Causal Effect of Linguistic Register Level on Sentiment Classification Models

Liam Hazan and Eyal Bar-Natan

March 2021

Abstract

With Black box models like DNNs on the rise, it is essential to find a way to understand what is affecting the model to make its decisions. Language models like all data-based models may contain unwanted biases from the data. Our goal is to estimate the causal effect of the register level concept on language models such as BERT. By replacing adjectives with their synonyms which are characterized with higher/lower register, we can simulate the do-operator to measure the causal concept effect of register level on language models in tasks like sentiment analysis. Our idea is based on the paper CausaLM by Feder, Oved, Shalit and Reichart[1].

1 Introduction and Causal Inference

In the usual causal inference setup we would have:

- Unit: can be a person, company,family, and in our case a sentence.
- Treatment T, usually a binary variable, 0 or 1.
- Outcome Y, when $Y = TY_1 + (1 - T)Y_0$
- Confounder X , a variable influences both T and Y and thus may lead us to incorrect causal connection between T and Y.
- Potential outcomes - Each unit i has two potential outcomes:
  - Y_0 is the potential outcome had the unit received treatment 0
  - Y_1 is the potential outcome had the unit received treatment 1

We would like to measure the effect of the treatment T on the outcome Y while avoiding the effect of the confounder X, in other words we would like to

measure the ATE:

$$ATE = \mathbb{E}[Y_1 - Y_0] \approx \mathbb{E}[Y|do(T = 1)] - \mathbb{E}[Y|do(T = 0)]$$

If we can intervene on the value of T it cancels the effect of X on T and we can measure the ATE without the confounding problem.

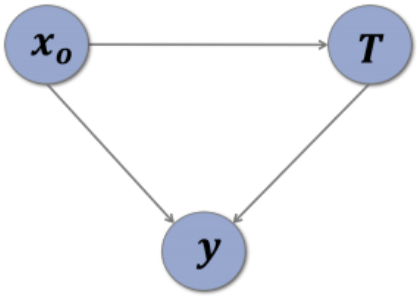

Figure 1: basic causal graph

1.1 Identification

Sufficient conditions for causal inference to be possible:

1. Stable Unit Treatment Value Assumption
   - The potential outcomes for any unit do not vary with the treatments assigned to other units
   - For each unit, there are no different forms or versions of each treatment level, which lead to different potential outcomes
2. Consistency - For a unit that receives treatment t, we observe the corresponding potential outcome Y_t.
3. Ignorability / No unmeasured confounders - The potential outcomes are independent of treatment assignment, conditioned on observed covariates x
4. Common support - $p(T = t|X = x) > 0 \quad \forall t, x$

1.2 Our Case

Usually, causal models are built for understanding real-world outcomes, while we would want causal model for ML model interpretability. In our case the unit is a sentence affected by some concepts like adjectives, topic, and register, the

classification decision (positive/negative) is the outcome, and the intervention is on the model's input, i.e. the sentence.
While abstract linguistic concepts express meaningful information, they are not explicitly encoded in the text. Such concepts might push the model towards making specific predictions, without being directly modeled and therefore interpreted. Training a generative model to condition on a concept, such as register level, and produce counterfactual examples that only differ by this concept is still intractable in most cases involving natural language. That is why we plan to use another method to generate counterfactual examples using the frequency of words and labeled data-set of the "level" of words (explained in detail in the do-operator section). While interfering on the text units by replacing some of the words within it, we actually notice both cases on the same text unit: the potential outcome while the unit belongs to $concept = 0$ (basic register level), as well as $concept = 1$ (change to high register level). That is to say, by such interfering we mentioned above, we are able to measure simultaneously the desired model outcome we described, with both variations of the concept of interest (two register levels).

As for the above assumptions, we can easily notice that each of them holds, in the same manner we introduced the potential outcomes we have in our causal research.

Yet, one can imagine a dependency between the potential outcomes, to the treatment assignment, conditioned on some covariates. Or in other words, a case where there are some unmeasured hidden confounders in our analysis. This scenario might occur if we think of some hidden concepts, besides register, that can be affected while transforming the text units by replacing words with their synonyms, as we will explain in detail here. The undesired dependency on the potential outcome, might occur in such cases, if those hidden affected concepts have an effect on the sentiment outcome we examine in our model.

