ETHICAL CONSIDERATIONS IN MULTILABEL TEXT CLASSIFICATIONS

An Australian community sector case study on the quest for explainable AI

MELBOURNE DATA ANALYTICS PLATFORM

SmartyGrants
Software, data science & grantmaking intelligence

An enterprise of:
ourcommunity.com.au
Where not-for-profits go for help

Ethical considerations in multilabel text classifications

An Australian community sector case study on the quest for explainable AI

A SmartyGrants white paper, April 2021

Report author: Kabir Manandhar Shrestha, Innovation Lab

Reviewers: Paola Oliva Altamirano, Nathan Mifsud, Sarah Barker, Edoardo Tescari and Daniel Russo-Batterham

This project was completed in partnership with Melbourne Data Analytics Platform (MDAP) at the University of Melbourne. Building on the success of this collaboration, MDAP is keen to continue to engage with the wider community.

Published by Our Community Pty Ltd, Melbourne, Victoria, Australia

SmartyGrants: Software, data science and grantmaking intelligence

Executive Summary

Tracking the flow of funding and other support to social sector organisations in Australia has always been challenging. This is in large part due to inconsistencies in categorisation, or the absence of categorisation entirely. To mitigate this, Our Community has developed CLASSIE (https://www.ourcommunity.com.au/classie), a classification system for social sector initiatives and entities, and CLASSIEfier (https://www.ourcommunity.com.au/classiefier), an auto-classification algorithm that uses CLASSIE to classify grant applications (and other social sector text).

We strive for Our Community's algorithms to be as robust and accurate as possible. Upon releasing CLASSIEfier in beta, we have taken immediate steps to identify possible biases and taken steps to ameliorate them.

Data scientists inevitably bring biases into the algorithms they write, but these are not always interrogated. '*The model's accuracy is almost 90%. Everything is fine.*' While this approach may be acceptable in some contexts, it's not good enough for algorithms that will drive decision-making at scale. It's even more troublesome for an algorithm that serves as a universal classification system for Australian social sector initiatives and entities – the results of which may influence funding decisions. This article describes the steps taken to dissect the CLASSIEfier algorithm and explains how the model can be improved.

We explored different types of biases that creep in during auto-classification: word biases, algorithmic biases, user biases and sensitive categories biases. We found different ways to tackle specific scenarios. However, our main recommendation is to maintain algorithm transparency and communicate with the user base about the possible gains and pitfalls of using auto-classification.

Introduction

Most people are familiar with the concept of a grant. However, if asked to write a grant application, the average person wouldn't know where to begin. Writing a grant application is complex. It's especially hard to write one with a high chance of being approved and fully funded. A well-written grant application demonstrates need, presents evidence and tells a compelling story. This means that grant applications contain a lot of text-based information. Manually deriving insights from the thousands of applications written annually (let alone hundreds of thousands of historical grant applications) is extremely time-consuming. CLASSIE, a social sector taxonomy, assists grantmakers in grants categorisation, easing the burden of reporting and statistical analysis.

The taxonomy separates information into several streams, including:

1. Population: Usually describes the direct beneficiaries of the grant. For example, chronically ill people, children, or Indigenous people.
2. Subject: The domain embedded in the grant application. For example, health, arts & culture, sport & recreation, or science.

CLASSIEfier is an algorithm that can auto-classify grant applications using the CLASSIE taxonomy. CLASSIEfier is a keyword-matching model that allows white box classifications, light deployment and easy maintenance. The keywords it uses encompass a controlled vocabulary drawn from existing grant applications that are reviewed by social sector experts.

The heart of the algorithm is the Population and Subject dictionaries that contain words relevant for the categories in each stream. For instance, the category *Chronically ill people* is associated with words such as *diabetes, arthritis, lupus, sclerosis,* and *asthma*. The algorithm identifies whether these keywords are used in a grant application. To further understand the keyword-matching algorithm, refer to our article which explains CLASSIEfier (https://www.ourcommunity.com.au/classiefier).

Types of bias

As humans, we all hold biases (even when we aren't aware of them). Algorithms inherit biases from their human authors. It's important, therefore, that algorithms don't define the decisions we make. Biases in an algorithm that classifies grant applications could influence funding decisions and adversely impact individuals and communities as a result.

