similar situation for manager's reviews in the real world.

3.3 Data Cleanup

The following cleansing guidelines were used:

- Description should match the category. If the review is reasonable and generally of a good quality, the category could be manually updated to the more precise one in order to keep a good example in the dataset.
- Keep only the reviews that are clear to describe a given Category with specific "Performance" and "Potential" degrees. Remove inconsistent or unclear descriptions, like:
  - *"Riles is a great person to have at the office. They put out average work and show average performance. However, this person has great potential and is starting to reach it. I can not wait until Riley achieves her full goals, she will be great."*
  - *"Andrea has always shown that she is such a hard worker and will push to accomplish. with her her hard work she does show some potential, but not the best."*
- Remove the examples where the word "Category" is explicitly mentioned to avoid making model biased to a certain index associated with this keyword.

- Remove contradictory examples. For example: - *"While Matthew Reid does not have the potential to learn extra skills due to his learning disability, he is still able to perform exceptionally well with the skills he does have. I have gained much confidence in Reid's ability to outwork and outlast his coworkers who have a lot more potential. I have even given him projects that high potential workers would receive, simply because I know he can do the work faster and more accurate. I can always count on Reid to get a major job done successfully."*

- Remove very short examples (e.g. 1 sentence), but keep the ones with a decently long description even if they consist of 2-3 sentences.

- Remove gibberish characters. For example: - *"In regards to EloiseÃâás quarterly work conduct, she has shown a great sign of her job fulfillment."*

- Bad punctuation or insignificant grammar mistakes were not considered as an issue.

Several examples of good reviews:

- *"Amoy has not improved in her personality to handle the customer service aspect of her job. She does not over-achieve on anything, as a matter of fact I believe that she is only here to get a paycheck and this is not good for the company, especially in the position of a front-desk receptionist. I also feel that she is a high performer of watching the clock. It would benefit the company to write her up to see if any improvement."*

- *"Shannon is dependable, clearing her expected work every day. She seems happy in her role and unlikely to want to move up the ladder. Shannon is a great addition to the team and you can ask her for help where needed. Recommended for projects."*

3.4 Final Dataset

In regards to the dataset breakdown for the model training, validation and testing, the following proportions were selected: Train/Dev/Test – 65%/15%/20%

Further experiments showed that the best performance is achieved for the Training examples of size between 150 and 600 characters, so such filter was applied to exclude very short and relatively long reviews respectively.

All the cleansing and filtration reduced the size of the original dataset by approximately 10%. This brought a certain disbalance in the final Training Set, and resulted in the breakdown illustrated in the Table 1.

Clategory	**N/V**	**V**	**Total**
Category 1: 'Risk' (Low performance, Low potential)	46	51	97
Category 2: 'Average performer' (Moderate performance, Low potential)	58	14	72
Category 3: 'Solid Performer' (High performance, Low potential)	68	2	70
Category 4: 'Inconsistent Player' (Low performance, Moderate potential)	57	23	80
Category 5: 'Core Player' (Moderate performance, Moderate potential)	56	32	88
Category 6: 'High Performer' (High performance, Moderate potential)	45	21	66
Category 7: 'Potential Gem' (Low performance, High potential)	23	8	31
Category 8: 'High Potential' (Moderate performance, High potential)	48	17	65
Category 9: 'Star' (High performance, High potential)	65	22	87
Grand Total	**466**	**190**	**656**

Table 1: Test Set (V - manually validated, N/V - not validated).

Dev Set contained 116 examples, out of which 35% were manually reviewed.

Final Test Set was collected from 25 separate examples for each class (225 total), and all those examples were manually reviewed and validated.

4 Models

The following models were selected for experiment validation:

4.1 Deep-learning/Transfer-learning models:

- [Huggingface] BERT base uncased
- [Huggingface] BERT large uncased
- [Huggingface] BERT base cased
- [Huggingface] BERT large cased

The size of BERT large model is 1.34GB, size of BERT base model is 0.44GB.

4.2 Classic models (for comparison):

- [Scikit-learn] SVM (SVC with 2 different kernels: 'linear' and default - 'rbf')
- [Scikit-learn] Multinomial Naïve Bayes
- [Scikit-learn] Random Forrest Classifier
- [Scikit-learn] Logistic Regression

4.3 Additional Configuration Specifics and Parameters:

- BERT: BertTokenizer with max sequence length = 512; Optimizer – AdamW.