We plan to avoid other concepts effect (potential confounders) on the register of the text by simulating the do-operator(Figure 3). By applying words replacements for each unit we force the register level to be high such that the replacements would suit the text grammatically and logically as much as possible.

2 Research Question

What is the causal effect of C_R on sentiment analysis models (which based on language models like Bert)?

We define the concept C_R - the register level of the text which is one of the concepts that generate the text.

3 Definitions

X - observed text
C - set of variables that indicate the existence of a predefined concept in the text: $C = \{C_j \in \{0,1\} | j \in \{0,1,...,k\}$
Φ - pre-trained language representation model (like Bert)
f - classifier
$\overrightarrow{z}$ - probability output for an example X - the vector of all probabilities

3.1 Causal Concept Effect (CaCE) (Goyal et al. 2019a)

The causal effect of a concept C_j on the class probability distribution $\overrightarrow{z}$ of the classifier f trained over the representation ϕ under the generative process g is:

$$CaCE_{C_j} = \langle \mathbb{E}_g[\overrightarrow{z}\big(f(\phi(X))\big) | do(C_j = 1)] - \mathbb{E}_g[\overrightarrow{z}\big(f(\phi(X))\big) | do(C_j = 0)] \rangle \quad (1)$$

3.2 The do-operator (Pearl's notation[2])

$P(A|do(B = b))$: the probability of A given an intervention that sets B to b.

4 Our do-operator

First, let us take a look at the dependencies graph (Figure 2) that describes the classifying task we mentioned above. Next, we will explain the process of applying the do-operator, that is changing this dependencies graph (Figure 3).

Let us begin with explaining the method we used to determine the text words registers. For this task we used the paper "Classification of word levels with usage frequency, expert opinions and machine learning" (Gihad N. Sohsah Muhammed Esad Ünal Onur Güzey)[3]. In their work they used raw survey results of words classifications as obtained from english teachers. They created a table consists of 7000 entries, each has an averaged evaluation of 3 different teachers for a single word with a specific part-of-speech. Then, they mapped the numeric results to corresponding popular CEFR standard levels (Common European Framework of Reference for Languages). The six reference English levels (A1=1 to C2=6) are widely accepted as the global standard for grading an individual's language proficiency. Afterwards, the frequencies of English 1-grams in Google books data were used to determine how frequent is each (Word,PoS) pair. This data then were used to train three classification models (a Random Forest classifier, a Neural Network classifier and a Support Vector Machine classifier). Finally, these trained models were used to classify all the words in google books dataset. Altogether it created a dataset of 51,353 different (Word,PoS) classified couples.

Using this whole classified dataset, for each word we wanted to replace or take as a replacement in a given paragraph, we could average the models scores in hand and normalize the score to a 0-1 scale by dividing it by 6.

For words that were not included in this dataset, we found their frequency in English, by the *wordfreq* Python library, which set those frequencies using data from Wikipedia, Subtitles, News, Books (from Google Books N-grams 2012), Web text, Twitter, Reddit, etc. They covered words that appear at least once per 100 million words.
We then normalized these numbers by a logarithm such that:
$level = \frac{-log(word\_frequency)}{-log(1/(10^8))}$.
Zero-frequent words were attached with $score = 1$ (the highest normalized level). This normalization creates a similar 0-1 scale as the previous. For example, the average scores of the adjectives 'magnificent' and 'prosperous' are 0.61 and 0.66 respectively, and both would have got a score of 0.55 if calculated by its frequency (assuming they did not appear in the classified dataset).

In order to address the need to find a synonym for each adjective in the text, we used the Offline Database of Synonyms by *thesaurus* website, which stores 169,009 entries of tuples consisted of word, key (some words have multiple meanings), PoS, synonyms array.
For example, using this database of synonyms, we are able to transform the sentence "Besides being boring, the scenes were oppressive and dark" to "Besides being mundane, the spectacles were oppressive and somber".

