EVALUESERVE

Leveraging Deep Learning for ESG Research

Introduction

Environmental, Social, and Governance (ESG) refer to the three central factors used when measuring an investment's sustainability and societal impact on a company or business. These criteria help to better determine the future financial performance of companies.

At **Evalueserve** we create an **ESG Assessment Report,** a collection of ESG data based on the information available from sustainability reports, that scores a company based on specified ESG parameters.

Research Analyst downloads the publicly available PDF format reports from the company website and can manually read the report and extract relevant ESG information based on a pre-defined set of questions. The analyst captures the qualitative summary of the company's ESG performance and the overall ESG rating at individual criteria (E, S & G).

The overall goal of this project is to automate some/or all parts of the manual data collection and information extraction process by utilizing **Natural Language Processing (NLP)** and **Machine Learning (ML)**. This first phase focuses on building a solution that can extract complete text from PDF reports and identify potential answers to the supplied question.

Approach

We gathered our data from text extracted from PDF reports with different sections belonging to Environmental, Social, and Governance parameters. Based on a set of questions in the three parameters mentioned above, we need to find the answers from the text extracted.

To find the answers, we split the text into sentences and then computed the degree of similarity between the sentences, i.e., which sentence from the report has maximum similarity with the question. These steps are taken for every question. Even if the exact answer is not found, the solution's design ensures that an answer will be delivered to each question.
Sentence Similarity is one of the most popular NLP concepts that we can quickly build a model for, but a baseline model's accuracy will be affected by complex tasks since it could fail to capture the sentence's full context.

An advanced sentence similarity model must be used to determine how the proximity of two sentences are in surface closeness (*lexical similarity*) and meaning (*semantic similarity*).

Below is an example of how the advanced sentence similarity model is employed:

Sentence 1: Next EU summit will be held in Brussels

Sentence 2: Belgium to hold the European Union Conference

Here, if we look solely at the surface closeness of the words being used, these two sentences appear to be totally different from each other. However, if we also look at the semantic similarity between these two sentences, then it becomes evident that they share the same meaning.

A baseline model only captures the lexical similarity and is unable to identify the actual meaning behind the words, or entire phrase, in the right context.

Technical Stack

Instead of doing a word-for-word comparison, we also need to pay attention to context to capture more of the Semantics. Also, the traditional approaches like **Bag of Words** does not take care of the order of words in a sentence.

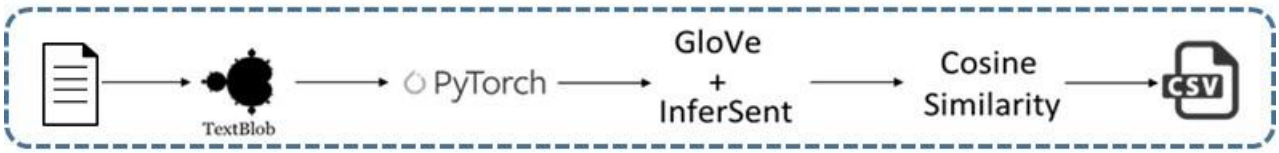
To overcome this complication, we employed advanced NLP techniques. This first is Word **Embedding,** which is one of the most popular representations of text vocabulary. It is capable of capturing the context of a word in a document based on parameters such as semantic and syntactic similarity, and its relation to other words.

Traditionally, we used to tokenize the sentence and average the vectors of all the words. These vectors can also be found in **GloVe,** developed by **Stanford NLP** team, which is a more advanced unsupervised learning algorithm used for obtaining vector representations in words. This technique is an improvement but loses accuracy by not accounting for the order of the words.

Then comes **Infersent**, a sentence embeddings method developed by the **Facebook Research** team that provides semantic representations for English sentences. It is trained on natural language inference data and generalizes well to several different tasks.

