Helvetas Project Analysis Tool

24.11.2021

Catalina Dragusin
Tom Haidinger
Jackson Stanhope
David Simon Tetruashvili

ETH Zurich, Analytics Club, Hack4Good 2021.

Introduction

This project was an application of Natural Language Processing (NLP) to assist Helvetas (a Swiss NGO) with interpreting internally generated reports. These reports (of which over 1000 were provided to work with) contain a large amount of human written text that pertains to the progress of specific projects within Helvetas. These reports may be written in an arbitrary language, and would need to be translated into English.

The motivation for using NLP to interpret these reports was to better allocate resources within Helvetas. The data contained within a report could provide a better insight into common issues or trends across projects, as well as better depicting which departments within Helvetas could be involved in a project.

The project was subdivided into two main tasks: Assigning each project outcome to a specific department, or Working Field, within Helvetas, as well as finding and assigning topics, or Accents, to each project description. These topics were generated by finding commonalities between all projects in the set.

Translation and Preprocessing

Similar to any other Natural Language Processing (NLP) pipeline, the first step was to preprocess the data. As Helvetas is characterised by a decentralized structure with projects all over the world, the languages that are used to report the details and outcomes of the projects are diverse, including English, French, Spanish and others. Facing the challenge of a multilingual corpus, the chosen approach was to translate all the data to English before the other preprocessing steps. This was handled by using the Google Translate API, which is also able to detect the source language. The other preprocessing steps that followed are: removing the stopwords and punctuation, lemmatization.

Accents Task

Overview

For this first main task, Helvetas asked us to develop an algorithm that identifies trends both on an individual project level, and on a global level across the entire set of projects. This would be able to support Helvetas staff in identifying hidden opportunities to help or expand specific projects (individual project level) or help identify needs to expand capabilities within a general area for their organisation.

Method

The specific implementation we used was an LDA (Latent Dirichlet allocation), and its topic discovery and classification capabilities (where a topic is a set of terms that, taken together, suggest a shared theme). The two sub-models this implementation creates are the word–to–accent model and the accent-to-project model. The first sub-model returns a list of weighted words that suggest a topic while the second sub-model returns a likely-hood distribution between the aforementioned topics and the Helvetas projects. This approach discovers topics based on the co-occurrence of individual terms (after preprocessing), however assigning a meaningful label is up to the user and often requires specialised knowledge. After creating our models, we therefore simply label them Accents and let the weighted word list speak for themselves. To allow an easy intuition for the topic meaning we display word clouds next to the Accents to visualise their word weightings.

Results and Limitations

What this gives us is an Accent discovery and classification tool that's easy to fine tune and very fast. The accents were found to be relatively distinguishable, however we did see frequent repetition of words such as: "improve, project, management" that occurred highly weighted in many Accents. As a group we did not want to incur our own biases upon the classification, but instead build multiple tools to leave that up to the user. These tools included the ability to add additional stop words (words to filter out in preprocessing) if they were very common and did not contribute to (or even took away from) distinguishing between Accents. Furthermore, we show the user a pie chart of the Accent allocations that can inform a decision on how to refine the model. This can be done through decreasing the amount of Accents discovered if the words associated with Accents are not discrete enough between topics, and can be increased if a few topics seem to cover all projects. This number can be varied between 1 and 10.

A reoccurring result we found was that Accents associated that suggested the theme water management seemed to cover a significantly large percentage of the projects. This result however need not invalidate the model usage however, because it might simply indicate that most projects could benefit the most by increased/improved water management (or whatever the user associates with this specific example).

A limitation is that we cannot predefine possible accents or use words not in the texts in such Accents. However we believe this can help more effectively distribute the allocation of resources on individual projects and on the organisation-wide level.

Future Work

An option to reduce the number of Accents between 1-10 can significantly increase coherence of word induced themes. The class methods for this have been written, but not implemented in the GUI at the moment.

WF Classification Task

Overview

The second main task in this project is the classification of a given Project Outcome into one of the Working Fields (WFs) within Helvetas. A project may have up to five Outcomes which themselves are usually free natural text of about 50-100 words. The language of this text is not fixed and was predominantly French in the context of this project. In total, 42 human-labelled (Outcome, WF) examples were given over 10 distinct Projects.