- Classic models: TfidfVectorizer, CountVectorizer

5 Experiments

5.1 Environment Configuration

Experiments were executed in Google Colab environment with 12GB NVIDIA Tesla K80 GPU and High RAM (25.5GB).
Code is written in Python 3 and leverages Pytorch platform for the parallel processing on GPU.

5.2 Model Training Time

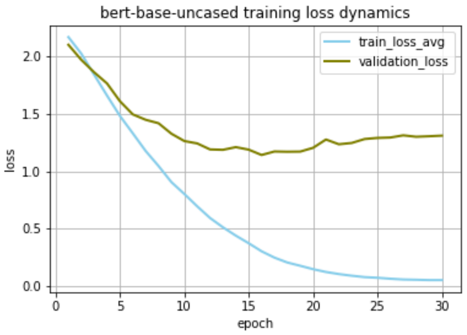
The following training time was required for each experiment:

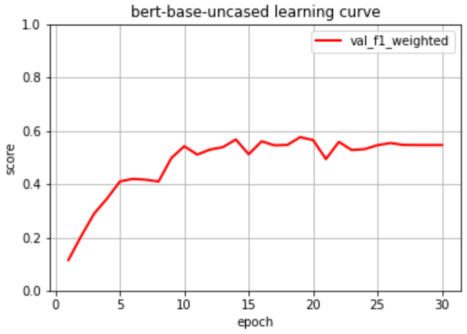
- Large BERT: up to 15 epochs (3min each. Total avg training time: 45min)
- Base BERT: up to 30 epochs (25sec each. Total avg training time: 15min)
- Classic Models: up to 1 sec

5.3 BERT Experiments

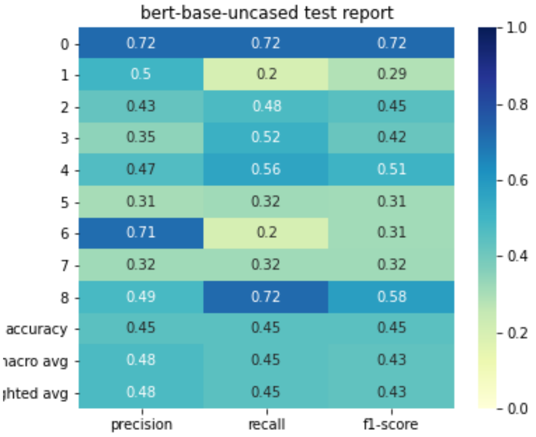
On the charts below ”9-box” Categories 1-9 correspond to the labels 0-8 respectively. Weighted AVG F1 metric was used for the model comparison.

5.3.1 BERT Base Uncased



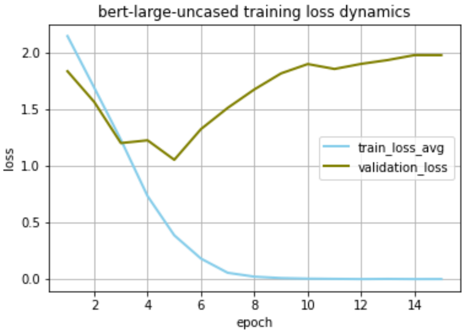


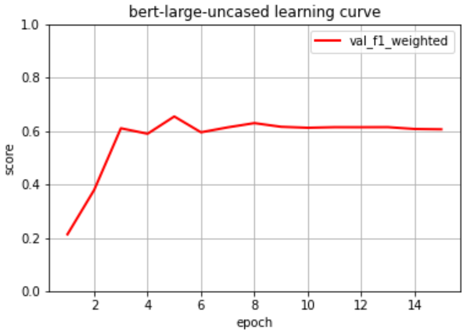
Best test results from the model based on 30-epoch training:


bert-base-uncased test report
0.0
0.2
0.4
0.6
0.8
1.0

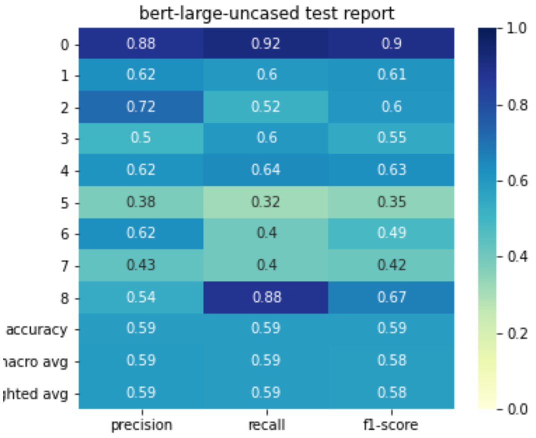
	precision	recall	f1-score
0	0.72	0.72	0.72
1	0.5	0.2	0.29
2	0.43	0.48	0.45
3	0.35	0.52	0.42
4	0.47	0.56	0.51
5	0.31	0.32	0.31
6	0.71	0.2	0.31
7	0.32	0.32	0.32
8	0.49	0.72	0.58
accuracy	0.45	0.45	0.45
nacro avg	0.48	0.45	0.43
ghted avg	0.48	0.45	0.43

5.3.2 BERT Large Uncased



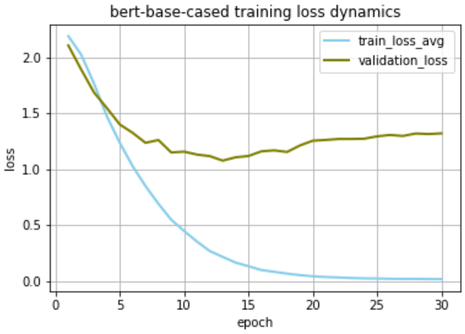


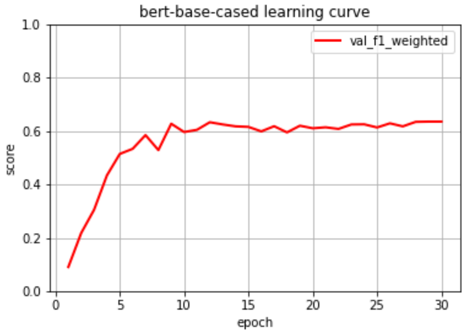
Best test results from the model based on 5-epoch training:


bert-large-uncased test report
0.0
0.2
0.4
0.6
0.8
1.0

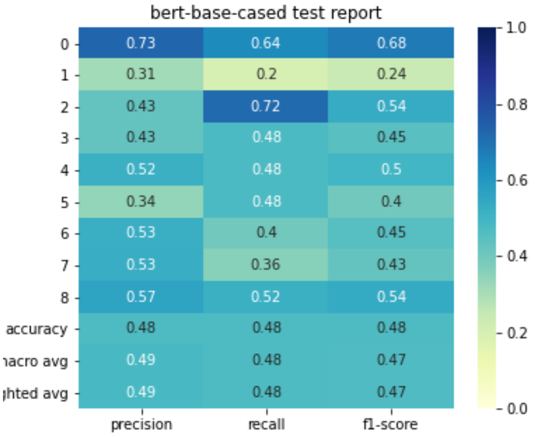
	precision	recall	f1-score
0	0.88	0.92	0.9
1	0.62	0.6	0.61
2	0.72	0.52	0.6
3	0.5	0.6	0.55
4	0.62	0.64	0.63
5	0.38	0.32	0.35
6	0.62	0.4	0.49
7	0.43	0.4	0.42
8	0.54	0.88	0.67
accuracy	0.59	0.59	0.59
nacro avg	0.59	0.59	0.58
ghted avg	0.59	0.59	0.58

5.3.3 BERT Base Cased





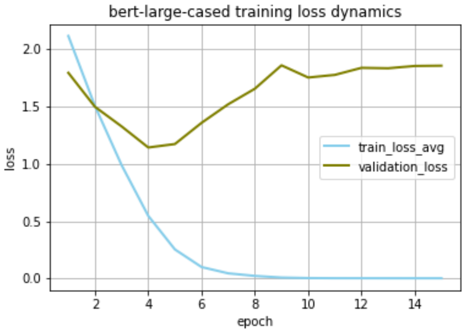
Best test results from the model based on 30-epoch training:

