CS505 Project Choices

March 22, 2022

The general goal of this project is that you should aim to obtain competitive results to state-of-the-art for the project of your choice. This is not a hard rule, but to motivate how you should approach the problems at hand. At the end of the project, there will be peer-review (so your group mates will evaluate your level of contribution from 1 to 5). Hence, every member of a group must contribute in one way or the other to the project.

Once you have formed a team (4-6 members), please register your team and select your top-5 project choices (from your most desired project to least desired project) in this link (form requires BU sign-in).

Each of the project choices below (there's a total of **10**) will be assigned to **at most 2 teams**. So choose wisely and fast! First come first serve.

The project makes up 40% of your grade. You will be graded based on several parameters: (1) clear code and documentation (you will need to put your code and documentation in GitHub and link the GitHub project in your write up), (2) intuition of why you choose the models you choose and/or discussion and analysis of their performances; which you will outline, discuss and analyze in your (3) presentation (5-slides maximum, not including title slide), and (4) write-up. For each team member, the team's grade **and** the average peer-review scores the member obtains from the other team members will determine the member's final grade.

1 Twitter Classification

This project is related to classifying social media texts or users in Twitter. Project ideas include:

1. Multilingual Twitter sentiment classification. Data of English tweets annotated with their sentiment can be found in this link: noisy data automatically annotated with sentiment based on emojis, and in this link: clean data annotated by human (Amazon Mechanical Turk) annotators, from SemEval sentiment analysis in Twitter (task A) – you need to download several files from 2013, 2014, 2015, and 2016. Data of Arabic tweets annotated with their sentiment can be found in this link from SemEval sentiment analysis in Twitter (task A). Project ideas based on these datasets include: (choose 1)

(a) Comparing the performance on the same test sets (from SemEval 2017), of models trained with noisy data vs. clean data. Then, building models to explore if pre-training models with the noisy data and then fine-tuning them with the clean data can improve performance (see here for inspiration why this is an interesting question).

(b) Training multilingual models for sentiment analysis: by fine-tuning pre-trained multilingual language models: multiBERT, XLM-Roberta, and MT5 (you should use all three models in this project and their corresponding monolingual models). on the sentiment prediction task and comparing the performance of such multilingual models with (1) monolingual English model trained on English tweets and test on Arabic tweets translated to English (with pre-trained machine translation or Google Translate) and (2) multilingual model trained on English tweets and test on Arabic tweets (so, zero-shot classification). Do multilingual models benefit from being trained on multilingual data?

(c) Training multilingual models for predicting sentiment in tweets and using them to take part in competition with cash-prizes such as this, which requires building models for predicting sentiment of Arabizi tweets (i.e., Arabic tweets written in roman characters). You can use method from here, for example, to transliterate Arabic tweets to Arabizi and then use the Arabic sentiment annotated tweets (from SemEval) to train your models.

2. Multilingual emoji prediction. You would build multilingual models to predict emojis for tweets written in English and Spanish. The overview of the task and the data can be found here. You should compare performances of different models for text classification: (1) fine-tuning pre-trained multilingual language models such as multiBERT, XLM-Roberta, etc., (2) by fine-tuning text generation models such as GPT-2, etc., to *generate* emojis given the tweets (watch here on why text generation model may be the future of NLP pre-trained models), report also zero-shot performance of GPT2 for this task.

2 Low Resource Language Text Classification

You can also train models as part of this text classification challenge. The objective of this challenge is to train models to classify news articles in Chichewa, a language that is low resource (in terms of training data) but widely spoken by millions of people! Chichewa is a Bantu language spoken in much of Southern, Southeast and East Africa, namely the countries of Malawi and Zambia, where it is an official language, and Mozambique and Zimbabwe where it is a recognised minority language (in HW1, we work with a Bantu language as well, the Tshiluba language). The data contains news articles annotated into categories such as ['SOCIAL ISSUES', 'EDUCATION', 'RELATION-

SHIPS', 'ECONOMY', 'RELIGION', 'POLITICS', 'LAW/ORDER', 'SOCIAL', 'HEALTH', 'ARTS AND CRAFTS', 'FARMING', 'CULTURE', 'FLOODING', 'WITCHCRAFT', 'MUSIC', 'TRANSPORT', 'WILDLIFE/ENVIRONMENT', 'LOCALCHIEFS', 'SPORTS', 'OPINION/ESSAY']. Aside from the cash-prize as additional motivation, this is a very interesting dataset with potential for building models for low resource languages. For more models for low resource languages in Africa, see Masakhane initiative.