The aim, then, is to create an algorithm free from bias. In practice, eradicating bias might be impossible. Still, we can strive for perfection, while being transparent about the biases we notice, and sharing the decisions that have been made to mitigate them. We aim to inform our users and adopt feedback if they identify issues. This, in turn, builds trust in our results, as the model will become more accurate over time.

Given our context, we categorise biases into four possible types:

1. **Word bias:** The biases introduced by words included in the dictionaries and/or in the text of the grant application that affect the algorithm.
2. **Algorithmic bias:** The biases introduced by the algorithm's design.
3. **User bias:** The biases introduced by the user while writing/assessing the grant application.
4. **Sensitive categories bias:** The biases inherent in the categories — such as ambiguity, overlaps or gaps — that make classification difficult (for the algorithm as well as for humans).

In the following sections we provide examples to better illustrate these four types of bias.

Are the grants being classified?

The first test for any text classification model is to see if a given text (in our case, a grant application) is actually being classified. The model is useless if it can't categorise the information that is being inputted. A total of 17,724 grant applications were selected for analysis. We selected applications belonging to a handful of unique grant domains to give the sample variety and balance.

Figure 1 shows the sample divided into classified and unclassified grant applications. The grey bars show grant applications that were classified by the algorithm and the red bars show unclassified grant applications. A total of 16,155 applications (91%) were classified, with 1,569 applications (9%) not classified.

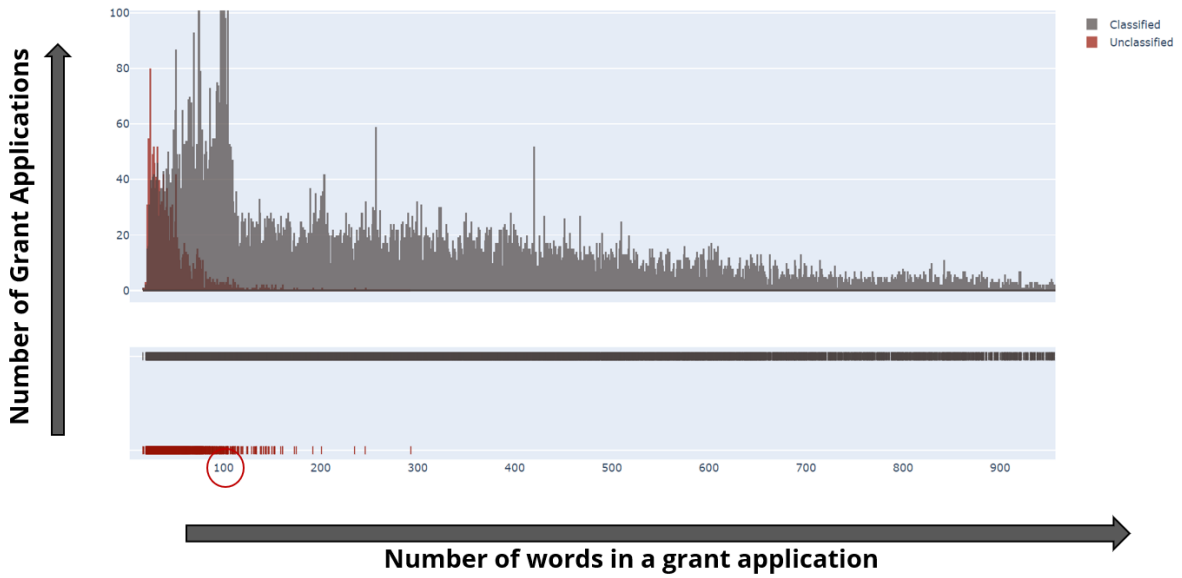

Figure 1: Number of grant applications (classified and unclassified) by length of their text.

Looking at the number of grant applications per number of words in the text, you can see that the number of words in classified applications is much higher. A total of 72% of the classified applications (11,636) had more than 100 words, while 96% of the unclassified applications (1,504) had fewer than 100 words. This implies that the algorithm may not have extracted enough information from short grant applications to enable classification. In future versions of CLASSIEfier, we will look for solutions that will enable us to better classify short text.