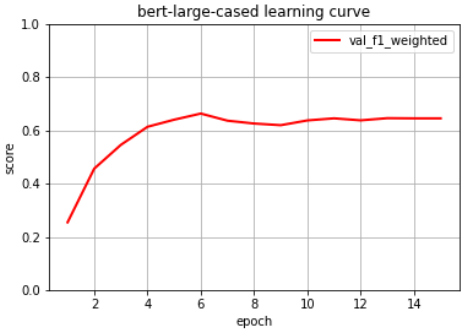

bert-base-cased test report

	precision	recall	f1-score
0	0.73	0.64	0.68
1	0.31	0.2	0.24
2	0.43	0.72	0.54
3	0.43	0.48	0.45
4	0.52	0.48	0.5
5	0.34	0.48	0.4
6	0.53	0.4	0.45
7	0.53	0.36	0.43
8	0.57	0.52	0.54
accuracy	0.48	0.48	0.48
ıacro avg	0.49	0.48	0.47
ȷhted avg	0.49	0.48	0.47

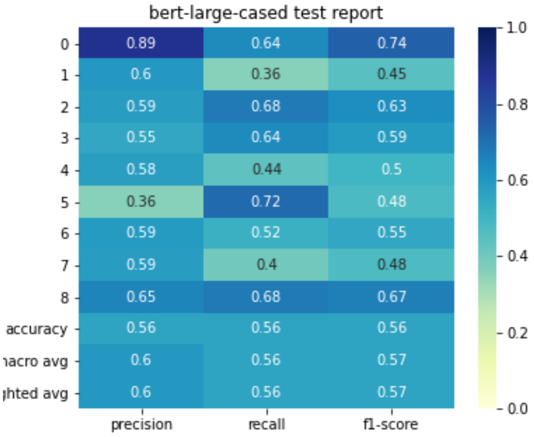
0.0
0.2
0.4
0.6
0.8
1.0

5.3.4 BERT Large Cased





Best test results from the model based on 15-epoch training:


bert-large-cased test report

	precision	recall	f1-score
0	0.89	0.64	0.74
1	0.6	0.36	0.45
2	0.59	0.68	0.63
3	0.55	0.64	0.59
4	0.58	0.44	0.5
5	0.36	0.72	0.48
6	0.59	0.52	0.55
7	0.59	0.4	0.48
8	0.65	0.68	0.67
accuracy	0.56	0.56	0.56
ıacro avg	0.6	0.56	0.57
ȷhted avg	0.6	0.56	0.57

0.0
0.2
0.4
0.6
0.8
1.0

For both Cased and Uncased versions of Large BERT, starting from epoch 6 model validation curve shows significant growth in overfitting, with no additional benefit to the validation performance.

The Table below combines the results for all tested BERT versions:

BERT Version	**F1 w.avg %**
BERT Base Uncased	43.15
BERT Base Cased	47.32
BERT Large Uncased	**58.34**
BERT Large Cased	56.58

Table 2: Comparison of BERT Final Test results.

5.4 Classic Models

3 additional but commonly-used data preprocessing methods were tested for classic models in order to boost their performance: 1) stop words removal; 2) stemming; 3) lemming. All of them were based on English corpus of nltk library.

For the given dataset, only lemming positively impacted results across different models and vectorizers. However the benefit from those modifications was not-significant (up to 1.5% of F1 w.avg, comparing to the original baseline), and in some cases models showed even lower results.[2]

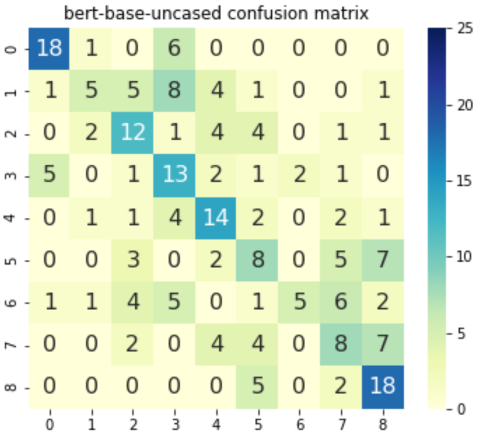
#	Model Configuration	F1 w.avg %
1	multinomialMB_alpha_.2_tfidf	24.38
2	multinomialMB_alpha_.2_tfidf_clean	25.59
3	multinomialMB_alpha_.2_count	29.67
4	multinomialMB_alpha_.2_count_clean	30.66
5	SVC_linear_C1.1_tfidf	27.72
6	SVC_linear_C1.1_tfidf_clean	26.74
7	SVC_linear_C1.1_count	27.63
8	SVC_linear_C1.1_count_clean	26.54
9	SVC_default_rbf_C2.5_tfidf	24.14
10	SVC_default_rbf_C2.5_tfidf_clean	25.71
11	SVC_default_rbf_C2.5_count	25.14
12	SVC_default_rbf_C2.5_count_clean	25.39
13	RandomForest_tfidf	20.49
14	RandomForest_tfidf_clean	19.65
15	RandomForest_count	22.91
16	RandomForest_count_clean	21.08
17	LogisticRegression_liblinear_tfidf	22.36
18	LogisticRegression_liblinear_tfidf_clean	22.68
19	LogisticRegression_liblinear_count	30.44
20	**LogisticRegression_liblinear_count_clean**	**31.69**
21	LogisticRegression_lbfgs_tfidf	23.62
22	LogisticRegression_lbfgs_tfidf_clean	23.33
23	LogisticRegression_lbfgs_count	28.39
24	LogisticRegression_lbfgs_count_clean	27.65