We will use GloVe **word2vec** embeddings with Infersent to generate the vector. So, going back to our ESG problem, the process now follows the sequence:

Figure 1: Updated ESG embedding process



The approach captures both the lexical and semantic similarity of sentences**,** which has allowed us to overcome the following challenges:

- **Underfitting larger datasets:** we were able to make datasets have a good fit for all data sets, as initial training was on small datasets.
- **Only returning the top matches sentences:** the system now returns multiple sentences since most answers have numerous sentences.
- **The algorithm was only answering some questions:** the algorithm now returns complete sentences to the answer, which better fits the context and eliminates the need for the operations team to do it manually.

Figure 2: The Python implementation of the Infersent model:

```
## Load the InferSent model
#V = 1
MODEL_PATH = 'infersent1.pkl'
params_model = {'bsize': 64, 'word_emb_dim': 300, 'enc_lstm_dim': 2048,
                'pool_type': 'max', 'dpout_model': 0.0}
infersent = InferSent(params_model)
infersent.load_state_dict(torch.load(MODEL_PATH))

## Load the Glove Word Embeddings data and train on InferSent
W2V_PATH = 'glove.840B.300d.txt'
infersent.set_w2v_path(W2V_PATH)
infersent.build_vocab(sentences, tokenize=True)
## Create dictionary with sentence & question embeddings
dict_embeddings = {}
for i in range(len(sentences)):
    print(i)
    dict_embeddings[sentences[i]] = infersent.encode([sentences[i]],
                                                     tokenize=True)

questions = list(df["question"])
len(questions)

for i in range(len(questions)):
    print(i)
    dict_embeddings[questions[i]] = infersent.encode([questions[i]],
                                                     tokenize=True)

## create backup copies - optional
train = df.copy()
dict_emb = dict_embeddings.copy()
```

This is a case of Unsupervised Learning. Since we don't have a target variable, we tried to find the answer from the text data by using various similarity metrics. In the diagram above, we retuned the sentence from the paragraph with a minimum distance from the given question.

Based on the similarity metric, we choose the sentence with a minimum distance from the question as our answer. We can further improve this model by adding more training data and some more NLP features. Additionally, we can also leverage Feature Engineering to improve the overall accuracy. Next, we can treat this as a **Supervised** problem by adding the answers as the target variable.

With small tweaks, this application can be used to create a **Question Answering system**, **Chatbots,** and a **Text Summarization** solution. Once implemented, this can significantly reduce the amount of manual effort needed and save costs.

These solutions are a precise stack of programming powered by various NLP tools, which are then combined with robotic process automation (RPA) and deep learning techniques. Once completely developed, this tool will take the research service industry to new heights with a combination of increased efficiency and lower economic impact.

Authors

Somesh Kumar

Somesh has 17 years of experience in operations experience, building large teams that deliver Machine Learning, and involvement in AI projects. He has consulted with Fortune500 companies and has offered a clear view on leveraging AI, ML, and digital technologies to solve complex problems and enhance business transformation.

Amit Kalra

Amit has 15 years of experience in data sciences, natural languages, deep learning, AI, and statistical modeling. Amit is a computer science postgraduate and a certified PMP, Prince 2, AWS Solution Architect, and a Machine Learning professional.

Manoj Kumar

Manoj is a manager in analytics, with 8 years' experience in data science, ML, and business analytics. He has experience in various domains such as banking, contact centers, life science, and telecom. Manoj is currently a postgraduate for Applied Statistics in IGNOU Delhi.

Sachin Kalra

Sachin has 7 years of experience providing data and analytics solutions using advanced analytics and AI to private equity, merchant banks, and investment management firms around the globe. He has built solutions in customer understanding, investment analysis, strategic decision-making, NLP, and data ingestion pipelines.

ABOUT EVALUESERVE

Evalueserve is a leading analytics partner to Fortune500 companies. Powered by mind+machine™, Evalueserve combines insights emerging from data and research with the efficiency of digital tools and platforms to design impactful solutions. A global team of 4,000+ experts collaborates with clients across 15+ industries.

CONNECT WITH US

Connect with us on

If you are interested in speaking with Evalueserve about how your organization can adapt for tomorrow, please contact us at info@evalueserve.com or for more information, visit www.evalueserve.com.

EVALUESERVE
POWERED BY MIND+MACHINE