If we further split the sample into those that remained unclassified by Subject and those that remained unclassified by Population, we found that 10% of the applications were missing a Subject category and 30% were missing a Population category. From this, we can deduce that the dictionary for Subjects as it stands can effectively identify categories of grant applications, whereas the Population dictionary may warrant some improvement.

Figure 2 gives a good sense of the relative performance of the two dictionaries. The green bars show the grant applications that were *not* classified by Subject, while the blue bars show those that were *not* classified by Population.

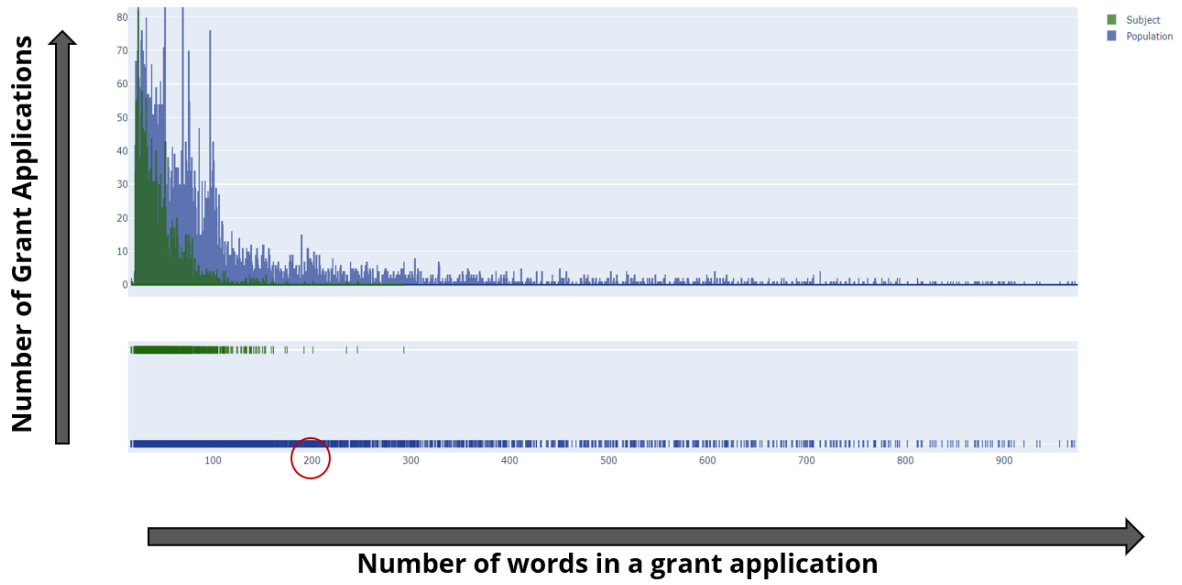

Figure 2: Number of unclassified grant applications by length of text, separated by missing Subject (green) and missing Population (blue).

Similar to the overall view of applications in Figure 1, the breakdown of unclassified applications in Figure 2 shows that a key reason that Subject was missing was not for lack of categories but because the input text was too short. This suggests that the dictionary for Subject is reasonably complete and can classify many types of applications given sufficient input.

However, the scenario for Population is quite different. Although the majority of grant applications missing a Population category (65%) had fewer than 100 words, a significant minority (35%) did have more than 100 words. This suggests that while application length is important, the Population dictionary may have other shortcomings. For example, there are many synonyms for words that describe Populations, which makes building a comprehensive dictionary difficult. These missing words/categories from the dictionary provide an example of what we consider **word bias**.

Also of note is that a significant portion of grant applications qualify as 'Universal', meaning a grant application is not targeting a specific group or population. A different sample with a pool of 22,699 applications that were *manually* classified by users showed that 5,814 (25%) were classified as Universal. Currently, the algorithm does not have a way of identifying Universal grants, so it could be that a large proportion of the 30% of applications missing a Population category are Universal.

Do the classifications make sense?

The next test was to validate the classifications suggested by the CLASSIEfier algorithm. It's not feasible to individually review each of the classified grants to confirm that the categories make sense, but we can perform spot checks to identify critical issues. However, a handful of spot checks is not sufficient to conclude that the algorithm is accurate. To come up with a reasonable evaluation of the model's classification, we selected a collection of grant applications known to be in a particular domain. The alignment between the expected domain and the categories assigned to the grant applications by the algorithm provides a fair assessment of its credibility.