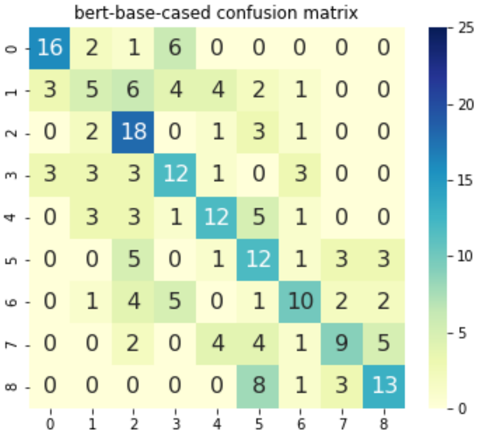
Table 3: Final Test results for Classic model configurations.

6 Result Analysis

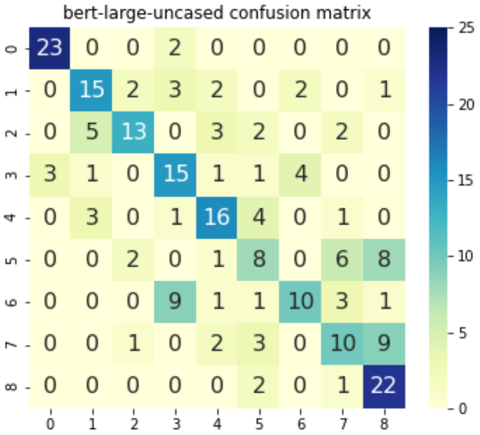
6.1 BERT Model Versions

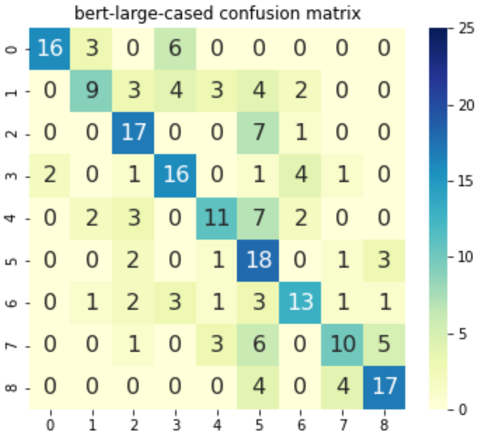
The following confusion matrices were generated in order to better showcase the model performance for all classes altogether:





Considering the main diagonal view of 2 matrices above, Base Cased version of BERT did a better prediction for non-extreme classes.





Similarly to the Base version, BERT Large made a better prediction for the extremes in the Uncased version and showed the highest overall average score, however the Cased version did better for the non-extreme classes.

In contrast with the Base version of the model,

[2] Suffix ”clean” in the model configuration name refers to the experiments where additional data pre-processing was applied.

Large Cased BERT got lower average scores than Uncased. This can lead to the hypothesis that with more ”embedded” knowledge available for the pre-trained model case-sensitivity becomes less important for this task.

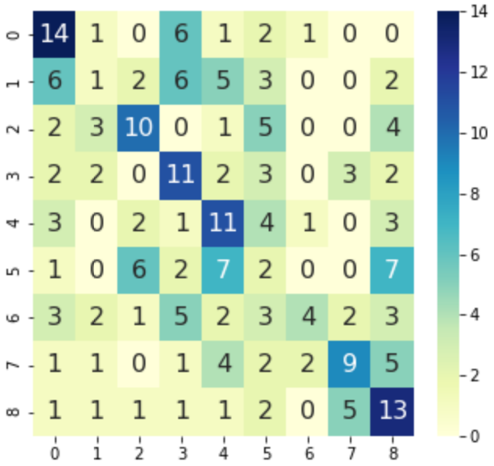
One of the other potential reasons of misclassification for the non-extreme classes is a degree of the non-validated data and not even the distribution in the training set.

