World Patent Information 61 (2020) 101961

Contents lists available at ScienceDirect

World Patent Information

journal homepage: www.elsevier.com/locate/worpatin

Construction and evaluation of gold standards for patent classification—A case study on quantum computing

Steve Harris [a,*], Anthony Trippe [b], David Challis [a], Nigel Swycher [a]

[a] Aistemos Ltd, 39-41 Charing Cross Road, WC2H 0AR, London, UK[1]
[b] Patinformatics LLC, 565 Metro Place S. Suite 3033, Dublin, OH 43017, USA[2]

ARTICLE INFO

Keywords:
Patent classification
Evaluation
Artificial intelligence
Information retrieval
Deep learning
Gold standard

ABSTRACT

This article discusses options for evaluation of patent and/or patent family classification algorithms by means of "gold standards". It covers the creation criteria, and desirable attributes of evaluation mechanisms, then proposes an example gold standard, and discusses the results of applying the evaluation mechanism against the proposed gold standard and an existing commercial implementation.

1. Introduction

There are a number of problems in the strategic patent decision making and portfolio management domain where artificial intelligence techniques can be applied. One of the more common is that of mapping patent assets to technologies, for example to perform patent landscaping, or for reporting on the contents of your own, or competitor portfolios. This is also one of the hardest tasks to perform mechanically, and has been identified as a source of friction in strategic patent decision making [1].

Conventional "mandrolic", or semi-automated solutions typically revolve around performing a boolean search over the assets to discover a superset of the assets to be identified, then manually reviewing returned results to determine if each individual asset falls into the desired class.

There are a number of compromises involved in this approach — predominantly related to the time taken to perform a thorough review of the technology domain, or the cost of outsourcing this work to external experts.

In addition there is also the issue of inconsistency of results from month to month, as the output of manual review by different individuals can be highly variable. In a study conducted by Elextrolux [2] across 29 outsourced patent search service providers it was found that there was a high degree of variability in the results. The requested search was "LED lighting of handle for refrigerator", which was believed to be precise enough to make interpretation of scope a minor factor. In total, across the 29 providers there were 194 distinct patent families identified, of which 114 were deemed to be relevant to the scope of the query by independent review. Within the relevant families 19 were identified as being highly relevant, and the number of those identified by a single provider varied from one to twelve, with a median of 4 and a mean of 5.2.

Because of these factors, automation of this process would be advantageous to the industry, resulting in more consistent reporting, and freeing up subject matter experts to work on higher value projects. As this article will show, measuring the accuracy of Machine Learning algorithms in a neutral and representative way poses challenges, even for experts in the field. This makes it difficult to answer questions such as "which operations are viable to automate?", and "how does the accuracy of algorithms compare to manual work?".

This article proposes an approach for generating gold standards for machine classification of patents, and presents one such example. It then describes a methodology to test against that gold standard, and presents the results of evaluation of a commercially available system against it. The gold standard is intended to be a representative reflection of a patented technology, such that it includes a number of positive labelled patents that cover the technology, and a number of negative labelled patents that do not, but are close enough in content that they would be challenging for an algorithm to identify. The technologies selected should be representative of real classification challenges faced by practitioners.

* Corresponding author.
E-mail address: steve.harris@cipher.ai (S. Harris).
[1] info@cipher.ai.
[2] info@patinformatics.com.

https://doi.org/10.1016/j.wpi.2020.101961
Received 1 July 2019; Received in revised form 9 March 2020; Accepted 12 March 2020

Table 1
Most frequent class codes in the $T_{\oplus}$ set (positive labelled training set), and their coverage - $\frac{|T_{\oplus} \cap C|}{|T_{\oplus}|}$, for class code C.

Technology	Class code #1		Class code #2		Class code #3	
Hybrid transmission	B60W 10/08	14.2%	B60K 6/445	13.9%	B60K 6/365	13.4%
Overhead cameras	B60R 1/00	62.2%	H04N 7/18	31.5%	B60R 11/04	15.9%
Pre-chamber combustion	F02B 19/12	35.2%	F02B 19/18	14.6%	F02P 13/00	14.0%
Selective catalytic reduction	F01N 3/2066	51.4%	F01N 2610/02	38.9%	F01N 3/208	20.2%
Tailgate actuation	E05Y 2900/546	38.0%	E05Y 2900/531	14.0%	E05F 15/622	12.4%
Mean		40.2%		22.6%		15.2%

The term "gold standard" is somewhat ambiguous, due to its use in many fields and contexts, but Aroyo and Welty [3] provide a description which covers many cases: "Gold standards exist in order to train, test, and evaluate algorithms that do empirical analysis. Humans perform the same analysis on small amounts of example data to provide annotations that establish the truth. This truth specifies for each example what the correct output of the analysis should be. Machines can learn (in the machine-learning sense) from these examples, or human programmers can develop algorithms by looking at them, and the correctness of their performance can be measured on annotated examples that were not seen during training".

In the following text, we use the binary classification convention of denoting the data labelled as positive (examples of in-scope patents) with $X_{\oplus}$, and those labelled as negative (counter-examples) with $X_{\ominus}$, where X is the labelled set.

We will also describe the processes in set notation for brevity, though restrict the use to just $\cup$ (union), $\cap$ (intersection), $\setminus$ (set difference), and $|X|$ (set cardinality).

2. Prior work

2.1. Existing gold standards

There exist a number of gold standards and more general test datasets for evaluation of machine classification, such as those published in the OpenML[3] online database of labelled machine learning test data. These datasets cover a wide range of topics, but are largely numeric in content, and do not include rich patent data labelled with the technologies which they cover.

There also exists a series of gold standard datasets in the patent domain, the IPC classifications from CLEF-IP,[4] however they are optimised for evaluation of other types of algorithm, chiefly the detection of prior art.

2.2. Using class codes for evaluation

There have been attempts to use the examiner class code information in CLEF-IP, or wider patent datasets to evaluate classification algorithms [4,5], and while the class code labels are plentiful and widely available there exists a question over their suitability for evaluation of this class of problem – the mapping of patents to industry-relevant technologies. Clearly class codes are a suitable gold standard for the automation of the process of patent examiners assigning class codes to applications, however the requirements of practitioners in the industry – mapping their assets against technologies that are relevant to the business – may differ.