Figures 3 and 4 reflect grant applications known to be in the *Animal welfare* domain. The majority of the Subject and Population categories identified for these grants were expected to be animals, which was the case. Figure 3 breaks down the classifications by Population categories. We can see that the dominant categories were *Companion animals*, with 212 grants, *Animals*, with 164 grants, and *Working animals*, with 158 grants.

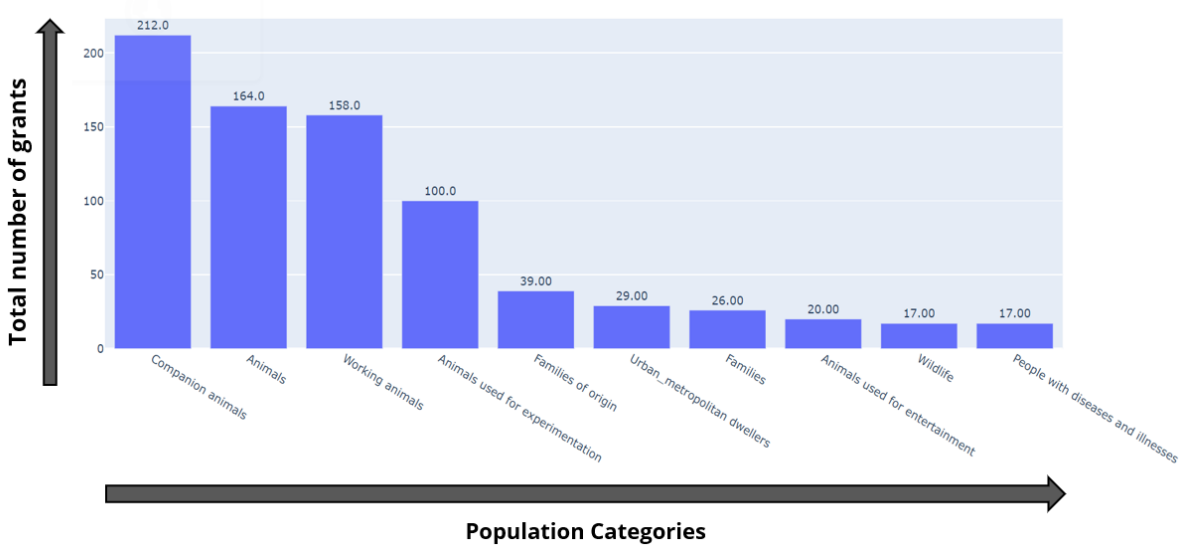

Figure 3: Classified grant applications in *Animal welfare* by Population categories.

Meanwhile, Figure 4 depicts grant applications classified according to Subject categories. The dominant categories were *Animal adoptions*, *Animal rescue and rehabilitation*, and *Animal training*. The results for several other domains showed similar consistencies; that is, the categories aligned with the expected domains. This is a positive result and provides evidence that the model is reasonably accurate.