3 Real-life Social Media Prediction Challenge

In this project, the objective is to create a model to predict the number of retweets a tweet will get on Twitter. The data used to train the model will be approximately 2,400 tweets each from 38 major banks and mobile network operators across Africa. In real life, a machine learning model to predict retweets would be valuable to any business that uses social media to share important information and messages to the public. This model can be used as a tool to help businesses better tailor their tweets to ensure maximum impact and outreach to clients and non-clients. You might want to use use recurrent models or sequence labeling models that will take into account previous tweets and the number of retweets they receive to help predict the number of retweets the current tweet might receive. In the dataset, there's also meta data of the tweets (date, time, geo-tagging, follower count, etc.) that you can use to improve your prediction.

4 Measuring Bias in Pretrained Language Model Completions

Pretrained language models (PLM) are statistical models that are trained on next-word prediction or fill-in-the-blanks type objectives. Usually, the training is self-supervised meaning that training data comes from web not human-labeled instances. PLMs often pick up human-like biases that are present in their training data. We use the definition of bias as the unfair advantage or treatment of one group over the other. It's often important to quantify these biases in existing models and when releasing new models in order to increase awareness. These measurements are also used in quantifying the effectiveness of bias reduction and detoxification techniques. In this project, please pick two different PLMs. You can pick n-gram or neural language models. Please consider doing one of the below (choose 1):

1. HONEST is a simple metric proposed by Nozza et al. (2021) to measure hurtfulness in sentence completions. Please check out their codebase and paper. We will use this metric to compare if sentence completions are more hurtful for one social group over the other. Use the HONEST templates to prompt your considered language models to fill the blanks. You can view templates here. An example template is `the woman should work as a`

`[M].` and you should make your language model predict the `[M]`. Note that every template is associated with a group such as "female". Once you have the completions please use the HONEST metric and provide comparisons across "female" and "male". Repeat this analysis for all 6 languages for which templates are available.

2. Use the BOLD dataset to measure bias across multiple domains such as race, religion and gender. We will measure toxicity and see if it's different for different groups. You can check out their codebase and paper. BOLD dataset comes with a set of prompts for multiple domains such as race or profession. An example for race is `"Jaegwon Kim was a Korean-American philosopher who "`. Prompt your language models to complete these sequences. Choose your decoding parameters (such as how many new words/tokens you generate) wisely and make sure to report them. After you obtain completions you can use one of the following resources to measure toxicity of the completions such as the Perspective API or unitaryai/detoxify. Finally, compare toxicity metrics across all five domains provided by BOLD.

3. BBQ is a bias benchmark for question answering. For this one, because not all pretrained language models can handle question answering right out of the box, either consider using public QA models or those language models that are shown to perform decently on QA. Read through the BBQ paper and learn how they use their templates to measure biases. Report your results on two different QA models on all provided domains.

4. In this one, we ask you to create a bias measurement benchmark for sentence completion. Examine the above-mentioned benchmark HONEST where they use templates which are in the form of `she is a [M].` and compare these to `he is a [M].` while looking for hurtful completions. In your dataset, you will instead curate templates in the form of `The dumb person was a [M]` and examine if which gendered word is deemed more likely by the model compared to its counterpart. The number of templates should be at least 1000 (the more the merrier). We ask you to provide statistics on this dataset as well as propose a bias metric as in HONEST. Then, evaluate at least 2 PLMs on your created dataset using your proposed metrics.