BMC Bioinformatics

METHODOLOGY ARTICLE

Open Access

Automatic consistency assurance for literature-based gene ontology annotation

Jiyu Chen[1], Nicholas Geard[1], Justin Zobel[1] and Karin Verspoor[1,2]*

*Correspondence: karin.verspoor@rmit.edu.au
[2] School of Computing Technologies, RMIT University, Melbourne, VIC 3000, Australia
Full list of author information is available at the end of the article

Abstract

Background: Literature-based gene ontology (GO) annotation is a process where expert curators use uniform expressions to describe gene functions reported in research papers, creating computable representations of information about biological systems. Manual assurance of consistency between GO annotations and the associated evidence texts identified by expert curators is reliable but time-consuming, and is infeasible in the context of rapidly growing biological literature. A key challenge is maintaining consistency of existing GO annotations as new studies are published and the GO vocabulary is updated.

Results: In this work, we introduce a formalisation of biological database annotation inconsistencies, identifying four distinct types of inconsistency. We propose a novel and efficient method using state-of-the-art text mining models to automatically distinguish between consistent GO annotation and the different types of inconsistent GO annotation. We evaluate this method using a synthetic dataset generated by directed manipulation of instances in an existing corpus, BC4GO. We provide detailed error analysis for demonstrating that the method achieves high precision on more confident predictions.

Conclusions: Two models built using our method for distinct annotation consistency identification tasks achieved high precision and were robust to updates in the GO vocabulary. Our approach demonstrates clear value for human-in-the-loop curation scenarios.

Keywords: Biological database quality, Gene ontology annotation, Text mining

Background

The gene ontology (GO) is a framework used to uniformly describe gene function and support the representation of biological systems, based on a set of hierarchical controlled vocabularies [1, 2]. GO annotation of genes involves two major components: the *GO information*, which includes GO terms and their definitions or descriptions, and *supportive evidence*, which includes the coding regions on a genomic sequence, or a reference to a document describing experimental findings relating to gene product function. Sequence-based GO annotations are produced by comparing different sequences drawn from coding regions; this process can transfer GO terms from an old sequence to a new sequence because two genomic sequences with structurally similar coding regions tend to produce gene products with similar functions [3, 4]. Literature-based GO annotations (GOA) are produced by reviewing the description of experiments in research papers, selecting appropriate GO terms for the experimental findings, and labelling the annotation with a GO evidence code[1] [5–8] indicating the nature of the evidence. While the majority of sequence-based and literature-based GO annotations are automatically produced, the most reliable are manually annotated by expert curators. There is a pressing need to implement reliable tools for automatic curation of GOA as the volume of biological data is constantly increasing.

There are currently around eight million GOA across 4743 species recorded in the GO Consortium Database.[2] However, fewer than 2% of these are manually curated; these are linked to 162,459 publications [9]. Automatic GO curation is efficient but the existing benchmarks are unreliable [9–11]. Furthermore, annotation tools that target a fixed set of terms cannot satisfy the open-world assumption, which requires that the collection of GO terms be updated with the discovery of new gene functions [11]. For example, the GO categoriser (GOCat) is based on a closed world assumption, which is that all relevant terms have been previously observed. It relies on a K-Nearest Neighbour algorithm to compare the semantic similarity of an existing GO annotated evidence text and a new text describing a gene function [10]. GOCat uses several strategies to select GO terms from the old evidence text and rank by relevancy to annotate the new text. However, if the new text describes a gene function that has large semantic distance from any existing evidence text, GOCat will fail to shortlist GO terms and thus will skip this functional annotation. Also, tools based on the closed-world assumption may be biased towards frequently selected GO terms [11, 12], such as assigning "protein binding (GO:0005515)" to a large proportion of genes [13]. Another tool called ConceptMapper [14] utilises dictionary-based concept recognition to achieve competitive performance in annotating GO concepts on the Colorado Richly Annotated Full Text (CRAFT) corpus [15, 16]. However, this tool cannot recognise GO concepts that do not explicitly occur as phrases within evidence texts. For example, "positive regulation of vesicle fusion (GO:0031340)" cannot be recognised from "Rat SYT1 gave rise to efficient Ca2+-promoted fusion activity".

GO annotation is not a one-time process [11]. After a GO term has been assigned to a gene product and linked to evidence, database curators need to continue to monitor the consistency of this annotation against new findings. For example, if a gene product was previously published as having negatively regulated behaviour but is later reported as being uncertain, then this GO annotation should be removed from the database. Poor-quality records within databases can cause cascading errors [17] that in turn may lead to significant negative impact to many down-stream tasks such as gene expression analysis. However, existing studies largely focus on methods for efficient GO annotation enrichment, with less emphasis on maintaining the quality of annotations that have already been recorded within databases.

[1] http://geneontology.org/docs/guide-go-evidence-codes/.

[2] http://geneontology.org/stats.html.

There are many challenges to assess the quality of GOA [18]. Some researchers have estimated a relatively high error rate of GO-curated sequence annotations [19]. A series of quality issues in literature-based GO annotations have also been identified. Some GOA are reported as being assigned to unsupportive evidence texts [20]. Some curators find difficulties in selecting informative GO terms at the proper level of specificity [9]. These quality issues can be seen as reflecting inconsistencies between GO information and evidence information. No tools have been proposed for automatic evaluation of the correctness of evidence code selection. The current solution is to create comprehensive curation guidelines and manually ensure annotation consistency. However, this is unscaleable [7, 21, 22].

To the best of our knowledge, there is no study focusing on the scale or the characteristics of literature-based GOA grounded at per-annotation level. There is no dataset specifically created for promoting automatic GOA inconsistency detection research. The feasibility of implementing automatic GOA inconsistency detection method is unknown. Therefore, a systematic exploratory study of building automatic tools for assisting real-world GOA inconsistency detection is needed. To address these concerns, we propose a novel method that uses text mining for the maintenance of literature-based GOA consistency. The method can automatically distinguish consistent GOA and four major types of inconsistencies as well as satisfying the GO open-world assumption. At current stage, the simulation experiment based on the proposed framework is applied on GOA instances grounded at evidence level but can be extended to process GOA in real-world format in future.

We model GOA (in)consistency as typed pairwise relationships between GO information/evidence code and associated evidence text. We formalise four primary types of GOA inconsistencies that violate curation guidelines [7, 21] or have been reported by previous researchers (Table 1):

- *Type A: Contradictory* description of gene regulation function
- *Type B: Over-specific* (or over-informative [9]) selection of gene ontology terms
- *Type C: Unsupportive texts* inappropriately selected as evidence [20]
- *Type D: Erroneous selection of experimental type evidence code*

Types A–C involve inconsistencies related to GO term selection while Type D relates to the broader nature of the evidence for the annotation; we model these groups separately. Type B and Type D inconsistencies are misleading to both human and automatic tools based on text mining. This is because the selection of GO term at proper specificity and selection of correct evidence code often requires strong background knowledge. Type A and Type C inconsistencies are misleading to automatic tools. The semantics of context in these inconsistencies are close to a consistent instance. For example, for the sampled Type C instance in Table 1, automated tools may incorrectly link this evidence sentence with the GO term "cytosol" as it explicitly appears as a keyword in the text, even though that term is not directly associated with any gene product in the evidence text. The formalisation of four types of GOA inconsistencies can help curators address detailed exploratory analysis of database consistency issues.