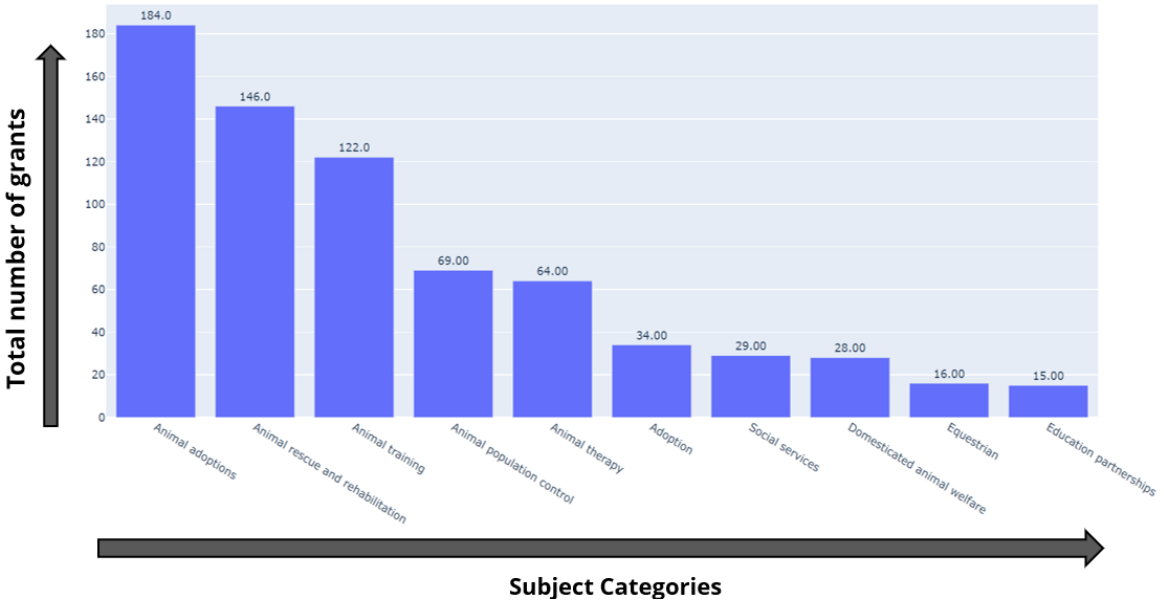

Figure 4: Classified grant applications in *Animal welfare* by Subject categories.

Figure 4 further reveals an instance of **sensitive categories bias**. The algorithm selected the category *Adoption* (which is meant for children), when a grant application was in fact related to *animal adoption*. We corrected this issue by adding exclusion words to the *Adoption* category, so that when animals are mentioned in a grant application the *Adoption* category is not triggered. This demonstrates a clear advantage of having a white box algorithm because once detected, many issues can be easily corrected.

Typographic errors

Spelling errors are inevitable. Often grant writers are time-poor (many are volunteers), working to a deadline and may have non-English-speaking backgrounds. For instance, consider this project title: 'Reducing *Defrestation* and Forest *Degradtion* can help in slowing down climate change' (we have added italics to highlight the misspelled words). This is an example of **user bias**. If words are misspelled, the algorithm can't find the categories related to those words.

We trialled various approaches to address this issue. We began by looking at several off-the-shelf tools for spelling error correction. While these tools remove typos efficiently, we realised that the tools themselves are heavily biased. Some of the biases include not using the entire context or sentence to correct words, replacing less-probable words with more-common words, and being limited to the vocabulary of the tool itself. For instance, in the sentence 'One of the *carers* in the child centre was diagnosed with COVID', the

word ‘carers’ was transformed into ‘careers’. Hence, while these tools may potentially address some biases, there is a very real chance that other biases would be introduced.

Next, we experimented with a more complex context-based spell corrector. While this approach avoided several of the introduced biases mentioned above, we became aware of other biases. For instance, while these tools were efficient at preserving the meaning of the text, in some cases, the tool autocorrected words incorrectly. The word ‘physio’ in the sentence 'Due to the club focusing on contact sports more than before, we need more *physio* to support athletes' — referring to physiotherapy — was transformed into ‘physician’. This indicates that there is a risk of context-driven tools replacing words that are in the algorithm’s dictionaries with words that are not.

We decided that a more reliable approach might be to combine the two types of tools mentioned so far. However, this approach came at a cost. We discovered that the computational time taken to spell-correct a grant application took as long as classifying the application itself. Despite the additional time taken to process the applications, the combined algorithm was not error-free. It was for these reasons that we decided not to incorporate a spell checker into the algorithm at this stage. We hypothesise that in a 300-word text, even if some words are misspelled, the majority of correctly spelled words will be sufficient to trigger a given category.

Ambiguity in language

> *'There is no greater impediment to the advancement of knowledge than the ambiguity of words.'* — Thomas Reid

Language is complex and words can be ambiguous. A word’s meaning can change depending on its given context. A text classification model can struggle to differentiate the context of individual words. For instance, in the sentence 'The *space* used by chronically ill people has been overbooked', the word ‘space’ might be associated with the *Aerospace engineering* category. Another example is 'Several of the patients have returned to *sound* health after taking part in this program', where the word ‘sound’ could be associated with the *Audio* category. This is another type of **word bias**.