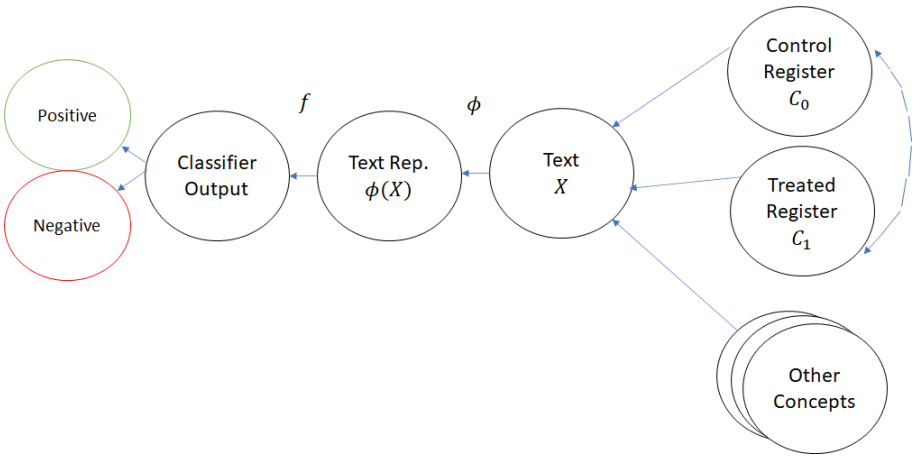

Figure 2: dependencies graph

4.1 Our Assumption

We aim to examine the different decisions that the sentiment classification model would take, under an intervention on the register level of the input text. That is to say, we assume that some bias might exist in a sentiment classification model that relies on a language representation model such as BERT. This might occur

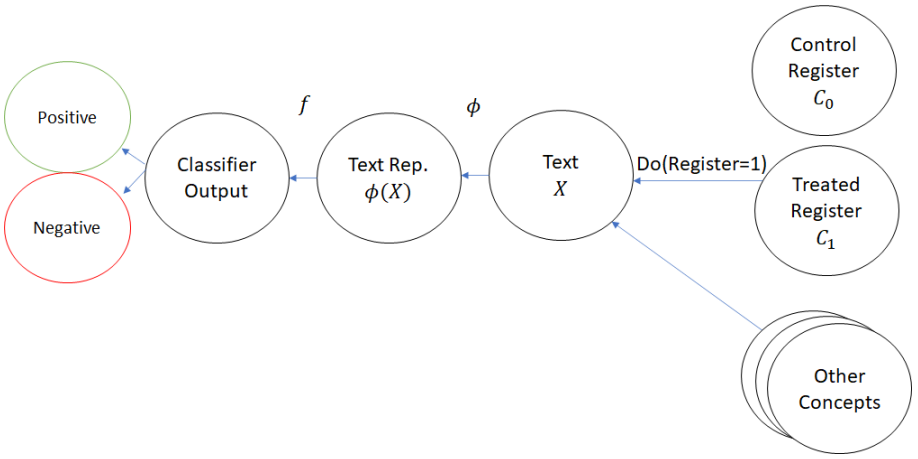

Figure 3: dependencies graph with do-operator

in cases where the different representations that the language model would supply, might differ too much, when applied on two variations of text: before and after an intervention on the register concept. Therefore, our approach to the texts transformations, is to affect, as much as possible, nothing but the register level of the chosen adjectives. That means that we intend to find substitute synonyms that would make no impact on a model without bias. Therefore, we assume that the best choice for replacing adjectives, would be the synonym option that is closest to the adjective in terms of register, frequency and sentiment (of the specific word itself) differences.

4.2 The Transformation

We will briefly explain the transformation method we applied on the input paragraphs. Let us review the algorithm steps:

1. Tokenize the paragraph into (word,PoS) tuples and take only the adjectives. For each adjective *adj* do the following steps:

2. Calculate the average register score of *adj* as described above. Use *adj* if its average score belongs to a pre-defined interval of register levels (according to choosing low or high level adjectives).