Table 1 Examples of four types of inconsistent Gene Ontology Annotations; 3 term-related and 1 related to evidence codes

Term inconsistency	**Type A—Contradictory description of gene regulation function**
	Evidence: As expected, rat SYT1 gave rise to efficient Ca2+-promoted fusion activity
	GO term: negative regulation of vesicle fusion (GO:0031339)
	Inconsistency: The evidence describes that SYT1 has a positive regulation of vesicle fusion which is contradictory to the negative expression in the selected GO term
	Type B—Over-specific GO term selection
	Evidence: Recent studies from this laboratory have provided strong evidence that endogenous GRK2 and GRK6 can regulate the responsiveness of M1 mACh receptor signalling in cultured rat hippocampal neurons
	GO term: negative regulation of G-protein coupled receptor protein signaling pathway (GO:0045744)
	Inconsistency: The provided evidence only support the regulation of G-protein coupled receptor protein singling pathway while its negativity is unknown. Thus, the annotated GO term is over-specific
	Type C—Unsupportive evidence text
	Evidence: In order to characterise the most prominent protein changes that arise in livers from rats fed control or ethanol-containing diets with or without betaine supplementation, cytosolic liver proteins were resolved by 1D PAGE.
	GO term: cytosol (GO:0005829)
	Inconsistency: The evidence text has the mention of a GO concept "cytosol" but does not express the cellular component information of any gene product thus the evidence text does not correctly support the selected GO term
Code inconsistency	**Type D—Erroneous selection of experiment type GO evidence code**
	Evidence: At 22h after pollination, we found pollen tubes in 37.5% of the ovaries following pollinations by EXPB1 pollen. In contrast, we found no pollen tubes in the ovaries at 22h after pollination when the silks were pollinated by expb1 pollen. Together these data indicate that the expb1 pollen grows more slowly in vivo than the EXPB1 pollen.
	Evidence code: IGI
	GO term: pollen tube growth (GO:0009860)
	Inconsistency: The evidence indicate the annotation is based on allelic variation and the experiment is conducted by comparing the single gene to the alleles of the same gene. Thus, the correct evidence code should be IMP instead of IGI

To evaluate our method, we generate a collection of GOA instances that fall into each (in)consistency category by directed manipulation of instances in the evidence-based BC4GO corpus [23], created using real-world GOA records in model organism databases. We fine-tune two BioPubMedBERT models [24] to distinguish consistent GOA from the three kinds of term inconsistency (*Model-Term*) and from evidence-code inconsistency (*Model-Code*). We propose a simple strategy to extend BioPubMedBERT with additional layer to encode section information marked by the location of evidence text in the article during fine-tuning. The performance of Model-Term and Model-Code are evaluated using *Precision*, *Recall* and F_1 metrics grounded at each evidence text on a test set that is independently generated from 49 full-text articles. Model-Term achieved 0.69 and Model-Code achieved 0.52 *micro-Precision* overall. We optimise training data using a term-overlap similarity measure and improve the ability to distinguish consistent GOA from other types of inconsistencies. We find a significant improvement in precision among each type of (in)consistency when the uncertainty (Shannon's entropy) of predicted outcomes decrease.

To identify the typical linguistic features that are influencing the models' performances, we undertake error analysis based on a linguistic test suite approach [25, 26] and find the length or typical composition structure of GO terms, the occurrence of digits or Roman numerals in the GO term, the length of evidence text versus the length of the GO definition text, and overlaps between a GO term and evidence text may all influence the model's prediction uncertainty and have overall consequences for model performance. Together these outcomes demonstrate the value of our methods as an organising framework, and for improving the efficiency and accuracy of human-in-the-loop GOA curation.

To provide context for our methods, we introduce the Gene Ontology and describe the existing evidence-based corpus that we exploit in our method and experiments. We also introduce different methods for measuring the semantic similarity between naturally written texts within documents, or different GO terms, modelled on a Directed Acyclic Graph (DAG); some of these are used in our methods. Finally, we discuss the design of linguistic test suite and measurement of prediction uncertainty for post-hoc error analysis. There are no prior methods for automatic maintenance of literature-based GOA consistency, to the best of our knowledge, but as we discuss there are several relevant resources.

The GO is a controlled vocabulary developed to uniformly describe the molecular activity of a gene product (*molecular function*) in a specific location of cell (*cellular component*) and how it contributes to a broad biological objective (*biological process*) [22]. The GO has a hierarchical structure and is modelled as a DAG with terms as nodes and relations between the terms as edges.[3] Parent terms are broad while child terms express more specific information; for example, "suckling behavior (GO:0001967)" is a child term of "feeding behavior (GO:0007631)", which indicates a more specific form of food intake via nourishment from the breast. Curators need to make sophisticated inferences in order to select the proper specific level of GO term from the hierarchical graph, balancing the need to specify the gene function as precisely as possible against the risk of exceeding the level supported by the evidence.

The BC4GO corpus was created by eight expert curators from five different model organism databases for the GO annotation task in BioCreative IV [23]. In contrast to a mention-based GO corpus such as CRAFT [15], BC4GO mirrors the real-world GO curation scenario, providing each GO annotation with traceable evidence grounded at sentence level within literature. For example, the GO term "growth (GO:0040007)" commonly appears within articles but not every sentence that mentions "growth" is truly supportive gene function evidence. The CRAFT corpus includes annotations of every appearance of "growth" as a GO concept but these are largely not directly relevant for GO curation. The BC4GO corpus categorises evidence sentences into either experiment type or summary type. The experiment-type sentences describe details of how an experiment was conducted and can be used to produce a complete GO annotation by referring to the GO definition and the decision tree for evidence code selection.[4] The summary-type sentences only describe the results of experiments and are used only to infer the selection of GO terms while the evidence code is labelled with "NONE". Evidence that spans multiple sentences is extracted and concatenated as a single long sentence. The

[3] http://geneontology.org/docs/ontology-relations.

[4] ftp://ftp.geneontology.org/go/www/images/diag-evCodeFlowChart.pdf.

inter-annotator-agreement of BC4GO is 42.7% in evidence sentence selection and 62.9% in GO term selection.

BioSentVec [27] is a sentence semantics representation model pre-trained on a vast volume of PubMed articles and clinical notes [28]. It can transform naturally written sentences into a lower-dimensional vector representation called sentence embeddings. Their model, utilising vectors of dimension 700, achieved competitive results in several biomedical sentence pair similarity prediction tasks [29, 30]. However, the performance of applying BioSentVec for the consistency estimation of two sentences in different level of biomedical information specificity is unknown.

BioPubMedBERT [24] is a contextual representation benchmark pre-trained on domain-specific full-text PubMed articles. It achieved competitive performance in many relation prediction tasks, such as the extraction of drug-drug interactions [31], gene-disease associations [32], and sentence-pair similarity estimation [29]. It uses special tokens "[CLS]" and "[SEP]" to mark the boundary of an entity pair and predict their relation type by mapping the last layer of "[CLS]" encoding into a linear layer for multi-type relation classification. However, the suitability of applying BioPubMedBERT model for open-world consistency inference of sentence pairs is unknown.

Previous work proposed a linguistic test suite for assessing the performance of automatic ontology concept recognition systems [25, 26]. The test suite is designed to extract a set of linguistic features of ontology terms such as the number of English words in the ontology term, the occurrence of digits or Roman numerals, or the length of associated evidence text. These linguistic features may impact the model's prediction uncertainty, which is a metric broadly used in the active learning field [33]. The uncertainties can be represented by the model's probability or entropy [34] for each prediction, with a lower probability or higher entropy indicating greater uncertainty. In principle, a robust model should perform best on more certain predictions, while flagging of various degrees of quality warnings to humans in real-world curation settings. The estimates of uncertainty also provides possibilities to quantify the model's performance during error analysis.

Method

Our approach combines a specific data source with modelling methods. We now introduce each of these components.

Data

There are no existing GOA resources with labelling of different types of (in)consistencies grounded at evidence level. Thus, to obtain suitable data, we transform consistent GOA available in the BC4GO corpus, generating instances of the four types of inconsistencies we have identified. We use 100 full-text articles from BC4GO to generate a training set and 49 articles to generate a test set. We randomly sample 20% of the generated instances from the training set to form a development set. By doing so, the generated test set is assured to be independent for post-modelling evaluation. We ensure that over 75% of the selected GO terms in the test set do not occur in the training or development set, enable the evaluation of the GO open-world assumption.