Our algorithm mitigates ambiguity by requiring word matches in different groups (topic and context), incorporating an exclusion group, and enforcing a minimum number of word matches to trigger a category. Following this model, for ‘space’ to link a given application to *Aerospace engineering*, it would need to be accompanied by other context words and comply with the minimum number of matches.

Abbreviations everywhere

It is common to replace words with their abbreviation. This can be a blessing, as it reduces repetition and helps when restricted by a word count. But for a keyword-based algorithm, abbreviations are problematic. If words that define categories are replaced by their abbreviations, the algorithm has difficulty picking them up. For example, in the sentence 'VPN issues have made transferring information on time difficult due to the lockdown', the acronym for Virtual Private Network is a strong indicator of the technology aspect of the grant application. The only way to overcome this **word bias** without changing the design of the algorithm is to add all relevant abbreviations to their respective categories, something that is currently not feasible.

Not all words are equally important

We present here a form of **algorithm bias**, which is more nuanced than the preceding issues. Let's start with an example sentence:

> 'The lack of proper care for children struggling with autism is due to fewer staff. These children require constant care and given fewer human resources, children aren't cared for well.'

When you read this, you easily understand its meaning and understand that grant applications can be described by more than one category; in this case, the grant application could be classified as serving both *Children* and *People with autism*.

How would you respond if asked to choose a single classification? It is difficult to know which category is more important. This challenge is present when designing an algorithm that allows the user of the model to set limits on the number of categories returned — some users choose to return only a single label.

To the model, the example sentence is just a combination of words. If you ask it to assign only one Population category, that category will likely be *Children*, because 'children' was used three times in the sentence. However, to our understanding, 'autism' is salient despite only appearing once. When grantmakers ask for a limited number of categories to classify their grants, it is difficult for the algorithm to rank which category will be most relevant in each case. We certainly don't want categories associated with common keywords to dominate our classification. A path forward is discussed below.

Areas of improvement

In the previous sections we discussed the current state of the CLASSIEfier algorithm. Although CLASSIEfier efficiently classifies grant applications with acceptable accuracy, we identified some of the biases that exist and to what extent we have addressed them. The following sections offer suggestions for future improvements.

TF-IDF to the rescue

Term Frequency – Inverse Document Frequency (TF-IDF) is a common and highly effective technique used to identify important words that characterise a document. Intuitively, we might think that if a word appears frequently in a document, it is because it is relevant to the topic at hand. We also know, however, that words that are common across many documents are part of normal writing and may have little to do with the topic of an individual document. Let's look back at our previous example: *'The lack of proper care for children struggling with autism is due to fewer staff. These children require constant care and given fewer human resources, children aren't cared for well.'* In this sentence common words such as 'The', 'of', 'for' and 'is' aren't helpful in defining the document's domain. However, words such as 'autism', ' care' and 'children' define the domain well. TF-IDF comprises two parts:

1. **Term Frequency (TF):** This part of the formula is the raw count(f) of the occurrence of a word/term(t) in a document(d) (in our case, a document is a grant application).

$$TF(t,d) = f(t,d)$$

2. **Inverse Document Frequency (IDF):** This part weights how common or rare a word is across all grants. This means it is a measure of the degree to which a word should influence the classification. IDF assigns greater significance to words that are rare relative to the corpus, which 'penalises' more common words.

$$IDF(t,D) = \log \frac{N}{|\{d \in D,\ t \in d\}|}$$

where **D** = All the documents(grants), **N** = Number of documents i.e., N=|D| & $|\{d \in D,\ t \in d\}|$ = number of documents where the term t appears. If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to adjust the denominator to $1 + |\{d \in D,\ t \in d\}|$

TF-IDF combines the two components in the following way:

$$TF - IDF(t, d, D) = TF(t, d) \bullet IDF(t, D)$$

We use these two measures to assign to each keyword a combined score. The score reflects how many times the keyword was repeated and how rare/unique this keyword is. To calculate the IDF weights for the keywords in our CLASSIE dictionary, two approaches were taken:

1. **Dictionary-based approach:** For this approach, we defined each CLASSIE category as a document and the keywords defining that category as its content. This meant that words that were unique to a particular category were given more weight/significance, while context and general words were assigned lower weights.
2. **Corpus-based approach:** For this approach, around 500,000 grant applications (the 'corpus') were used. Each grant application was filtered for only words that were relevant to us, i.e. those defined in the Population and Subject dictionaries. We then calculated the keyword weights using the whole corpus. Deriving the weights in this way meant that words that were unique across the documents would be considered more significant than common words such as 'children', 'family', and so on.