3. Find a proper adjective substitute (a synonym) according to the next steps.

4. Each *syn* among the substitute synonyms candidates should be of a higher register level than *adj*. According to our assumption, the best option for substitution should maintain the following similarity constraints: the deltas between the register, sentiment and frequency values of *syn* and *adj* should be no higher than respective hyper-parameters setting the desired

similarity thresholds. We use the *SentimentIntensityAnalyze* function from *nltk.sentiment* library in order to find the sentiment of the words.

5. After this filtering, in order to substitute *adj*, we choose the candidate *syn* which has the lowest delta between its *word_embedding* to *adj*'s embedding. We use the GloVe word embedding from *torchtext*.

Thus, we implement our do-operator, which can be visualized in Figure 3.

5 The Dataset

In our experiment we use the IMDB review Dataset containing 50000 movie reviews with rating of 1-5 scale. We consider rating of 4 and above as positive and the rest as negative.

The reviews can be longer than 1000 words which is a problem for the version of BERT we used so we used only the first 512 words.

The data wasn't completely clean so it we needed to remove some html tags, special characters and more.

In addition we also used tweets complain dataset containing 3200 tweets while half of them are complaints on air companies an the are not.

The tweets also needed to be cleaned from mentions (@..) and special characters.

6 Experiments

In our experiment we utilize the uncased BERT-base pre-trained text representation model (12 layers, 768 hidden vector size, 12 attention heads, 110M parameters), trained on the BookCorpus (800M words) (Zhu et al. 2015[4]) and Wikipedia (2, 500M words) corpora.

In both experiments we divided the data to train set and test set with ratio of 9:1, trained the model and examined the result probabilities on the test set before and after the transformation to calculate the CaCE.

For the classification task we employ a FC layer that receives as input the CLS token produced by BERT and produce predictions.
In the training phase we froze BERT's parameters.

Training the model for the movie reviews experiment was too heavy for our PC so we used randomly 10000 examples from the original dataset and trained only 3 epochs and it reached accuracy of 0.84 (Figure 9).

For the tweet experiment we trained the model for 20 epochs and reached accuracy of 0.72.

Because there are sentences that went untouched after the transformation we will compute the expectation $\mathbb{E}_g[\overrightarrow{z}\big(f(\phi(X))\big)|do(C_R = r)]$ weighted by the amount of changes each sentence got. e.g. if there are only 2 sentences and one got 2 changes and the other got 3, the weights will be 2/5 and 3/5.

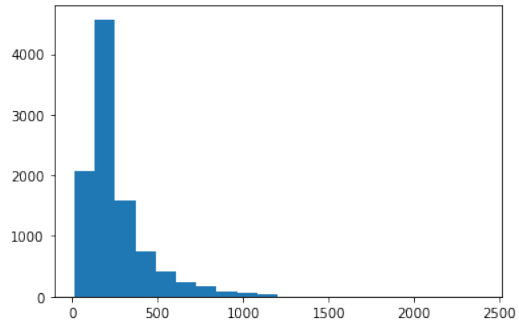

Figure 4: reviews length histogram

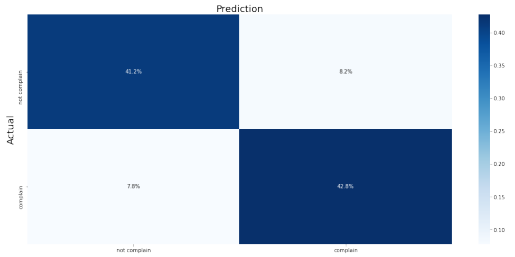

Figure 5: sentiment classifier (movie reviews) confusion matrix

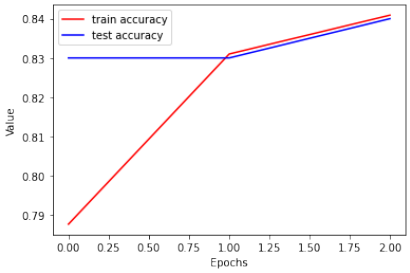

Figure 6: train and test accuracy of the sentiment classifier (movie reviews)