To simplify our study, we focus on detecting the main single type of inconsistency in each individual record. Thus, we assume that a GOA can only be in one type of (in) consistency, and further that they are independent of each other. We assume the gene product (usually represented by unique GeneID) and the organism described by the evidence text are consistent with that annotated by the GO term and only focus on detecting the (in)consistencies in gene function descriptions (Eq. 1, where y_* denotes a specific type of (in)consistency, ε denotes evidence information, θ denotes GO information, and γ denotes evidence code).

$$\begin{aligned} &P(y_j|y_i, \epsilon, <\theta;\gamma>) = 0,\ \ if\ P(y_i) = 1\ \ and\ \ i \neq j \\ &P(y_j|y_i, \epsilon, <\theta;\gamma>) = P(y_j|\epsilon, <\theta;\gamma>),\ \ if\ \ P(y_i) \neq 1 \end{aligned} \tag{1}$$

Each GOA instance contains two major components: GO information including the GO term ID, the GO term string and the term definition; and evidence information including evidence texts and spans, evidence codes, section information marked by the location of the evidence text in the article, gene name, gene identification, and gene synonyms. All information is directly extracted from BC4GO annotations or retrieved from the database using QuickGO [35]. The process for preparing each type of (in)consistent instance is described below. We also discuss potential bias in generating different types of inconsistencies under each bullet point.

- *Consistent* GOA from BC4GO: We extract GO annotations from full-text articles within BC4GO and transform them to produce instances of consistent GOA.

  We concatenate evidence text that spans more than one sentence in any GO annotation into a single sentence. We use QuickGO [35] to retrieve any information that was not originally provided in BC4GO annotations such as the GO definition. We remove any GOA in which the GO term is indicated as being obsolete on QuickGO.

  The quality of consistent GOA is provisioned by expert curators. We assume annotation of consistent instances in BC4GO is gold-standard.
- Type A—*Contradictory* description of gene regulation function: We apply keyword matching on GO terms in the transformed consistent GOA instances and swap the mention of any "positive regulation" with "negative regulation" and vice versa. We use the manipulated GO term to retrieve associated information such as GO identification, GO definition, and GO synonyms using QuickGO.
  Potentially incorrect generation of Type A inconsistencies can be caused by misannotations of original consistent instances in BC4GO corpus. For example, if curators incorrectly labelled a Type A instance as a consistent instance. This automatic synthetic strategy will modify it as a consistent instance indeed while incorrectly label it as Type A.
- Type B—*Over-specific* GO term: We retrieve a list of direct descendants with either "is_a" or "part_of" relationship to each GO term in every consistent GOA instances using QuickGO and manually assure these descendants are over-specific against the evidence texts. If a GO term in the consistent GOA is the leaf on the GO DAG, we will skip synthesising its over-specific inconsistencies.
  We use two alternative strategies to select an over-specific GO term from the retrieved descendants and use it to manipulate the consistent GOA into Type B

GOA. (1) We replace the GO term in a consistent GOA with a randomly selected over-specific descendant; (2) We replace the GO term in a consistent GOA with the direct descendant of that term that has the greatest word overlap [36]. For example, "feeding behavior (GO:0007631)" has one overlapping word with descendant "suckling behavior (GO:0001967)" and two overlapping words with descendant "regulation of feeding behavior (GO:0060259)". Thus, the second descendant will be selected for the replacement.

Two sets of Type B instances are generated individually based on the two strategies and were used to build different inconsistency detection models for comparative study. The second strategy may be biased by the word overlap similarity measure. For example, "feeding behavior (GO:0007631)" entirely overlap with "larva feeding behavior (GO:0030536)" but partially overlap with "drinking behavior (GO:0042756)". The second strategy will exclude the partially overlapping terms as candidate for replacement although it is also semantically similar to "feeding behavior".

- Type C—*Unsupportive* evidence text: We produce unsupportive variants of each consistent GOA by replacing the evidence sentence with another piece of semantically similar but unsupportive text from the same article. The GO term string in the original consistent GOA occurs as keywords within the chosen text but does not express meaningful gene function information.

  To find these texts, we refer to the task description of BioCreative IV [20] which states that text that is not annotated with a GO term can be treated as unsupportive. We extract unsupportive texts from the BC4GO corpus by article and segment them into sentences using TextBlob [37]. Each segment is considered as a piece of unsupportive evidence. Then, we start to iterate through every consistent GOA. To find certain pieces of texts that may be confused with valid evidence texts in each consistent GOA, we first apply GO concept recognition as implemented by [16] in the CCP-NLP-Pipelines to recognise any mention of a GO term in these unsupportive evidence sentences. We then pair each GO concept recognised unsupportive sentence with Consistent GOA instance in the same article that share the same GO concept. If there is no GO concept being recognised in the evidence sentence, we will skip synthesizing its unsupportive inconsistencies.

  In order to select an unsupportive sentence that is most similar to the evidence sentence in the consistent GOA, we represent the sentences as embeddings using BioSentVec [27] and calculate the cosine similarity between each pair of matched evidence sentence and unsupportive sentence. We produce the final instance by replacing the evidence sentence in the consistent GOA with the unsupportive sentence that is most similar. We update information in manipulated GOA such as evidence section information and formalise it into Type C GOA instance.

  This strategy may be biased in three aspects. Firstly, the sentence segmentation in TextBlob may incorrectly split a single sentence into two pieces and lead to grammatically ill-formed evidence sentence. Secondly, the CCP-NLP-Pipelines may fail to recognise an explicitly appearing GO concept and exclude it's associated sentence as a candidate for modification. Thirdly, the measurement of semantic similarity between two sentences using BioSentVec and cosine similarity do not guarantee a perfect quantification of semantic expression in gene function infor-

Table 2 The number of generated instances in each (in)consistency category

		100 Articles		49 Articles
		Train set	Dev set	Test set
*	*Consistent*	1466	367	1579
Term inconsistency	(A) *Contradictory*	160	40	128
	(B) Over-specific	1172	294	1231
	(C) Unsupportive	403	101	1579
Evidence code inconsistency	(D) Error-code	958	239	897

mation. Thus, replacing the tools with other alternatives in the three aspects can lead to the generation of different Type C inconsistencies.

- Type D—*Erroneous evidence code*: We select consistent GOA instances where evidence sentences are experiment type, based on an experimental type evidence code label. Those are "`IDA`", "`IMP`", "`IPI`", "`IGI`", "`IEP`". We exclude the selection of summary-type evidence sentences as they do not support the selection of an evidence code. We iterative through the left GOA instances and replace the evidence code of each with another code randomly selected from the decision tree mentioned in "Background" section. For example, we replace "`IMP`" with "`IGI`" in Table 1 (Type D inconsistency example).

After generating GOA instances, we manually confirmed the true (in)consistency of each automatically generated instance and the associated information. While we did not have a formal manual annotation process, our approach of targeted manipulation of annotated examples leads to reliable labels. The statistics of the generated dataset is shown in Table 2.

Modelling and data generation strategy

Baseline

We set up two baselines using a prior-biased classifier and section information rule-based model work for each modelling task.

The prior-biased classifier will make predictions according to the distribution of labels in the training set (Table 2). For example, the probability for prior-biased classifier to predict a new instance as *consistent* in the first task is 0.46, and second task is 0.60.