6.2 Classic Model Configurations

The best results for the Classic models were achieved on the configuration with the LogisticRegression classifier, liblinear kernel and CountVectorizer applied to the pre-processed text.

However even in that case there is a broad and inconsistent distribution of the miss-predictions around the class space. See the confusion matrix below:

LogisticRegression_classifier_liblinear_clean



	0	1	2	3	4	5	6	7	8
0	14	1	0	6	1	2	1	0	0
1	6	1	2	6	5	3	0	0	2
2	2	3	10	0	1	5	0	0	4
3	2	2	0	11	2	3	0	3	2
4	3	0	2	1	11	4	1	0	3
5	1	0	6	2	7	2	0	0	7
6	3	2	1	5	2	3	4	2	3
7	1	1	0	1	4	2	2	9	5
8	1	1	1	1	1	2	0	5	13

6.3 Result Comparison

Comparing the set of the confusion matrices for Classic models and BERT, it becomes clear that the biggest chunk of the missed predictions is located much closer to the main diagonal in the case of BERT. That means that there is a certain, but relatively small contextual difference between the classes, which shows potential in even more precise classification with additional training examples.

7 Conclusion

The set of conducted experiments showed that BERT indeed outperforms the well-known Classic models, with just a set of basic configuration tweaks for optimal training speed. That also proves the benefits of applying transfer-learning methods on a new domain without significant research on the dataset, features and model tuning.

In the real world the datasets are not always clean, and can be sensitive to the certain human biases. Review of the BERT classification results, even considering the current state of collected dataset, has shown that model can be more precise comparing to the initially described Category (based on the judgment of multiple people), which is very promising and shows a great degree of generalization. It has also proved that such approach does not require a large training dataset in order to get to the very decent results.

However one should keep in mind the size of such Transfer-learning models, required hardware and the time it takes to train them. Classic models are still a bit more simple to build and use, and may require less effort to start and set the initial baseline.

Additional data validation and cleansing seem to be the best way to further improve the results. Newly collected examples should be properly reviewed and analyzed in order to sustain the Category definition consistency. More examples and their equal distribution within the training set should help as well.

Acknowledgments

This work has been greatly supported by Adam Klein who provided continuous mentorship and direction. And also by my wife - Hanna Ryzhykava, who assisted with the data review and provided additional point of view during data analysis.

I want to thank Professor Chris Potts for encouraging ideas as well as providing a structured approach to perform the experiment and create this paper. Additional appreciation to Steve Haraguchi and Stanford N224U course team for giving the prompt guidance during the course.

I also want to thank the Hugging Face organization for making the pre-trained BERT models available to the World and specifically for experimentations leveraged for this work. And lastly, to Ari Anastassiou who made a very clear and helpful course on ”Sentiment Analysis with Deep Learning using BERT” (Coursera, 2020) that became a basis for this experimentation.

Authorship

Fiodar Ryzhykau - general task idea, model coding, data analysis, experimentation, task generation for MTurk, content and compilation of the final paper. Hanna Ryzhykava - MTurk data review and consistency validation.

References

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2011. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Language in Social Media (LSM 2011)*, pages 30–38, Portland, Oregon. Association for Computational Linguistics.

Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models.

Amy Barnhill. 2017. Using the performance values matrix alongside a 9-box grid.

Esteban Castillo, Ofelia Cervantes, Darnes Vilariño, David Báez, and Alfredo Sánchez. 2015. UDLAP: Sentiment analysis using a graph-based representation. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 556–560, Denver, Colorado. Association for Computational Linguistics.

Coursera. 2020. Sentiment analysis with deep learning using bert.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Mickel Hoang, Oskar Alija Bihorac, and Jacobo Rouces. 2019. Aspect-based sentiment analysis using BERT. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 187–196, Turku, Finland. Linköping University Electronic Press.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. 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)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

McKinsey. 2008. Enduring ideas: The ge–mckinsey nine-box matrix.