In order for class codes to be representative of such real-world problems they should resemble the scope and coverage of technology definitions in use by practitioners in the field. To evaluate this we consider some classes from the Cipher Automotive taxonomy of technologies. This was co-developed with a number of well-known companies in the automotive industry, and is widely used for patent classification in that domain, so can be said to be a reasonable reflection of common practice. Five technologies were selected at random from the Cipher Automotive[5] taxonomy, and their relationship to (CPC[6]) class codes is observed. In the following text T denotes the training set manually constructed in order to train a classifier to the given technology topic.

Table 2
The number of distinct class codes appearing in each set, and the intersection of the class codes of the two sets. The percentage indicates the proportion of class codes appearing in $T_{\oplus}$ (positive labelled training set) that also appear in $T_{\ominus}$ (negative labelled training set)

Technology	Unique class codes in			
	$T_{\oplus}$	$T_{\ominus}$	$T_{\oplus}$ and $T_{\ominus}$	
Hybrid transmission	956	2,265	245	25.6%
Overhead cameras	431	2,339	302	70.1%
Pre-chamber combustion	464	1,101	185	39.9%
Selective catalytic reduction	664	1,106	249	37.5%
Tailgate actuation	548	1,239	265	48.4%
Mean	613	1,389	249	44.3%

As can be seen in Table 1 there is no single class code that spans every patent in $T_{\oplus}$ in any of these cases. The maximum being only 62.6%, and the mean being 40.2%. From Table 2 we can see that a minimum of 25.6%, and mean of 44.3% of the class codes in $T_{\oplus}$ also appear in $T_{\ominus}$.

Taken together this indicates that class codes do not discriminate between technology domains at the level which is expected by practitioners — the class codes are both too narrow in scope, such that it requires many hundred codes to circumscribe an industry-relevant technology; and too broad in that many of them span both the $T_{\oplus}$ and $T_{\ominus}$ sets.

It should be noted that the relationship to the older IPC class codes was not evaluated, as IPC is essentially a subset of CPC. From the World Intellectual Property Organization (WIPO) FAQs "CPC is the Cooperative Patent Classification scheme used by the European Patent Office (EPO) and the United States Patent and Trademark Office (USPTO), which was jointly developed by the two Offices based in a large part on the existing European Classification System (ECLA) and on the USPC, respectively. It is based on the IPC, but it is much more detailed."[7]

2.3. Cross validation

Outside of gold standards, another popular technique for measuring classification accuracy is cross-validation [6]. Cross-validation has the benefit that it requires no additional manual effort to evaluate the accuracy, and it is a useful technique for evaluating classification accuracy in the absence of external data. However, cross validation suffers from the problem that the scope of the evaluation is bounded by the training set, which is not guaranteed to reflect the domain as a whole [7].

[3] https://www.openml.org/.
[4] http://ifs.tuwien.ac.at/~clef-ip/.
[5] http://cipher.ai/automotive.
[6] https://www.cooperativepatentclassification.org/.
[7] https://www.wipo.int/classifications/ipc/en/faq/.

Even in the cases where it can be determined that the training set is truly representative of the domain, there are well-known issues with the inherent inaccuracies of the various cross-validation methods [8]. While these can be compensated for to a degree, they can be avoided altogether with an independently created gold standard, against which robust information retrieval characteristics can be calculated.

Equally it would not be reasonable to take an existing training set from an academic or commercial system to use as a gold standard. There will be some inherent bias towards the system under which it was constructed, due the choices made by the operator to correct identified errors during the training and evaluation cycle, unrepresentatively penalising other systems to which it may be compared.

3. Desirable characteristics of a patent classification gold standard

A number of challenges are faced in the construction of a gold standard for use in the evaluation of classification algorithms, including those described by Aroyo and Welty [3], and those specific to the patent domain.

To address these, the following criteria are proposed for a robust gold standard in this domain:

Scope Defining a scope which is both clear enough to offer a reasonable level of agreement between subject matter experts, and also reflective of real-world use cases. An embodiment of this in the patent domain could be a scope which clearly delineates patents which cover a particular feature, which is relevant to licensing activity.

Agreement Ideally the gold standard covering each topic would be reviewed by multiple subject matter experts — allowing testing against the consensus, most generous, and most narrow definitions. This requires a definition which is clear enough to allow subject matter experts to independently reach the same conclusion as to membership.

Diversity of technology Different patented technology areas have quite differing characteristics in terms of variety of terminology, density of class codes, and quantity of patents, so it is reasonable to assume that different systems will perform with differing degrees of accuracy against each. A good implementation of this would be multiple gold standards covering different technological areas, such as mechanical engineering, software, business methods, semiconductors, and so on.

Size of dataset There is a tension between selecting technologies that are precise enough to be representative of real requirements, yet large enough that multiple experiments can be run without substantial overlap, and withholding enough data for the evaluation to be robust and representative.

Challenging Classifying against the gold standard should be sufficiently difficult that existing solutions cannot easily achieve 100% accuracy, which would render any comparison impossible. If, for example a simple classification technique such as naive bayes could achieve 100% accurate results then the test is not sufficient to discriminate high from low performing solutions.

Independent The gold standard should be created without reference to any existing system, independently, and as far as possible through manual research, to avoid systematic bias — such as the preponderance of a small number of class codes. If during the construction of the gold standard data, the person constructing the set relies upon existing known metadata, then the data set will be compromised by including an artificially easy to discover feature.

Identification One of the more trivial though persistent problems in patent data is the lack of standardisation of patent serial number formatting. The gold standard should use whatever format is the most widely understood. For example, the string "US10012345" may denote the US patent "US10012345B2" (Method and apparatus for an icemaker adapter, 2001), or the US application "US10/012,345" (Multi-mode print data processing, 2015).