The section information rule-based model exploits different distributions of section information among instances shown in Table 3. This model will predict any instance as *unsupportive* if its evidence section information belongs to either Background, Supporting Information, Supplementary, or Other. Otherwise, the instance will be predicted as *consistent*. Specifically, only 4 out of 3480 instances appear in the Conclusion section in the test set. Thus, predicting them as either consistent or inconsistent will not statistically impact the overall performance of the baseline. However, this rule-based model will not predict any instance as *contradictory*, *over-specific*, or *erroneous code*. To overcome this issue, we apply another prior-biased classifier to process rule-based model outputs from the *consistent* type into these three types. For example, the probability of an instance that belongs to *over-specific* under the prior-biased rule-based model in the first modelling task is 0.42.

We explored the lexical distance between evidence text and the concatenation of GO term and GO definition using the Jaccard Index measure to gauge whether such a measure could provide a viable strategy for distinguishing consistent and inconsistent cases, or between inconsistency types. However, we found that different types of (in)consistencies have very similar similarity distributions. Thus, we abandoned the implementation of a baseline using a simple lexical distance measure; its performance would be close to a random guess and worse than the prior-biased classifier and section information rules we utilise.

Basic system

We model each generated instance as a paired sequence of "*[CLS]* evidence text *[SEP]* GO term + GO definition *[SEP]*" or "*[CLS]* evidence text *[SEP]* evidence code *[SEP]*" as input to BioPubMedBERT. We model the different types of (in)consistencies in two ways: (1) in a multi-class setting (*Model-Term*) that aims to distinguish between consistent, Type A, Type B and Type C inconsistencies by comparing evidence information with GO information; (2) and in a binary setting (*Model-Code*) for distinguishing between consistent and Type D inconsistencies by comparing evidence information with evidence code. In the basic system, two models are fine-tuned on training and development sets generated using random GO descendant selection based on the strategy mentioned in "Data" for Type B instances.

Optimization of training set

In the system with training set optimization, alternative Model-Term and Model-Code variants are fine-tuned on a collection of training and development set using the term overlap weighting strategy introduced in "Data" for Type B instances within 20 articles. The generation of Type B instances from the remaining 80 articles follows the original strategy for preventing the model biases towards greater overlapping GO terms. The test set is retained unmodified. This optimisation aims to boost the model's performance in distinguishing different types of instance where the semantics of GO terms is very similar, such as when "feeding behavior (GO:0007631)" and "suckling behavior (GO:0001967)" are associated with the same piece of evidence.

Addition of evidence section information

The evidence section information is first marked by the document section title in BC4GO corpus where the evidence text located and further normalised into 10 categories. The distribution of normalised section information in *consistent* and *unsupportive* instances is illustrated in Table 3. The distribution of evidence section information is consistent with a previous statistical report on BC4GO corpus [23] where a majority of GO annotations are supported by evidence text within the results and discussion section. The *contradictory*, *over-specific*, and *erroneous-code* instances are generated without manipulating the evidence text from the original consistent GOA instances. Thus, they retain the canonical distribution of evidence section information with Consistent

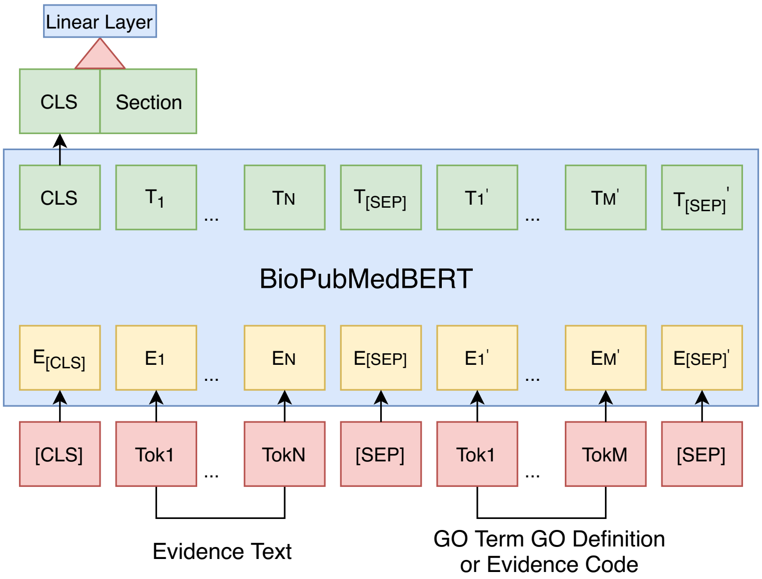

Fig. 1 Concatenation of evidence section encoding with [CLS] encoding in classical BioPubMedBERT model. (*Tok** denotes a naturally written token, *E** and *T** denote a token embedding, *Section* denotes section encoding, *[CLS]* and *[SEP]* are special tokens that mark the boundary of a sequence pair.)

Table 3 Distribution of evidence section information across *Consistent* type and (C) *Unsupportive* type instances within training set

Section information	Consistent (%)	(C) (%)
Title	1.2	1.8
Abstract	11.4	3.4
Introduction	2.7	13.4
Background	–	1.2
Materials and method	3.9	12.6
Results and discussion	80.6	53.3
Conclusion	0.2	0.6
Supporting information	–	2.8
Supplementary	–	10.5
Other	–	0.6

GOA. The *unsupportive* instances are generated by replacing the original evidence text with unsupportive evidence sentences and therefore have different distribution of section information from consistent GOA. We concatenate the 1-dimension section encoding with 768-dimension *[CLS]* encoding in BioPubMedBERT's last hidden layer and forward to a linear layer in Pytorch (Fig. 1).

Experiment design

We develop models to recognise the five types of (in)consistent GOA using the baseline setting and basic system. We then run additional experiments using training set optimisation and evaluate the impact of the addition of evidence section information using F_1 measure and *Precision*. We use BioPubMedBERT-uncased with 768 hidden states and the BERT-base architecture. We use the AdamW [38] optimiser with 0.01 weight decay and 300 warmup steps. We fine-tune model with 3 epochs, batch size 16 for the training set and batch size 64 for the development set. We use the huggingface AutoModelForSequenceClassification [39] framework for the fine-tuning implementation.

Evaluation metrics

In contrast to the existing literature-based GOA resources in GO consortium database where GO annotation is linked to evidence at the article level, we grounded annotation evidence to sentences via the BC4GO corpus. This strategy can better reflect the model's ability to detect (in)consistent GOA. Considering an article that has two evidence sentences supporting the annotation of the same GO term, where the first evidence sentence is correctly identified as being consistent to the GO term and the second evidence sentence is not, the *Precision* for consistent GOA recognition is 1 at the article level but 0.5 at the evidence level. *Precision* is appropriate for interpreting model's performance because the same GO term may be assigned to multiple evidence sentences in the same article. Once a single inconsistent GOA is detected in an article, all GOA that are linked to that article will be forwarded to curators for further inspection. Although there may be other inconsistent GOA linked to that article that were not detected by the system, they will still be manually reviewed. Thus, the detection of only one inconsistent GOA effectively achieves a perfect *Recall* at article level. A low volume of false negatives—missed (in)consistent GOA within a single article—can be tolerated in a real-world human-in-the-loop curation scenario.

We also use *Recall* and F_1 as evaluation metrics for the model's performance grounded at evidence level for each (in)consistency type. The evaluation is of the predictions over the test set, which is independent from any data point used in the previous fine-tuning stage.

To support further analysis of Model-Term's performance, we calculate the uncertainty (H) of each predicted label (i) using Shannon entropy; see Eq. 2, in which p_i denotes the probability that a GOA instance is of ith (in)consistency type.

$$H = -\sum_{i=0}^{3} P_i \log_2 P_i, H \in (0, 2) \tag{2}$$

We cluster test set instances into 15 collections using an uncertainty sampling strategy derived from [33]. We use two hyper-parameters $\tau(0.2 \leq \tau \leq 1.7, step = 0.1)$ and $\alpha = 0.1$ to represent the boundary of each sample in which the uncertainty of any prediction is between $\tau - \alpha$ and τ. This strategy can sample the same instance into more than one consecutive collection where τ represents the aggregated uncertainty of that collection.