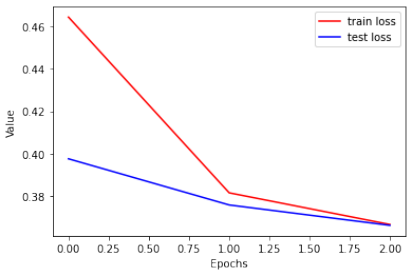

Figure 7: train and test loss of the sentiment classifier (movie reviews)

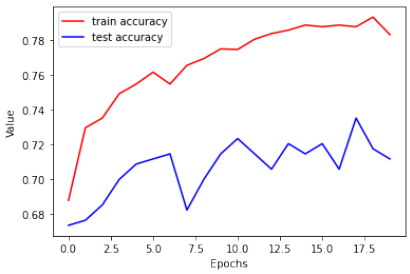

Figure 8: train and test accuracy of the complaint classifier (tweets)

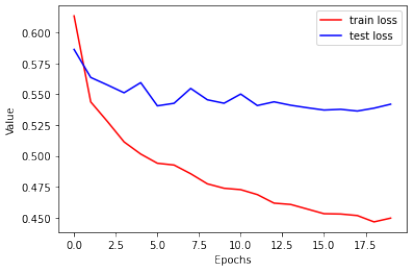

Figure 9: train and test loss of the complaint classifier (tweets)

Results:
For the movies experiment:

$$CaCE_{C_R} = \langle \mathbb{E}_g[\overrightarrow{z}\big(f(\phi(X))\big)|do(C_R = 1)] - \mathbb{E}_g[\overrightarrow{z}\big(f(\phi(X))\big)|do(C_R = 0)]\rangle = 0.00055$$

It means that high register texts are slightly more likely to be considered positive according to the model.
For the tweets experiment:

$$CaCE_{C_R} = \langle \mathbb{E}_g[\overrightarrow{z}\big(f(\phi(X))\big)|do(C_R = 1)] - \mathbb{E}_g[\overrightarrow{z}\big(f(\phi(X))\big)|do(C_R = 0)]\rangle = -0.0054$$

It means that high register texts are slightly less likely to be considered complain according to the model.

7 Discussions and possible weaknesses

- It seems that higher register texts are more likely to be considered positive and non complaint. Although the effect we measured is very small it can be due to our transformation process which does not change a lot in each text.
- The automatic text transformation we applied in our work, using an algorithm we created for this task, is not sufficient as it produces, in some cases, words replacements that do not hold logically. Therefore, for this task, there is a need for experts, in order to achieve better transformation.
- Under constraints of our computational resources, the model was not fully trained. We believe that with more powerful computation abilities, the sentiment classifying model could achieve a better accuracy and thus we could perceive more precisely, the average concept affect we wanted to examine.

- We can evolve our work by taking into consideration tagging of multiple types of registers. For example, we can distinguish types of registers like formal/informal, slang, vulgar etc.
- As we know and experienced in our work, there are words that may have different meanings in different contexts. Therefore, we believe it would be helpful to have a defined corpus of synonyms that can be replaced at any context, for example: wonderful $\rightarrow$ terrific.
- Even if register level affect such models significantly we should consider the possibility that it is not unwanted bias. That is to say, higher register text can actually be less negative than the same text with lower level register.

References

[1] A.Feder, N. Oved, U.Shalit, R.Reichart.

CausaLM: Causal Model Explanation Through Counterfactual Language Models, *arXiv:2005.13407v3*, 2020.

[2] J.Pearl. CAUSALITY: MODELS, REASONING, AND INFERENCE, *Cambridge University Press*, 2000.

[3] G.Sohsah, M. Esad, G. Onur. Classification of Word Levels with Usage Frequency, Expert Opinions and Machine Learning, *British Journal of Educational Technology, v46 n5 p1097-1101*, 2015.

[4] Y.Zhu, R.Kiros, R.Zemel, R. Salakhutdinov, R.Urtasun, A.Torralba, ,Fidler. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books, *arXiv:1506.06724v1*, 2015.