4. The process of creating the quantum computing gold standard

The quantum computing gold standard was created by Anthony Trippe of Patinformatics.[8]

The summary is "Qubit Generation for Quantum Computing", and the scope (i.e. the natural language definition of what technologies were included, and what were excluded) was given as:

"Qubit Generation for Quantum Computing refers to patents that discuss the various means of generating qubits for use in a quantum mechanics based computing system. Types of qubits included superconducting loops, topological, quantum dot based and ion-trap methods as well as others. The excluded technologies are applications, algorithms and other auxiliary aspects of quantum computing that do not mention a hardware component, and hardware for other quantum phenomena outside of qubit generation".

This scope was selected by Patinformatics as representative of a real-world problem, and because they have significant experience of analysing patents in the area [9].[9] Because of this background knowledge, the belief was that there would be a sufficient quantity of patent families to allow the construction of a large enough gold standard. It was also felt that the difficulty of identifying the patents in-scope (asin, those which cover technologies such that they fall into the class of positives) would be challenging for machine processing, based on their experiences of manually classifying the technology.

The source data was a mixture of existing known data, and manually directed searching with manual review. During the process of creating the gold standard data Patinformatics did not have access to Cipher, as this could have skewed the selection process in favour of machine processing over human review.

Patinformatics were compensated for their time by Aistemos to allow open publication of the resulting data, however there is no other relationship between the organisations.

The version of Cipher evaluated in this text predates the creation of this gold standard, so there is no optimisation (for example of the text embeddings used to generate the intermediate vectors) specific to this data in the results.

Instructions for obtaining the data produced can be found in Section 10.

5. Analysis of the gold standard data

The cardinality of the example gold standard created for this study ($|G|$) is 1429 EPO simple patent families, broken down as $|G_{\oplus}| = 435$, and $|G_{\ominus}| = 994$, these consist of 2282, and 2801 publications, respectively.

In order to understand the contents of the gold standard further, we can compare the members to the data presented in Section 2.2. Ideally there should be a degree of similarity in the makeup, coverage, and intersection of the class codes that is inline with real-world classifier training sets. Some differences are expected, as the gold standard is by definition a superset of the data required for accurate training, and the quantum computing domain is likely to be different in terms of class

[8] Patinformatics LLC, 565 Metro Place S. Suite 3033, Dublin, OH 43017, USA, https://patinformatics.com/.

[9] https://patinformatics.com/quantum-computing-report/.

Table 3
Most common class codes appearing the gold standard, and the proportion of families in each set that include the class code.

Class code	$G_{\oplus}$ coverage	$G_{\ominus}$ coverage
B82Y 10/00	63.7%	4.2%
G06N 99/002	34.0%	4.8%
H01L 39/223	10.8%	0.5%
H04B 10/70	3.0%	5.3%
H04L 9/0852	1.4%	12.9%
H04L 9/0858	1.1%	4.2%
G04F 5/14	0.2%	7.0%
H03L 7/26	0.0%	7.1%
H04L 9/08	0.0%	6.1%
G04F 5/145	0.0%	5.2%

Table 4
The confusion matrix for binary classification.

Actual class	Recognised as	
	Positive	Negative
Positive	tp	fn
Negative	fp	tn

code coverage from automotive technologies, though the characteristics should be broadly similar.

Table 3 shows the coverage of the most common class codes in the gold standard data. This is broadly in line with the Overhead cameras technology from the automotive data, though somewhat above the mean at 63.7%, 34.0%, 10.8% for G against 62.2%, 31.5%, 15.9% for Overhead cameras.

For the gold standard the number of unique class codes in $G_{\oplus}$ is 596, in $G_{\ominus}$ is 1,403, and the number appearing in both is 130. The intersection is lower than any examples from the automotive domain, being almost half of the mean.

Without a more comprehensive study of the distribution of class codes across different training sets in different technologies it is not possible to be confident if this is reflective of the technology area, or indicative of some minor bias in the construction of the data. The results are similar enough to the automotive technologies to not cause concern about intrinsic problems in the construction of the data — for example class codes that are uncommonly discriminative.

6. Quantifying classifier performance

Demšar [10] identifies the most frequently used information retrieval metrics used in the analysis of supervised learning classifier performance in the literature as *precision*, *recall*, F_1, and *accuracy*, with AUC also being used, though less often.

These metrics are defined in terms of the binary classifier confusion matrix, shown in Table 4, where tp are true positives (correctly identified positives), tn are true negatives (correctly identified negatives), fp are false positives (Type I errors), and fn false negatives (Type II errors).

The metrics are all presented as numbers in the range 0 to 1, and are defined as follows:

$$precision = \frac{tp}{tp + fp} \tag{1}$$

$$recall = \frac{tp}{tp + fn} \tag{2}$$

$$F_1 = 2\,\frac{precision \cdot recall}{precision + recall} \tag{3}$$

$$accuracy = \frac{tp + tn}{tp + fp + fn + tn} \tag{4}$$

In plain language these can be thought of in the following terms:

precision of the answers returned, the proportion that are correct.

Table 5
Results for randomly generated training sets, with $|T| = 300$, evaluated against $G \setminus T$.

Run	tp	tn	fp	fn	*prec.*	*recall*
1	261.0	765.0	79.0	24.0	0.768	0.916
2	264.0	777.0	67.0	21.0	0.798	0.926
3	248.0	782.0	62.0	37.0	0.800	0.870
4	253.0	779.0	65.0	32.0	0.796	0.888
5	257.0	767.0	77.0	28.0	0.769	0.902
6	259.0	777.0	67.0	26.0	0.794	0.909
7	253.0	783.0	61.0	32.0	0.806	0.888
8	257.0	777.0	67.0	28.0	0.793	0.902
9	259.0	770.0	74.0	26.0	0.778	0.909
10	260.0	774.0	70.0	25.0	0.788	0.912
Mean	257.1	775.1	68.9	27.9	0.789	0.902

recall the proportion of the correct answers in the domain that are found.

F_1 the harmonic mean of *precision* and *recall*, provides a simple way to combine them into a single value, such that poor performance in either metric is visible in the F_1 score.

accuracy of all the answers given, what proportion are correct.