We draw on the linguistic test suite of [25, 26] to define several metrics that characterise various linguistic aspects of GOA or GOA-evidence text pairs.

The Pearson correlation coefficient between the scores on these metrics and prediction uncertainty can then be investigated to provide insight into how uncertainty varies with linguistic characteristics. This analysis can be done either by taking the sum of all the instances metric scores and divided by the number of instances within a collection

(*Per-Instance*), or in aggregate across a collection (*Per-Collection*). Some metrics only support (*Per-Collection*) analysis as the aggregated uncertainty value of each sampled collection equals to the value of hyper-parameter τ being set during the uncertainty sampling process.

Either per-instance or per-collection

GOLen: The count of tokens in a GO term split by the blank sign.

#GOLen-2 | feeding behavior (GO:0007631)

AlignRatio: The count of tokens in evidence text divided by the count of tokens in GO definition text.

#AlignRatio-0.32 | Evidence: CeCDC-14 and ZEN-4 are interdependent in their localization | GO definition: Any process that modulates the frequency, rate or extent of any process in which a protein is transported to, or maintained in, a specific location.

GEORatio: The count of word overlaps between GO term and evidence text divided by the GOLen.

#GEORatio-0.25 | GO term: regulation of protein localization | Evidence: CeCDC-14 and ZEN-4 are interdependent in their localization

Per-Collection

%ContainRoman: The percentage of instances that has the occurrence of Roman numerals in the GO term in each sampled collection.

#%ContainRoman-0.5 | feeding behavior (GO:0007631) | photosynthesis, light harvesting in photosystem II (GO:0009769)

%ContainDigit: The percentage of instances that has the occurrence of digital numbers 09 in the GO term in each sampled collection.

#%ContainDigit-0.5 | cellular response to interleukin-1 (GO:0071347) | cellular response to peptide (GO:1901653)

%ContainStop-OF: The percentage of instances that has the occurrence of stopword "of" in the GO term in each sampled collection.[5]

[5] Note that GO terms are drawn from strictly controlled vocabularies where the stopword "of" expresses a compositional structure, with a parent GO term often appearing within a child term. Thus, this trick of counting the occurrence of such typical compositional structure via the stopword "of" is only feasible for GO terms but not GO definitions or naturally written evidence text.

Table 4 The performance of model-term and model-code in different modelling tasks, and in comparison with two baselines using *Precision* (P), *Recall* (R), F_1 measures grounded at evidence level in each (in)consistency type and *Micro-Precision* (P*), *Micro-Recall* (R*) averaged over every predicted instances in the test set

	Model-term											
	Consistent			**(A)**			**(B)**			**(C)**		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Basic system	0.54	0.70	0.61	0.48	0.29	0.36	**0.79**	0.48	0.60	**0.65**	0.96	**0.78**
+Training Opt	**0.74**	**0.71**	**0.72**	**0.54**	0.33	**0.41**	0.76	**0.57**	**0.65**	0.61	0.93	0.73
+SectionInfo	0.69	0.65	0.67	0.46	**0.35**	0.40	0.77	0.52	0.62	0.52	0.96	0.68
+Opt & SectionInfo	0.69	0.64	0.66	0.45	0.31	0.37	0.75	0.51	0.61	0.50	0.96	0.66
Baselines	**First modelling task**											
prior-biased classifier	0.48	0.35	0.41	0.05	0.02	0.03	0.36	0.28	0.31	0.10	0.33	0.15
rule-based model	0.53	0.38	0.44	0.09	0.04	0.06	0.41	0.29	0.34	0.18	**0.99**	0.30
	Model-term			**Model-code**								
	Overall			**Consistent**			**(D)**			**Overall**		
	P^*	R^*	F_1	P	R	F_1	P	R	F_1	P^*	R^*	F_1
Basic system	0.64	0.64	0.64	0.75	0.50	0.60	0.31	**0.58**	**0.41**	0.52	0.52	0.52
+Training Opt	**0.69**	**0.69**	**0.69**	–	–	–	–	–	–	–	–	–
+SectionInfo	0.65	0.65	0.65	**0.82**	0.48	0.61	0.21	0.56	0.31	0.50	0.50	0.50
+Opt & SectionInfo	0.63	0.63	0.63	–	–	–	–	–	–	–	–	–
Baselines	**First modelling task**			**Second modelling task**								
prior-biased classifier	0.30	0.30	0.30	0.6	**0.64**	**0.62**	**0.41**	0.37	0.39	**0.53**	**0.53**	**0.53**
rule-based model	0.36	0.36	0.36	0.6	**0.64**	**0.62**	0.41	0.37	0.39	**0.53**	**0.53**	**0.53**

The highest metric scores for the identification of each type of (in)consistency is bolded

#%ContainStop-OF-0.5 | feeding behavior (GO:0007631) | regulation of feeding behavior (GO:0060259)

Results

Table 4 shows that the model is competitive in distinguishing consistent GOA from all other types of inconsistencies compared to the baseline, with the best performance of 0.74 *Precision* for Model-Term and 0.82 *Precision* for Model-Code. The training set optimisation and the addition of evidence section information further contribute to improving the *Precision* (+ 0.2 & + 0.15) in recognising consistent GOA. However, these two strategies do not demonstrate positive impact in distinguishing inconsistent GOA other than Contradictory (Type A). The performance of Model-Code on Type D inconsistency recognition is low, indicating that evidence code errors are difficult for the model to accurately identify.

Analysis of the uncertainty of predictions from Model-Term demonstrates a significant negative correlation between *Precision* and the aggregated uncertainty of predictions among Consistent, Type B and Type C (Fig. 2). Specifically, the model achieved above 0.9 *Precision* in recognising Type C GOA and above 0.8 *Precision* in recognising Consistent GOA among instances with prediction uncertainty lower than 0.2. The radius of scattered dots on the trending line represent the size of each sampled

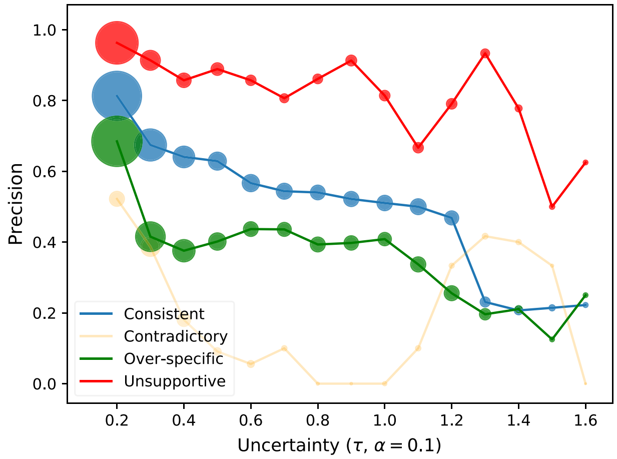

Fig. 2 The change in *Precision* with respect to different samples under an uncertainty sampling strategy with $0.2 \leq \tau \leq 1.7, \alpha = 0.1$and *step* = 0.1. The radius of dots represent the size of sampled predictions that support the calculation of metric score. (The detailed Pearson correlation coefficient values can be found in Table 6 and the size of each sampled collection by (in)consistency type can be found in Table 5)

Table 5 The size of each sampled collection using uncertainty sampling strategy by (in)consistency type

τ	Size of sampled collection			
	Consistent	**Contradictory**	**Over-specific**	**Unsupportive**
0.2	1022	90	1069	758
0.3	430	150	366	162
0.4	192	72	205	84
0.5	132	22	122	63
0.6	120	18	87	42
0.7	103	10	78	31
0.8	87	3	89	36
0.9	92	2	83	46
1.0	96	6	71	43
1.1	102	10	92	42
1.2	79	9	90	43
1.3	39	12	51	30
1.4	29	10	19	18
1.5	14	3	8	10
1.6	9	1	8	8

collection (Table 5), indicating that most of the predictions have low uncertainty. The Type A predictions do not correlate with prediction uncertainty due to the small collection size when $\tau > 0.4$ is limited (less than 50 instances in each). There is an uptrend of precision for Unsupportive typed instances when τ range between 1.2 and 1.4. This increase may be due to either the small collection size as well (less than 40 instances in each) or the occurrence of linguistic features (illustrated in the test suite in "Evaluation metrics") in the GO term, GO definition, or evidence text.