As mentioned previously, it is impossible to design an algorithm that avoids human biases. Now that there are two sets of weights for each keyword, which one should we choose? Neither will be perfect. This decision can only be made after running statistical tests and comparing the two options side by side.

What's more important: rarity or frequency?

Let us state clearly that the discussion to follow is highly subjective. Once we have the Term Frequency for each word in an application and our Inverse Document Frequency scores for words, we can use the two to compute a combined score. Let's explore what would happen in the example provided earlier, where the word 'children' was repeated three times while the word 'autism' was repeated once.

After calculating the IDF (rarity) scores across grant applications, the relative weight for 'children' was 2.5 and 'autism' was 6. In other words, 'autism' was given more than twice the weight of 'children'. If we assume that Term Frequency is equally important as Inverse Document Frequency, the combined scores would be:

```
3 (TF) • 2.5 (IDF) = 7.5 for children
1 (TF) • 6 (IDF) = 6 for autism
```

As you can see, using this approach, the word 'children' still seems to be more significant than 'autism'. To reduce the effect of Term Frequency (TF), we propose the following formula:

$$\left(\min_{TF} + \left(\left(TF - \min_{TF}\right) \bullet weight\right)\right) \bullet IDF = Score$$

where $\min_{TF} = Lowest\ Term\ Frequency\ of\ the\ relevant\ words$ & ***weight*** *= penalizing factor for frequency. When weight = 0, we only look at IDF as for all words* $\left(\min_{TF} + \left(\left(TF - \min_{TF}\right) \bullet weight\right)\right) = \min_{TF}$. *When weight=1, we get* $TF \bullet IDF$.

Using the above formula to the previous example, $\min_{TF} = 1\ for\ Autism$. let's take *weight = 0.1 (penalize TF heavily).* We have:

```
(1 + ((3-1) • 0.1) • 2.5 = 1.2 • 2.5 = 3 for children
(1 + ((1-1) • 0.1) • 6 = 1 • 6 = 6 for autism
```

The above approach results in the word 'autism' being more significant than the word 'children'. Whether or not to apply a weight to penalize Term Frequency, and if so, what should the weight be, is a judgement call. If a decision is made to incorporate this measure, the algorithm's performance will need to be further tested against real data.

Hierarchy: the final topic of investigation

Much thought has gone into the hierarchy of the CLASSIE taxonomy. To summarise, the hierarchy is arranged so that more generic/common categories occupy the top and more specific categories emerge as we approach the bottom. An example of the hierarchy can be seen in Figure 5, where *Health* is a generic category at the top of the taxonomy and *Breast cancer* and *Lung cancer* are specific forms of cancers beneath it.