The *accuracy* score can be misleading in the presence of unbalanced classes [11] (e.g. more negatives than positives), which is generally the case in patent classification, however it has been included here for consistency with other work.

7. Naïve training set construction

For less challenging classification tasks accurate results can be obtained by constructing a training set from a random subset of $T_{\oplus}$ and $T_{\ominus}$, and evaluating the classifiers built from these training sets against $G \setminus T$. However, from practical experience in the field it has been discovered that this technique is not effective for patent classification against commercially relevant topics.

Because of this it is expected that a similarly constructed training set drawn randomly from G will produce relatively low *precision* and *recall* numbers, and this can be used as a test of the appropriate difficulty of the classification challenge.

Anecdotally, *precision* and *recall* scores (and hence F_1) in excess 0.9 have been reported by end-users as being approximately equivalent to results produced by manual search and review of patents by skilled practitioners. Based on this we would expect to see a random construction of training sets produce scores under this threshold for a robust gold standard.

A series of 10 training sets were randomly constructed, such that $|T_{\oplus}| = |T_{\ominus}| = 150$, thus $|T| = 300$. Classifiers as described in Section 9.1 were then trained against the random training sets. The confusion matrices from evaluation against $G \setminus T$ were then calculated, and the results can be seen in Table 5.

The value of 300 was chosen as a typical training set size, from practical experience of patent technology classification, with manually curated training sets.

From this we can see that, at least for this implementation, randomly generated training sets do not produce a level of *precision* that meets user expectations. This gives a degree of confidence that the task presented by classifying the gold standard is not so trivial as to be unrepresentative of real-world technology classifications.

It is interesting that the mean *recall* is just in excess of the 0.9 threshold, though the micro-average [12] mean F_1 score for this test is 0.842, suggesting that the overall perceived accuracy would be insufficient.

The disparity between *precision* and *recall* scores for random training sets is substantial. In order to illustrate possible causes, a UMAP dimensional reduction [13] was performed on the entire gold standard

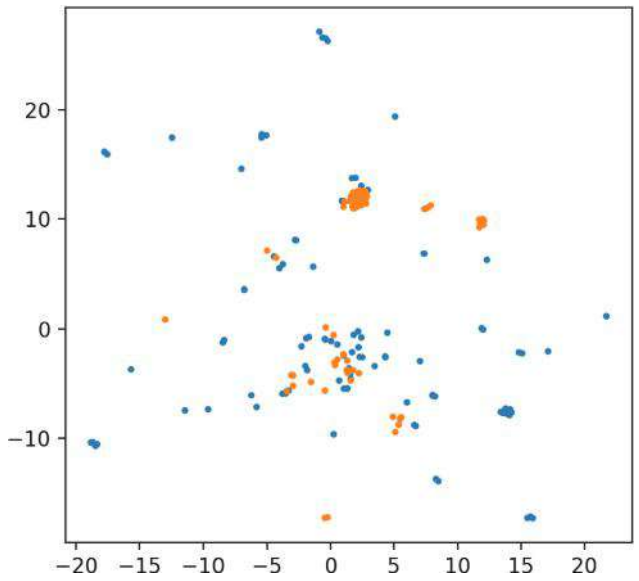

Fig. 1. UMAP dimensional reduction of 100 randomly selected positives (orange), and 100 randomly selected negatives (blue). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

data, using a pre-existing deep learning embedding[10] of CPC class codes. 100 positive, and 100 negative families were then selected at random, and plotted in Fig. 1. The parameters used were $n\_neighbours = 5$, $min\_dist = 0.5$, and the euclidean metric was used for distance calculations.

This reduction gives some indication of why this may occur. The positive points are mostly densely clustered in a small number of locations in the space, whereas the negatives are more scattered. If this is a meaningful representation of the information space, then it would be hard for the classifier to define the boundaries of the space denoting positive class, and will tend to be over-inclusive of positive results. This would be a cause of the high *recall*, but low *precision*.

With further work, it could be established if the process of following the algorithm described in Section 8 also identifies negative points that help to define the boundary in such a space.

8. Modelling real-world directed training

As we have shown, random construction of training sets does not yield results that are either representative of practical experience, or sufficiently accurate to be useful. Consequently it is necessary to define a representative, repeatable, and fully algorithmic way to model operator-directed construction of training sets, so that classification accuracy can be evaluated in a robust manner.

The domain of interest can be characterised as per Fig. 2, where U is the entire domain (all patents relevant to the subject of the gold standard, whether positive or not), P is the patents that are positive to the class, K is the patents that are currently known to the operator, or will become known during the training process, and T is the current training set. By definition the training set must be a (non strict) subset of the known patents.

From this we can define the sets:

$$N = U \setminus P \qquad \text{all negatives} \qquad (5)$$

$$E = K \setminus T \qquad \text{evaluation set} \qquad (6)$$

[10] A description of the embedding is beyond the scope of this article, but as it is simply used to illustrate the relationship between labels it is of little consequence.

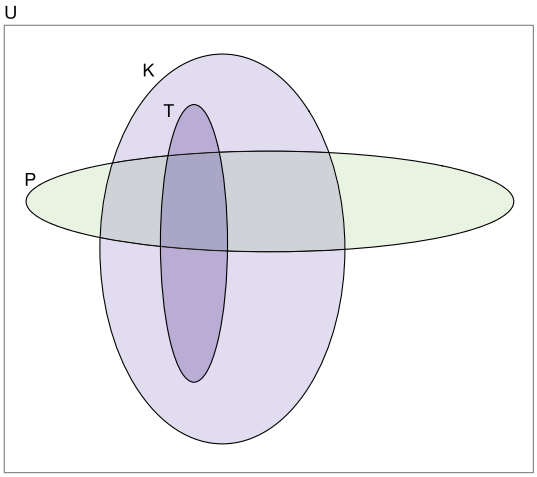

Fig. 2. Euler diagram showing sets of interest to the training process.

$$T_{\oplus} = T \cap P \qquad \text{training set positives} \qquad (7)$$

$$T_{\ominus} = T \cap N = T \setminus P \qquad \text{training set negatives} \qquad (8)$$

The process of training a supervised learning classifier can be characterised as moving patents from E into T in such a way as to increase the extent to which $T_{\oplus}$ and $T_{\ominus}$ represent the characteristic differences between P and N, enabling the classifier to "learn" what those differences are.