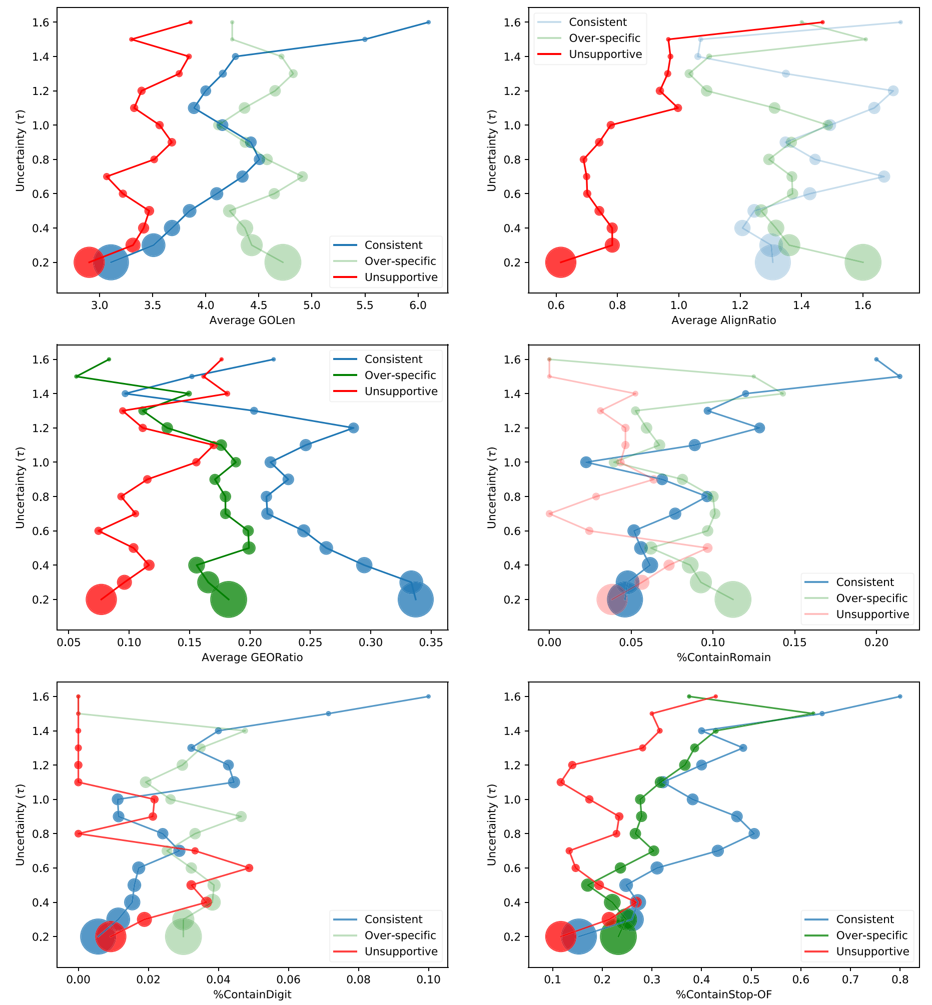

Fig. 3 The correlation trends between each metric in the linguistic test suite and aggregated uncertainty (τ) of sampled collections. The highlighted correlation trend lines have a significant Pearson correlation coefficient ($p < 0.05$). (A list of Pearson values either by instance or by aggregated collection of sample can be found in Appendix, Table 6)

Figure 3 and Table 6 demonstrate the metrics in the test suite all have significant correlation with either model's aggregated or per-instance prediction uncertainty. The highlighted correlation trend lines indicate the correlation is significant ($p < 0.05$) using a Pearson correlation coefficient measure. A detailed discussion is provided in Discussion.

Discussion

We explored the effectiveness of distinguishing Consistent GOA and the four kinds of inconsistencies. The basic setting achieves good results in identifying Type B (over-specific) and Type C (evidence unsupportive) GOA (Table 4). However, the models failed to generalise to extreme cases where one component of the input sequence pair is highly similar to components in other instances.

For example, if two pieces of semantically similar GO information (such as "regulation of feeding behavior: Any process that modulates the rate, frequency or extent of the

Table 6 * Pearson correlation and associated p value between each score and aggregated uncertainty (the aggregated uncertainty value of each sampled collection equals to the value of hyper-parameter τ being set during the uncertainty sampling process, the test suite scores are averaged by taking the sum of all the instances scores and divided by the number of instances within each collection); ^ The Pearson correlation and associated p value between each evaluation metric and per-instance prediction uncertainty

	Predictions	**Pearson R**	***p* value**
Precision *	Consistent	− 0.95	9e−08
	Type A	− 0.06	0.82
	Type B	− 0.85	7e−05
	Type C	− 0.70	3e−03
GOLen^	Consistent	0.27	6e−28
	Type B	− 0.04	0.14
	Type C	0.16	3e−07
AlignRatio^	Consistent	0.05	0.03
	Type B	− 0.09	9e−05
	Type C	0.16	2e−07
GEORatio^	Consistent	− 0.13	4e−08
	Type B	− 0.04	0.09
	Type C	0.1	9e−04
%ContainRoman*	Consistent	0.79	5e−04
	Type B	− 0.26	0.35
	Type C	− 0.46	0.08
%ContainDigit*	Consistent	0.81	3e−04
	Type B	− 0.44	0.10
	Type C	− 0.65	9e−03
%ContainStop-OF*	Consistent	0.81	2e−04
	Type B	0.80	3e−04
	Type C	0.58	0.02

behavior associated with the intake of food" and "positive regulation of feeding behavior: Any process that activates or increases the frequency, rate or extent of feeding behavior") are paired with the same piece of evidence text, the model has difficulty discriminating between them. This is reflected in the fact that most of the error cases are caused by mis-categorisation between *consistent* and *over-specific* GOA during evaluation on the test set (Table 7).

The training set optimisation contributes significant F_1 gain in distinguishing *consistent* GOA from inconsistent GOA (Table 4). This improvement results from the correct recognition of consistent instances that were previously falsely predicted as Type B inconsistencies in the baseline setting (Table 7). However, the training set optimisation strategy does not improve the ability to distinguish different types of inconsistencies, and performance at identifying Type C inconsistencies worsened. This is because Type B and Type C inconsistencies do not strictly follow the inconsistency independence assumption (Eq 1). The Type C instances can also be seen as a Type B scenario where the associated GO term is over-specific, which makes the evidence text not supportive enough. However, the training set optimisation strategy reinforces the categorisation of such instances into either *consistent* or Type B only, which leads to the mis-categorisation of some Type C instances as Type B. A potential solution may be to group Type

Table 7 The confusion matrix of model-term on the test set with basic, training set optimisation fine-tuning strategy

	Predicted labels			
	Consistent	**(A)**	**(B)**	**(C)**
True Labels				
Basic system				
Consistent	850	56	656	17
(A) Contradictory	15	61	52	0
(B) Over-specific	186	52	972	21
(C) Unsupportive	156	42	354	1027
+*Training set optimisation*				
Consistent	1164	56	309	17
(A) Contradictory	47	69	12	0
(B) Over-specific	239	39	933	20
(C) Unsupportive	199	44	379	957

B and Type C inconsistencies into one category or relax the independence assumption between the two classes via a multi-label classifier.