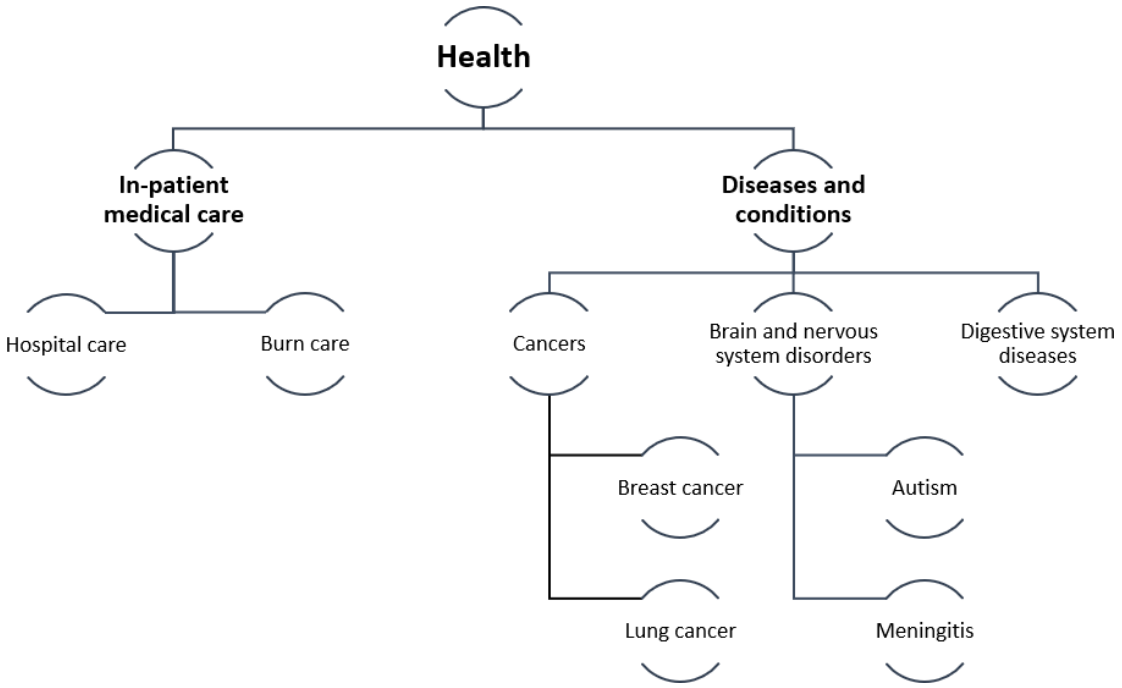

Figure 5: Snapshot of the CLASSIE hierarchy.

Although the hierarchy holds important information about the relationship between categories, the algorithm so far hasn't used the hierarchy to improve its classification — the categories are treated as a flat list. The hierarchy could be used to improve the model's accuracy. For example, say a grant application receives the following TF-IDF scores:

```
Initial TF-IDF ranking:

1.    In-patient care: 120
2.    Lung cancer: 80
3.    Breast cancer: 70
4.    Cancer: 50
5.    Astronomy:20
```

If we look only at the TF-IDF scores, we would conclude that the application relates to *In-patient medical care*. However, if we combine these TF-IDF scores with knowledge about the hierarchy, we can see that the first four categories are all related to the category *Health*. Likewise, we will also be able to infer that the three categories *Cancer*, *Lung cancer*, and *Breast cancer* are closely related to each other. The combined score of those three categories is greater than the score for *In-patient medical care*, which could be used to place these more accurate categories at the top of the ranking. Thus, we may conclude that the document is about *Cancer* in particular and *Health* in general.

New TF-IDF + hierarchy ranking:

1. **Cancer: 50**
2. **Lung cancer: 80**
3. **Breast cancer: 70**
4. In-patient care: 120
5. Astronomy:20

(Cancer related combined score = 200)

In addition, by incorporating knowledge of the hierarchy, we may be able to automatically identify anomalies. In the example above, the category *Astronomy* is not related to any of the other categories and is possibly an irrelevant label. As we have seen previously, ambiguous words can result in such anomalies.

This hypothetical scenario suggests that there is potential to improve the algorithm's accuracy by incorporating knowledge of the CLASSIE hierarchy. Of course, again, this approach needs to be tested.

Conclusion

We hope this report sheds light on several types of biases that may be present in auto-classification algorithms and in particular keyword-matching algorithms.

It may be of interest to note that in building CLASSIEfier, earlier versions considered the application of machine learning, neural networks and transformers. However, these approaches were abandoned because there was insufficient labelled data for each of the 1,100 categories, particularly in niche areas (which introduces the risk of compounding the disadvantage of vulnerable groups). Added difficulties arose from the hierarchical nature of the CLASSIE taxonomy and the multi-label approach to classification. As it stands, the keyword-matching approach outperforms machine learning and has the added benefits of transparency and ease of maintenance.

We are aware we haven't explored in detail mitigation of sensitive categories biases. We may delve into that topic in future.

We are keen for feedback. Are there any biases you've identified that we didn't cover? Let us know. Our next step is to auto-classify grants in SmartyGrants and solicit feedback

from grantmakers on its usefulness and accuracy. We intend to continue testing and improving CLASSIEfier over the coming years.