Hence, during a typical training process the operator follows the following cycle:

1. Identify a small number of members of P to form the initial $T_{\oplus}$ – provided by an end-user, discovered by manual search, or some combination
2. Identify a small number more-or-less arbitrary members of N to form the initial $T_{\ominus}$
3. Train the classifier on T
4. Apply the classifier to some subset of K
5. Correct the most obvious errors, adding to T
6. Repeat from 3., until the classifier evaluates to some success criteria, such as $F_1 \geq 0.9$

In real-life situations the operator is a human expert, who is responsible for training the classifier, in the algorithm described in this article it is a software simulation of that operator, modelled as the selection of new members of the training set based on highest log-loss.

The way this was modelled in software was to start with randomly selected initial K, and H sets, such that $K \cap H = \emptyset$ and $T \cup H = G$, where the gold standard is G, and the holdout set is H (representing $U \setminus K$ in the user-driven case). T, the training set starts out with a balanced random subset of K, and progressively acquires members from $K \setminus T$ to reduce errors observed in K. The pseudocode is given in algorithm 1.

In this way the data is divided into three portions - a randomly selection withheld set (H) that is purely used for testing, an initial training set (T), and the remainder which is used to augment the training set (E).

8.1. Selection of constants

The constants $\alpha, \beta, \gamma, \delta$ were given the values 100, 350, 286 and 5 respectively for this study. The choice of α, γ, δ are chosen largely for reasons of computational efficiency, and to match the scale of the quantum computing gold standard, as described below.

Reducing α, the initial cardinality of T, causes the evaluation scores to be slightly higher in earlier iterations, and increases computational

Algorithm 1 Training process

function SAMPLE(S, n)
  return ROUND(n) elements from SHUFFLE(LIST(S))
end function

function HIGHLOSS(S, n)
  $l \leftarrow$ LIST(S), sorted by LOGLOSS, descending
  return first n elements of l
end function

$H \leftarrow$ SAMPLE($G_\oplus, \frac{|G_\oplus|}{|G|}\gamma$) $\cup$ SAMPLE($G_\ominus, \frac{|G_\ominus|}{|G|}\gamma$)
  ▷ hold out proportion, stratified sample
$T \leftarrow$ SAMPLE($G_\oplus \setminus H, \frac{\alpha}{2}$) $\cup$ SAMPLE($G_\ominus \setminus H, \frac{\alpha}{2}$)
  ▷ initial training set, balanced sample
while $|T| \leq \beta$ **do**
  $f \leftarrow$ TRAIN(T) ▷ train classifier
  $E \leftarrow G \setminus T$ ▷ define evaluation set
  $r \leftarrow f[E]$ ▷ apply classifier to evaluation set
  WRITE(r) ▷ log confusion matrix
  if PRECISION(r) $\geq$ RECALL(r) **then**
    $T \leftarrow T \cup$ HIGHLOSS($E_\oplus \setminus H, \delta$) ▷ add δ positives
  else
    $T \leftarrow T \cup$ HIGHLOSS($E_\ominus \setminus H, \delta$) ▷ add δ negatives
  end if
end while

effort, though from some experimentation this effect on evaluation scores did not appear to be substantial.

Increasing β, which governs the maximum cardinality of T, and hence the number of iterations, substantially increases the evaluation time, and can have some negative impact on the selection of patents in $T_\oplus$, as discussed in Section 8.2.

γ, the cardinality of the held-out set was chosen as 0.2 $|G|$ – 20% of the data, a typical proportion for this kind of evaluation.

Increasing δ, the number of families added to T in each iteration, reduces the computational effort of evaluation, at the cost of decreasing the resolution of the analysis of the rate of change of the metrics with respect to training set size seen in Section 9. A value of 10 or 20 would be more representative of human-directed training, though some experiments revealed that it does not materially affect the evaluation results.

The conditional on precision and recall reflects the user choosing to correct for false positives or false negatives, depending on which are more apparent, and the function HIGHLOSS(S, n) reflects a user identifying the most obvious errors, and correcting them. From analysing user behavioural statistics of the Cipher system (see Section 9.1), it has been observed that users tend to briefly focus on identifying runs of positives, then runs of negatives, so this has been reflected in this process.

8.2. Limits of β

As $|T_\oplus|$ approaches $|G_\oplus \setminus H|$ the training set becomes equivalent to a randomly generated one, due to the reduction in selection freedom of the training model. Consequently, at some point the metrics for the classifier will decrease with increasing training set size.

As we can see in Fig. 3, when $|T| = 400$, the available set of positives ($G_\oplus \setminus H$) that has been incorporated into $T_\oplus$ is around 58%. At $|T| = 500$ the consumed proportion is 72%, significantly constraining the selection that can be made from $T_\oplus$. This is due to the size of the held-out set, and the tendency of the training set to be balanced between $\oplus$ and $\ominus$, while the gold standard as a whole is not.

This effect could be attenuated by reducing $|H|$, however this methodology would then be a worse representation of the real-world situation, as the unknown portion of data is typically substantial in comparison to the size of the training set, and would reduce the accuracy of the evaluation.

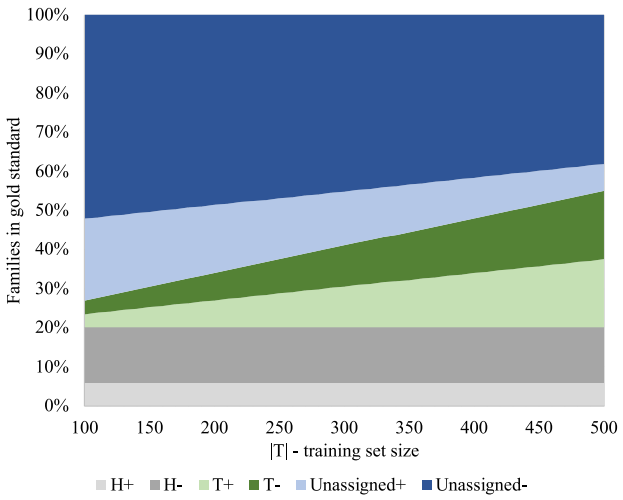

Fig. 3. Chart showing usage of families in gold standard as training set size increases, "Unassigned" is $G \setminus (H \cup T)$.