The evidence section information is a strong indicator for discriminating Consistent GOA from other types of inconsistencies. It outperforms the basic system but can also cause biases as the mixture of *Opt&SectionInfo* under-performs the *Training Opt* method (Table 4). This is because the distribution across section segments varies between Consistent and Type C inconsistencies (Table 3). The consistent GOA only appear in the sections of Title, Abstract, Introduction, Materials & Method and Results & Conclusion, while *Unsupportive* (Type C) inconsistencies can appear anywhere in the document.

The model is effective in assisting GOA inconsistency detection although it is not effective in distinguishing different types of inconsistencies (Table 4). In real-world curation settings, the performance of distinguishing consistency and inconsistencies is more important than distinguishing between different types of inconsistencies. This is because any inconsistent instances flagged by the automatic models will be passed to human curators for further review. The model does not need to precisely identify the specific type of inconsistency of the instance as human curators will make that judgement. In addition, an instance may be categorised into multiple types of inconsistencies at the same time. For example, a gene regulation contradictory instance can also be considered as an unsupportive instances. However, these issues will not affect the feasibility of the model in real-world use case.

Model-Code was not successful at identifying *Erroneous Code* (Type D) inconsistencies. This is because the relationship between an evidence code and evidence text is more complex than the relationship between a GO definition and evidence text. For example, in Type A inconsistencies, there are many lexical alignments from “negative regulation” within GO terms to “decrease”, “prevent”, “deactivate” within evidence texts. In Type B instances, over-specific GO terms often have large term overlap with the correct GO term. The pairwise semantic relation patterns of GO definition and evidence text in Model-Term are restricted. However, the Model-Code input of pairwise relations between evidence code and evidence texts do not have any text alignments or term overlaps. The assessment of consistency between the two requires a comprehensive knowledge inference process which relies on both the identified evidence text and prior knowledge or other text in the article. For example, the decision of whether an evidence text such as "*Here we show that a knock-out of the ybeB gene causes a dramatic adaptation block during a shift from rich to poor media and seriously deteriorates the viability during stationary phase. YbeB of six different species binds to ribosomal protein L14. This interaction blocks the association of the two ribosomal subunits and, as a consequence, translation*" should be labelled as "`IDA`" [PubMed Central article PMC3400551] is decided based on information such as whether or not there is a genetic mutation or allele variation, a 1-on-1 physical interaction, or the expression pattern of gene product. It requires that the result be determined through direct assay for the function, process, or component of the gene product. This requires a sophisticated process of considering the evidence text in relation to several decision rules rather than a direct association between the text and the evidence code.

Advanced assessment of Model-Term

Model-Term demonstrated strong potential for feasibility for real-world GO curation, particularly for the more confident predictions, as shown in Fig. 2. The results of linguistic test suite analysis revealed some critical linguistic features that have significant correlation with the model's prediction uncertainty. Specifically, the overlaps between GO term and evidence text (GEORatio) and the typical composition structure signalled by the occurrence of stopword "of" correlate well with prediction uncertainty. The correlation with this typical structure confirms that a valuable future research direction may be to develop better models of the hierarchical relationships between parent GO terms and more specific child GO terms. Additionally, we found that a longer GO term correlates with higher uncertainty in predictions of *consistent* and Type C; the differences of text length between GO definition and evidence text potentially influence the model's uncertainty in recognising Type-C GOA; the occurrence of Roman numerals and digits in the GO term demonstrate possibilities in influencing the prediction uncertainty of *consistent* and Type-C GOA as well.

We found a small number of error cases that were caused by the presence of biological or chemical formulas within GO definitions. These are not particular to any type of (in)consistency. For example, the definition for "geranyltranstransferase activity (GO:0004337)" is "*Catalysis of the reaction: geranyl diphosphate + isopentenyl diphosphate = 2-trans,6-trans-farnesyl diphosphate + diphosphate.*" We found that over 30% of formula-containing GOA instances are miscategorised by Model-Term.

Comparison with related work

At present, not every piece of information in the generated GOA instance is exploited by our modelling: for example, GO synonyms or larger context (such as the full paragraph) from where the evidence text was extracted were not used. Some researchers have developed methods to identify hypothesis statements or new knowledge from scientific literature using language meta-knowledge [40]. According to the GO curation guidelines, evidence is unsupportive if it only express the author's assumption of a gene function.

Thus, the analysis of evidence meta-knowledge may contribute to the identification of Type-C GOA.

Our proposed method uses *vertical* consistency estimation between the GO definition and evidence text as two texts are at different levels of specificity in expressing a gene product function. The GO definition describes the gene function more abstractly while the evidence expresses more detailed information. A previous related benchmark called GOCat [10] uses *horizontal* sentence pair similarity estimation [29] where two pieces of gene function description are on the same language specificity level. It first compares the semantic similarity between new evidence text and an old GO annotated evidence text. Then it selects relevant GO terms from the old evidence text to annotate the new one. This strategy has two shortcomings: it cannot deal with new knowledge, as described in the Background; and it can be biased toward frequently selected GO terms [12]. Our system can overcome these limitations and still maintain promising performance on a test set in which over 75% of GO terms are new (Table 4). Model-Term in the basic system achieved 0.68 *micro-averaged Precision* on the test instances with new GO terms, compared to 0.74 *micro-averaged Precision* on test instances with seen GO terms. The results demonstrate that our model is effective at processing new knowledge.

Conclusion

Continual monitoring of the consistency of GO annotation records in modern organism databases is important to maintain currency and quality of the information in these resources. We formally identify five major types of (in)consistent GO annotations that reflect the major GO annotation quality concerns for GO curation community. We propose a novel and efficient method to apply state-of-the-art text mining models to automatically detect these five major types of (in)consistent GO annotations, evaluated using an automatically generated data set. Our method satisfies the open-world assumption and is therefore robust to changes in the GO terminology.

We have demonstrated a novel method that can be adopted for real-world human-in-the-loop curation. Our implementation achieved 0.74 *Precision* for Model-Term and 0.82 *Precision* for Model-Code in distinguishing consistent from inconsistent GOA. This method can improve the efficiency of human curators by enabling curators to focus their efforts on correcting identified inconsistencies and by categorising these inconsistencies, therefore reducing the number of records that need to be manually reviewed.

Another strength is that the model has achieved competitive performance among predicted results with less prediction uncertainty, which can be used by human curators to further focus their efforts. We were able to further improve performance through training set optimisation and the addition of evidence section information. Through a detailed performance analysis using a linguistic test suite, we identified superficial linguistic features that may impact the model's prediction uncertainty.

In future work, we aim to produce a more comprehensive evidence-based GO annotation corpus focusing on inconsistencies. We will seek assistance from expert curators to test and extend the proposed methods on real-world database records with broader gene function perspective, and will specifically seek to improve the identification of evidence code inconsistencies. We will also examine the use of meta-knowledge analysis to improve the model's performance in identification of instances that lack supportive evidence. We will refine the modelling of semantic hierarchical relationship between parent and children GO terms.

Acknowledgements
Not applicable

Authors' contributions
JC contributed to the exploration, experiment, analysis and drafted the manuscript. NG JZ and KV contributed to the conceptualization of the research and in revising the manuscript. All authors read and approved the final manuscript.

Funding
Funding for this work was provided by an Australian Research Council Discovery Project grant, DP190101350, to investigators KV NG and JZ. The funding body played no role in the design or execution of the research.

Availability of data and materials
All data presented and analyzed in the present study was generated from the BC4GO corpus and retrieved from GO Consortium Database. Code for the generation of adversarial instances is accessible at https://github.com/jiyuc/BioConsistency. There were no additional materials created for this study.