Method

Due to the scarcity of labelled examples, we have focused on unsupervised approaches first. This led us to use nearest word-embedding classification as the base solution. We would embed the literal of each WF (e.g., "WASH and Water Governance") into a 300-vector representation and compute the average (or centroid) embedding of the non-stop-word members of the Outcome we are to classify. The classification is then the WF with the closest vector representation to this centroid via some metric. In our testing we have used both Euclidean distance and Cosine similarity to gauge how distant a centroid is from a WF representation. This approach proved to be susceptible to influence from common words such as 'governance,' 'project,' 'WASH,' etc. to be useful since they may appear in an Outcome arbitrarily attributed to a WF and hence make the embedding centroid be less discriminative (close to all WFs) thereby enforcing high entropy (and low predictive power).

Seeing that an unsupervised approach would need a more extensive preprocessing which would rely on our judgement, it was preferred to move to a supervised approach to avoid injecting author-bias with said preprocessing. The current approach relies on the assumption that the joint distribution over a Project's overall WF assignment (which is available for all given projects - i.e., c. 1000 labelled examples) and its natural-text description is similar enough to the true unknown distribution of a given Outcome and its WF assignment that using one to infer the other is feasible and justified.

Using this assumption we suddenly had access to a much larger dataset of over 1000 examples on which we could successfully train a supervised model. The model itself is a Linear Support Vector Classifier trained on the TF-IDF features of the descriptions. We use the Classifier Calibration with Cross-validation to create confidence estimates for the Linear SVC predictions.

The model is re-trained on the (Project Description, Project WF) dataset each time the input data is updated. We do not save a pre-trained model to increase robustness to unseen data. This does have the disadvantage of creating overfitting if the input data is unbalanced in its Project WF representation (e.g., if some WFs do not appear in the input data, they will be less likely to be predicted by the model).

Results and Limitations

In our tests, the trained model achieves a 90% accuracy on classifying the overall WF of an unseen Project via its Description. As to the 42 given (Project Outcome, Outcome WF) pairs, the model correctly classifies about 50% of them. A reason may be that most given pairs are originally in French while the majority of Project Descriptions are in English and we are experiencing information/ discrimination loss during the translation step. Other reasons could be bias in the human-labeled data itself and the imperfect assumption that the relationship between descriptions and WFs is the same as the relationship between the Outcomes text and the WFs.

We should add that this is significant in interpreting the result of our model. If the user can notice a severe discrepancy between the language used in the Project descriptions and Project outcome (differing languages aside), we advise to be sceptical of our model's output.

We should also note that in our initial integration tests, applying the model to all of the given data is known to produce results which are skewed to the "WASH, and Water Governance" WF. We suspect that this issue occurs in part for the same reason as the failure of our unsupervised model as well as terms such as 'WASH' and other abbreviations being translated as is, and also containing non-literal information (e.g., the term 'WASH' has nothing to do with the literal act of washing something).

Future Work

An obvious first step in future work is the development of a more robust and sophisticated data cleaning tool. As discussed, our methods suffered from noisy data, be it by frequently occurring words which diluted the lexical meaning of the overall text blob resulting in low discriminative power between the WFs, or by information loss during translation. We have implemented simple solutions within the Accent assignment task which allows the user to exclude a set of words from input data if they feel a word's prevalence in the output is muddying up the result. The same was implemented here, however due to the fact that it is harder to visualize which words are more responsible for a given supervised model classification, we decided against its inclusion.

One could develop an automatic, perhaps entropy-based filter to remove such prevalent lexical-meaning-diluting words from input data. This would implicitly improve any supervised approach and make it easier to apply the data in other approaches, all bypassing the need for used input. Of note is that we believe that this step would greatly increase the efficacy of our initial unsupervised approach.

Additionally, one could also investigate the underlying assumption that the Project Description and Outcome distributions are truly similar (language difference aside). This proved to be out of the scope of this project, but answering this would also enable a more educated choice of model in the future.

To add to this and the previous point, we noticed that there were some Outcome examples which carried zero to none information about the assigned WF, for example, we noticed a number of Outcomes containing the single digit "1" while belonging to projects of different WFs. Our solution does not filter these before making a prediction which in all likelihood does dilute the result.

Returning to the unsupervised approach, one glaring flaw in using WF literals to compute their embedding representation is that the WF literals themselves share key words (e.g., the word ‘governance’ is present in “WASH, and Water Governance” and “Governance and Civic Spaces” WFs and is used in different meanings). This results in the representations appearing closer to each other than they should be based on the presence of a single shared word. A better approach would find the embedding representations of WF in a way which does not only rely on the word members of each literal. For example, contextual word embedding would be a good fit, as would be training an architecture like BERT to produce implicit representations based on the Description data for example.