Consequently we limit β to 350 for evaluation of this gold standard, to minimise the impact of this effect. Further work would be required to determine the exact significance of this factor. By observation, with $|H| = 0.2\ |G|$, $\beta = 350$ the evaluation results appear to be unaffected for this gold standard – *precision* and *recall* at $|T| = 350$ are greater than or equal to those at $|T| < 350$, see Table 6. For future gold standards other choices of β may be required, to reflect differing cardinalities of $G_\oplus$ and $G_\ominus$.

It is not clear how much the reduction in the rate of change of *precision* and *recall* with respect to $|T|$ is due to diminishing returns from the increased data, and how much to the lack of choice in candidate patents to add to $T_\oplus$. Further work would be required to determine this.

8.3. Alternative algorithms

An obvious alternative would be to only evaluate against the H set, which gives the advantage of a constant evaluation set size. In order to reduce the variance of the scores it was found that large values of γ were required, limiting values of β, as described in Section 8.2 to levels where limits of values for *precision* and *recall* could be obtained. For this dataset, and this implementation the combinations of β and γ that were evaluated also resulted in a high variance of the F_1 score.

A much larger gold standard would reduce this effect, though would introduce other issues. Commercially relevant technologies to be classified tend to be relatively specific, and it is rare for examples with tens of thousands of candidates for $T_\oplus$ to exist. Any such training set is likely to be idiosyncratic, and not reflective of many real-world tasks.

Additionally, this variant approach would not reflect what users experience when evaluating classifier results — the results from $U \setminus T$ seen in the final output are a mixture of results that are not known to the operator (H), and ones that have been observed, though deemed to be correct (E).

8.4. Comparison to randomly selected training sets

Fig. 4 shows the difference in *precision* and *recall* for classifiers built from randomly selected training sets (as in Section 7) and the directed training algorithm described here, for the same classification engine.

The data in Fig. 4 is drawn from Table 5, and the source data for Table 6.

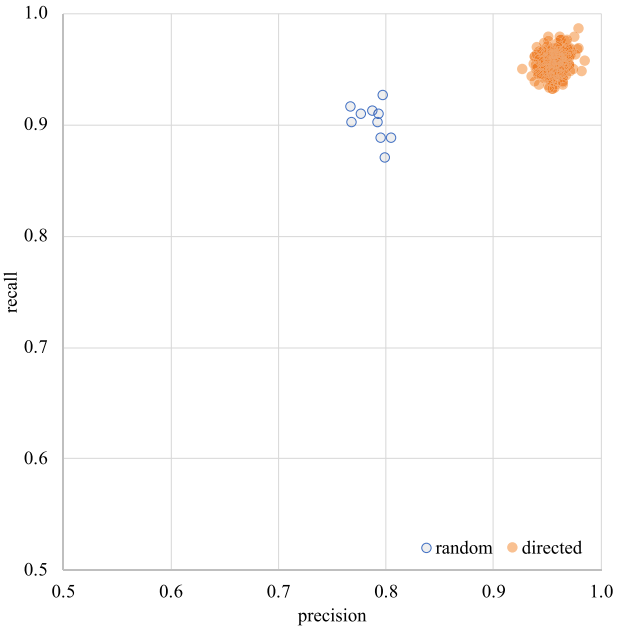

Fig. 4. Scatter plot of *precision* and *recall*, for randomly selected training sets and directed training sets, with $|T| = 300$.

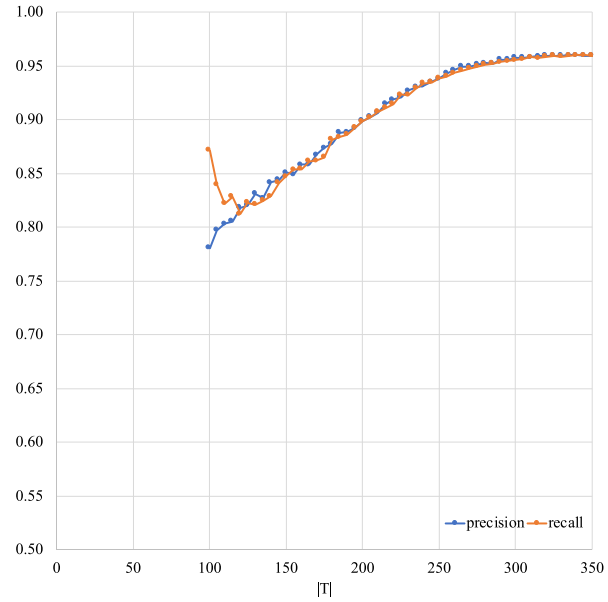

Fig. 5. Means of *precision* and *recall* with training set size, calculated over 200 runs from random starting point, and with random held-out data.

As can be clearly seen, the directed training produces substantially higher *precision* and higher *recall* for the same training set size.

The difference in mean *recall* between the methodologies of 6.1% (0.956 and 0.901) may not seem substantial, however this reflects a 44.9% reduction in the False Negative Rate, from 0.098 to 0.044.

9. Evaluating Cipher's classification AI

9.1. About Cipher

In the results that follow the TRAIN() function in the pseudo-code is provided by the July 2019 version of the Cipher classification algorithm, as described in this section.

Cipher[11] is a commercially available strategic patent information system, the key feature of which is the ability to use trained AI classifiers to tag patent assets against defined technologies.

The classification system is based on an ensemble of learners, each trained on embeddings generated from patent data.

Textual and medadata embeddings for model training are obtained through a combination of domain specific normalisation and transformations, and a separately trained patent-specific language model.

Model parameters were typically obtained either through a random or directed hyperparameter searches.