Declarations

Ethics approval and consent to participate
Not applicable

Consent for publication
Not applicable

Competing interests
KV is a Section Editor for BMC Bioinformatics. KV reports receiving funding from Elsevier BV and speaking fees from Pfizer. All other authors declare no competing interests.

Author details
[1]School of Computing and Information Systems, University of Melbourne, Melbourne 3010, Australia. [2]School of Computing Technologies, RMIT University, Melbourne, VIC 3000, Australia.

Received: 12 April 2021 Accepted: 15 November 2021
Published online: 25 November 2021

References

1. Gene Ontology Consortium. Gene ontology consortium: going forward. Nucleic Acids Res. 2015;43(D1):1049–56.
2. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, et al. Gene ontology: tool for the unification of biology. Nat Genet. 2000;25(1):25–9.
3. Zhou N, Jiang Y, Bergquist TR. The CAFA challenge reports improved protein function prediction and new functional. Genome Biol. 2019;20:1.
4. Cozzetto D, Jones D. Computational methods for annotation transfers from sequence. Methods Mol Biol (Clifton, NJ). 2017;1446:55–67.
5. Gene Ontology Consortium. Expansion of the gene ontology knowledgebase and resources. Nucleic Acids Res. 2017;45(D1):331–8.
6. Ruch P. Text mining to support gene ontology curation and vice versa. Methods Mol Biol (Clifton, NJ). 2017;1446:69–84.
7. Balakrishnan R, Harris MA, Huntley R, Van Auken K, Cherry JM. A guide to best practices for gene ontology (go) manual annotation. Database. 2013.
8. Du Plessis L, Škunca N, Dessimoz C. The what, where, how and why of gene ontology-a primer for bioinformaticians. Brief Bioinform. 2011;12(6):723–35.
9. Škunca N, Altenhoff A, Dessimoz C, et al. Quality of computationally inferred gene ontology annotations. PLOS Comput Biol. 2012;8(5):1–11.
10. Gobeill J, Pasche E, Vishnyakova D, Ruch P. Managing the data deluge: data-driven go category assignment improves while complexity of functional annotation increases. Database. 2013.
11. Gaudet P, Dessimoz C. Gene ontology: pitfalls, biases, and remedies. Methods Mol Biol (Clifton, NJ). 2017;1446:189.
12. Haynes WA, Tomczak A, Khatri P. Gene annotation bias impedes biomedical research. Sci Rep. 2018;8(1):1–7.
13. Schnoes AM, Ream DC, Thorman AW, Babbitt PC, Friedberg I. Biases in the experimental annotations of protein function and their effect on our understanding of protein function space. PLoS Comput Biol. 2013;9(5):e1003063.
14. Tanenblatt M, Coden A, Sominsky I. The conceptmapper approach to named entity recognition. In: Proceedings of the seventh international conference on language resources and evaluation (LREC'10). 2010.
15. Bada M, Eckert M, Evans D, Garcia K, Shipley K, Sitnikov D, Baumgartner WA, Cohen KB, Verspoor K, Blake JA, et al. Concept annotation in the craft corpus. BMC Bioinform. 2012;13(1):161.
16. Funk C, Baumgartner W, Garcia B, Roeder C, Bada M, Cohen KB, Hunter LE, Verspoor K. Large-scale biomedical concept recognition: an evaluation of current automatic annotators and their parameters. BMC Bioinform. 2014;15(1):59.

17. Gilks WR, Audit B, De Angelis D, Tsoka S, Ouzounis CA. Modeling the percolation of annotation errors in a database of protein sequences. Bioinformatics. 2002;18(12):1641–9.
18. Škunca N, Roberts R, Steffen M. Evaluating computational gene ontology annotations. Methods Mol Biol (Clifton, NJ). 2017;1446:97.
19. Jones CE, Brown AL, Baumann U. Estimating the annotation error rate of curated go database sequence annotations. BMC Bioinform. 2007;8(1):1–9.
20. Mao Y, Van Auken K, Li D, Arighi CN, McQuilton P, Hayman GT, Tweedie S, Schaeffer ML, Laulederkind SJ, Wang S-J, et al. Overview of the gene ontology task at biocreative IV. Database. 2014.
21. Poux S, Gaudet P. Best practices in manual annotation with the gene ontology. Methods Mol Biol (Clifton, NJ). 2017;1446:41–54.
22. Thomas P. The gene ontology and the meaning of biological function. Methods Mol Biol (Clifton, NJ). 2017;1446:15–24.
23. Van Auken K, Schaeffer ML, McQuilton P, Laulederkind SJ, Li D, Wang S-J, Hayman GT, Tweedie S, Arighi CN, Done J, et al. Bc4go: a full-text corpus for the biocreative IV go task. Database. 2014.
24. Gu Y, Tinn R, Cheng H, Lucas M, Usuyama N, Liu X, Naumann T, Gao J, Poon H. Domain-specific language model pretraining for biomedical natural language processing. ACM Trans Comput Healthc (HEALTH). 2021;3(1):1–23.
25. Cohen KB, Roeder C, Baumgartner Jr WA, Hunter L, Verspoor K. Test suite design for biomedical ontology concept recognition systems. In: Proceedings of the seventh international conference on language resources and evaluation (LREC'10). 2010.
26. Groza T, Verspoor K. Automated generation of test suites for error analysis of concept recognition systems. In: Proceedings of the Australasian language technology association workshop. 2014. pp. 23–31.
27. Chen Q, Peng Y, Lu Z. Biosentvec: creating sentence embeddings for biomedical texts. In: 2019 IEEE international conference on healthcare informatics (ICHI). IEEE; 2019. p. 1–5.
28. Johnson AE, Pollard TJ, Shen L, Li-Wei HL, Feng M, Ghassemi M, Moody B, Szolovits P, Celi LA, Mark RG. MIMIC-III, a freely accessible critical care database. Sci Data. 2016;3(1):1–9.
29. Soğancıoğlu G, Öztürk H, Özgür A. Biosses: a semantic sentence similarity estimation system for the biomedical domain. Bioinformatics. 2017;33(14):49–58.
30. Wang Y, Afzal N, Fu S, Wang L, Shen F, Rastegar-Mojarad M, Liu H. Medsts: a resource for clinical semantic textual similarity. Lang Resour Eval. 2020;54(1):57–72.
31. Herrero-Zazo M, Segura-Bedmar I, Martínez P, Declerck T. The DDI corpus: an annotated corpus with pharmacological substances and drug–drug interactions. J Biomed Inform. 2013;46(5):914–20.
32. Becker KG, Barnes KC, Bright TJ, Wang SA. The genetic association database. Nat Genet. 2004;36(5):431–2.
33. Settles B. Active learning literature survey. 2009.
34. Shannon CE. A mathematical theory of communication. ACM SIGMOBILE Mob Comput Commun Rev. 2001;5(1):3–55.
35. Binns D, Dimmer E, Huntley R, Barrell D, Odonovan C, Apweiler R. Quickgo: a web-based tool for gene ontology searching. Bioinformatics. 2009;25(22):3045–6.
36. Pesquita C. Semantic similarity in the gene ontology. Methods Mol Biol (Clifton, NJ). 2017;1446:161.
37. Loria S. Textblob documentation. Release. 2018;15:2.
38. Loshchilov I, Hutter F. Decoupled weight decay regularization. In: International conference on learning representations. 2018.
39. Wolf T, Chaumond J, Debut L, Sanh V, Delangue C, Moi A, Cistac P, Funtowicz M, Davison J, Shleifer S et al. Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations. 2020. pp. 38–45.
40. Shardlow M, Batista-Navarro R, Thompson P, Nawaz R, McNaught J, Ananiadou S. Identification of research hypotheses and new knowledge from scientific literature. BMC Med Inform Decis Mak. 2018;18(1):46.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.