As the hyperparameters are determined algorithmically it is necessary to perform a large number of iterations, to ensure representative results.

9.2. Methodology and results

In order to obtain representative results, we executed algorithm 1, a total of 200 times. This required 183 CPU-hours, when executed on 4.3 GHz Intel i7-7740X CPUs with NVIDIA GeForce GTX 1080 Ti GPU accelerators.

The results of executing algorithm 1 are shown in Table 6, abbreviated to only show the results of every 5 iterations (5δ increments to $|T|$).

[11] http://cipher.ai/.

Table 6
Result of testing Cipher against the quantum computing gold standard — mean over 200 runs, for every 5 iterations, with $\alpha = 100$, $\beta = 350$, $\gamma = 286$, $\delta = 5$.

Iter.	$\|T\|$	tp	tn	fp	fn	$prec.$	$recall$	F_1	$acc.$
0	100	335.5	850.0	94.0	49.5	0.781	0.871	0.824	0.892
5	125	309.8	859.8	67.8	66.6	0.821	0.823	0.822	0.897
10	150	308.2	861.5	53.8	55.4	0.851	0.848	0.849	0.915
15	175	303.6	859.2	43.9	47.3	0.874	0.865	0.869	0.927
20	200	303.8	857.1	34.0	34.1	0.899	0.899	0.899	0.945
25	225	300.1	853.4	25.6	24.8	0.921	0.924	0.922	0.958
30	250	293.7	846.8	19.3	19.2	0.938	0.939	0.939	0.967
35	275	285.2	839.2	14.6	15.1	0.951	0.950	0.951	0.974
40	300	275.6	828.5	12.1	12.8	0.958	0.956	0.957	0.978
45	325	265.3	816.5	11.1	11.1	0.960	0.960	0.960	0.980
50	350	255.0	802.7	10.7	10.7	0.960	0.960	0.960	0.980

Note that the results in Table 6 and Figs. 5–7 are computed as the micro-average of the confusion matrices from multiple runs, rather than the macro-average [12]. This allows direct comparison of $\overline{F_1}$ and $\mathrm{Var}(F_1)$.

Figs. 5 and 6 show the evolution of *precision* and *recall*, and F_1 and *accuracy* as the training set grows.

9.3. Observations

We can see that the initial *recall* is much higher than *precision* – as expected from a random starting point, as observed in Section 7. From this point *precision* grows steadily, while *recall* reduces quickly with respect to training set size, then increases as T develops towards a more optimal selection. Eventually *precision* and *recall* converge around similar values, due to the balancing action of the training simulation algorithm. This reduction in *recall* causes a corresponding reduction in the F_1 score in the range of $100 < |T| < 150$.

As we can see, the ultimate *precision*, *recall*, and F_1 of the Cipher implementation are around 0.96, and these values are reached for values of $|T| = 300$, or $|T_{\oplus}| \approx |T_{\ominus}| \approx 150$. This reflects anecdotal reports from users who have trained the Cipher system against real-world topics (such as the automotive taxonomy), and provides reassurance that both the gold standard data, and the training simulation algorithm are a reasonable reflection of real-world situations.

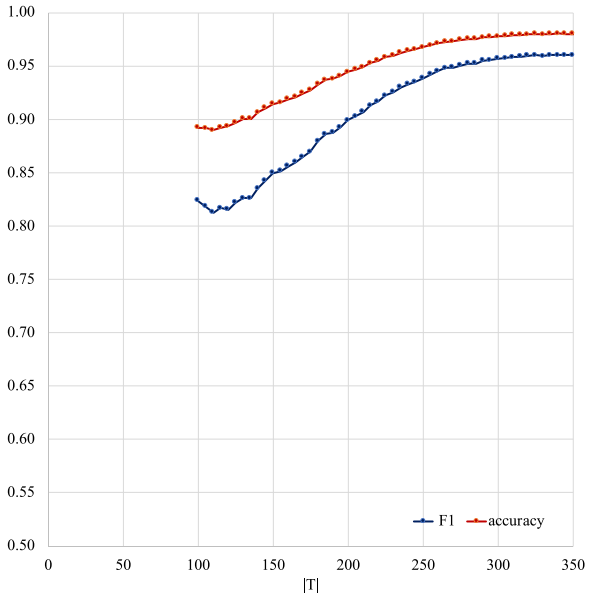

Fig. 6. Means of F_1 and *accuracy* with training set size, calculated over 200 runs from random starting point, and with random held-out data.

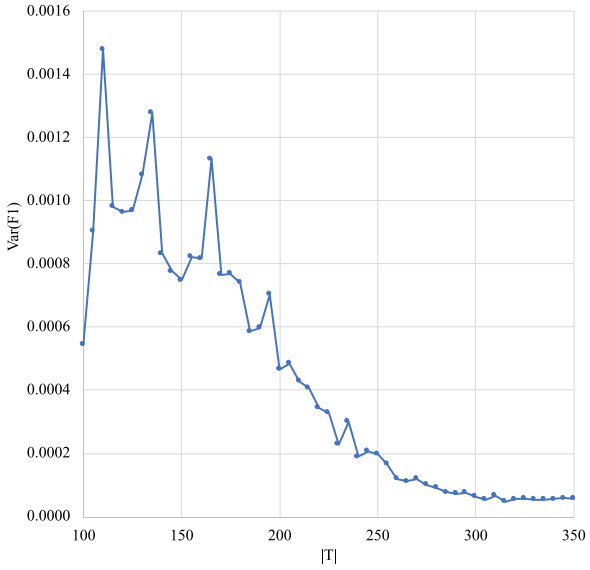

Fig. 7. Variance of F_1 with training set size, calculated over 200 runs from random starting point, and with random held-out data.

The ultimate *accuracy* is 0.98, though as discussed previously this is not a robust measure for unbalanced sets.

One important aspect of classifier performance that is often under reported is the variance of results over repeat runs with different datasets [10]. The variance of the F_1 score can be seen in Fig. 7, and the distribution of F_1 is rendered as a scatter plot in Fig. 8. As can be seen from the plots, the variance is initially high (when the training set is close to random), and converges on stable values as the training set is increasingly optimally selected from the pool of available data.

10. Conclusion

Though this is early work on the analysis of this class of classification problem, the quantum computing gold standard appears to be representative of real-world experience of classification of patents.

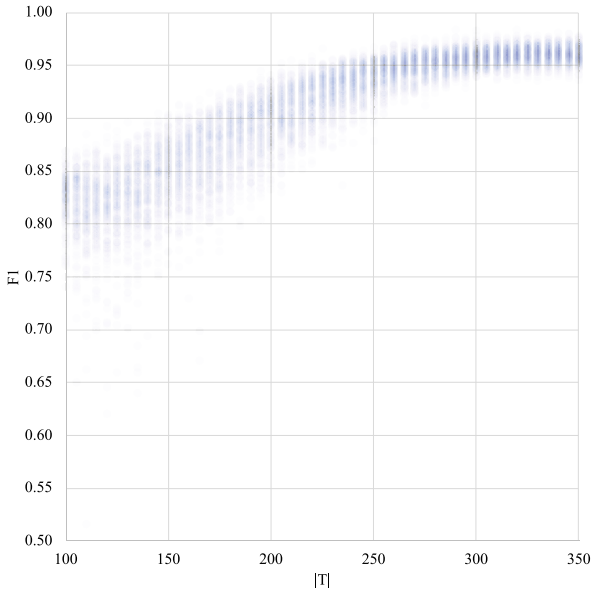

Fig. 8. Scatter plot of F_1 against training set size, calculated over 200 runs from random starting point, and with random held-out data.

The required training set sizes, and eventual accuracy of classification match anecdotal evidence from real users, and statistically the data resembles known-good training sets.

In so far as possible to this point we have addressed the criteria presented in Section 3, though some remain for future work:

Scope Appears to be an accurate reflection of commercial practise, though more evaluation would be beneficial.

Agreement The published data represents a single view, future work includes obtaining blind reviews of the data by other practitioners. This could also be used to establish consensus, maximal, and minimal classification targets.

Diversity of technology Currently only one technology is covered. Future work includes adding further gold standards from different industries.

Size of dataset The size of the dataset appears to be sufficient for robust evaluation, whilst also being a representative technology.

Challenging Analysis of the behaviour of random training sets indicates that the difficulty of analysis of this technology is comparable with real-world classification tasks.

Independent The gold standard was created in the absence of an algorithmic classification system, by a neutral third party.

Identification Data is published in the widely-used European Patent Office format, with titles and dates available to aid cross-checking.

The data for the quantum computing gold standard can be found at https://github.com/swh/classification-gold-standard/tree/master/data. It is made available under the BSD 3-Clause License, to allow reuse in other projects in a variety of ways. The site includes documentation for the file and data format the gold standard is represented in.

In the future, additional gold standard datasets will be published at this location, to allow a more comprehensive analysis of the behaviour of various patent classification algorithms, across multiple domains, and created by different authors.

Practitioners wishing to independently construct their own gold standards following this method can apply the analysis in Section 2.2 to determine if they are sufficiently independent of class codes, and apply algorithm 1 with a series of classifiers in order to determine the spread of metrics.

Future work includes the creation and analysis of further gold standards, and the comparison of multiple classification algorithms to determine the distribution of results.

CRediT authorship contribution statement

Steve Harris: Conceptualization, Methodology, Formal analysis, Writing - original draft. **Anthony Trippe:** Conceptualization, Investigation, Data curation, Writing - review & editing. **David Challis:** Methodology, Software, Writing - review & editing. **Nigel Swycher:** Conceptualization, Writing - review & editing, Funding acquisition.

References

[1] T. Cattaneo, Patent Intelligence for Competitive Benchmarking: Brembo Case Study (Master's thesis), Universita Carlo Cattaneo - LIUC, 2013.
[2] H. Hallesius, Patent search quality, 2016, presented at CIP Forum 2016.
[3] L. Aroyo, C. Welty, Truth is a Lie: Crowd truth and the seven myths of human annotation, 36, (1) 2015.
[4] J. Guyot1, K. Benzineb1, G. Falquet, myClass: A Mature Tool for Patent Classification, CLEF-IP, 2010.
[5] D. Molla, D. Seneviratne, Overview of the 2018 ALTA shared task: Classifying patent applications, in: Proceedings of the Australasian Language Technology Association Workshop 2018, ACM, 2018, pp. 84–88.
[6] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Proceedings of the International Joint Conference on Artificial Intelligence IJCAI, Morgan Kaufmann, 1995, pp. 1137–1143.
[7] S. Varma, R. Simon, Bias in error estimation when using cross-validation for model selection, BMC Bioinformatics 7 (91) (2006).
[8] T. Fushiki, Estimation of prediction error by using K-fold cross-validation, Stat. Comput. 21 (2) (2011) 137–146.
[9] A. Trippe, B. Scanlon, G. Mishra, Practical Quantum Computing: A Patent Landscape Report, 2017.
[10] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[11] F. Provost, T. Fawcett, R. Kohavi, The case against accuracy estimation for comparing induction algorithms, in: Proceedings of the Fifteenth International Conference on Machine Learning, ICML-1998, Morgan Kaufmann, 1998, pp. 445–453.
[12] F. Sebastiani, Machine learning in automated text categorization, ACM Comput. Surv. 34 (1) (2002) 1–47.
[13] L. McInnes, J. Healy, J. Melville, UMAP: Uniform manifold approximation and projection for dimension reduction, 2018, arXiv:1802.03426 [stat.ML].

Steve Harris worked as a researcher at the University of Southampton in the UK, from 1996–2006, working in the areas of AI, Machine Learning, Graph databases, and Hypertext. In 2005 he joined a newly founded fraud prevention startup, Garlik, taking the role of CTO until its sale in 2011. In 2013 he co-founded a new startup, Aistemos — applying AI technologies to strategic patent problems. Steve is the author of around 30 peer reviewed Computer Science papers, more information can be found at https://www.linkedin.com/in/swharris/.