Categorizing Stack Overflow Questions With A Tag Hierarchy

Philip Roozendaal
Supervisor(s): Dr. Maliheh Izadi, Prof. Dr. Arie van Deursen
EEMCS, Delft University of Technology, The Netherlands
22-6-2022

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

Abstract

Software Question & Answer platforms such as Stack Overflow allow users to annotate their posts with tags in order to help organize them and aid in their discoverability. This work sets out to study the machine learning techniques used to determine these tags automatically, and see how, and to what extent, these determinations could be improved by organizing the tags in a hierarchical fashion and using this hierarchy as a heuristic. This is a multi-label classification problem. The tag hierarchy is built by clustering the tags by subject, connecting these clusters, and then fine-tuning the results. Then, after gathering and preparing the training data consisting of Stack Overflow question titles, bodies and tags, a DistilBERT based multi-label classifier is trained and serves as the baseline. Then, this baseline is extended such that it incorporates the newly constructed hierarchy in its final predictions. Finally, the classifier is evaluated on the accuracy of its predictions, and on its usefulness, which is derived from a survey performed with expert users in the area of Computer Science. The resulting model evaluation results in an LRAP score of 54% and an F1 score of 65%, improving over the baseline with 2% and 2% respectively.

1 Introduction

As of the writing of this document, Stack Overflow, a widely known Question & Answer (Q&A) platform in the area of software development, hosts over twenty-two million questions, with thousands of new questions being asked every day[1]. This increasingly large number of questions can make it difficult to navigate the site: how does a user know if a question has already been asked? How does a user find questions that they have knowledge of and would like to answer? One of the tools Stack Overflow provides users to help with this, is the ability to annotate questions with tags.

When users assign tags collaboratively, they might use different notations for representing the same concept. Some users use hyphens to separate words in a tag, others use underscores. Some use capitalized letters where others do not. And some users use abbreviations where others spell out the full name. Because of this, the number of distinct tags increases drastically while their quality degrades [4].

As an alternative to collaborative tagging, machine-learning based classifiers can be deployed to automatically assign tags to a post. A lot of work is being done in the field of multi-label classifiers, and even in its specific application to Q&A sites [1, 5, 7, 8, 11].

This work tries to improve the multi-label classification of tags for Stack Overflow posts, by organizing the set of candidate tags into a hierarchy, and using this hierarchy to add a bias towards tags that appear close to each other. This improvement is determined by the resulting model's Label-Ranking-Average-Precision (LRAP) and F1-scores and by the usefulness is evaluated by performing a survey with expert-users in the area of Computer Science.

[1] https://stackoverflow.com/questions

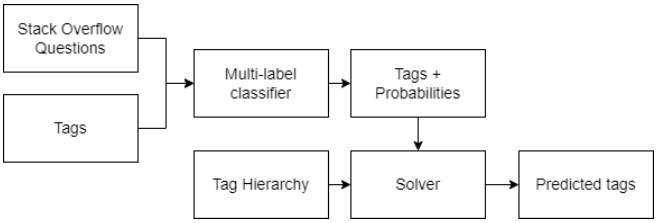


Figure 1: Problem Overview

The resulting model evaluation results in an LRAP score of 54% and an F1 score of 65%, improving over the baseline with 2% and 2% respectively.

2 Problem Definition

This work sets out to construct a categorizer for Stack Overflow posts. These posts are categorized by one or multiple tags, and therefore, this is a multi-label classification problem. Additionally, Stack Overflow posts can be assigned up to five tags maximum[2].

A set of Stack Overflow posts $P = \{p_1, p_2, ..., p_n\}$ are given where each post p_i consists of the post's title and body, and the candidate set of all possible tags $T = \{t_1, t_2, ..., t_n\}$ are provided.

Problem 1: The first problem is to assign to each tag $t_i \in T$ a score representing its probability of being a suitable annotation for the post.

Problem 2: The second problem is to apply a bias to tags that are in close proximity in the hierarchy. Given the hierarchy as a set of tags $H = \{t_1, t_2, ..., t_n\}$ where each tag is a node in a directed acyclic graph (DAG), and also given the set of predicted tags for a post $T_p = \{t_{p,1}, t_{p,2}, ..., t_{p,n}\}$, find the top five tags that together, with the inverse of their distance in the hierarchy, have the highest probability of being a suitable annotation for the post. The distance between two tags in the hierarchy is defined as the ratio between the number of edges they need to traverse to reach a common ancestor, and the maximum possible distance they could have (two times the maximum depth of the tree).

[2] No official source documents this, but this limit is enforced nonetheless.

3 Background

This work makes uses of the state-of-the-art language representation model DistilBERT [9], which is an extension of the Bidirectional Encoder Representations from Transformers (BERT). BERT exploits the attention mechanism, which helps draw connections between any parts of a sequence of text. A pre-trained BERT model can be further fine-tuned with only a single additional output layer to create state-of-the-art models [3]. DistilBERT is developed to pre-train a smaller model compared to BERT, while being faster and by-and-large maintaining the accuracy. DistilBERT decreases the model size by 40%, while retaining 97% of its language understanding capabilities and being 60% faster.

4 Related Works

A lot of work is actively being done in the areas of machine learning, multi-label classification problems and recommender systems.

In their work, Izadi et al. [5] study multi-label classification techniques for recommending topic tags to GitHub repositories using a select number of high-quality featured topics rather than the ever-growing list of varying-quality user-defined topics. This is accomplished by mapping the user-defined topics to featured topics before training the supervised models. The models consist of the traditional classifiers Multinomial Naive Bayes, Gaussian Naive Bayes, Logistic Regression, Facebook's FastText and HuggingFace's DistilBERT. The results are a recommender system achieving Recall and LRAP scores of 0.890 and 0.805 respectively for the top 5 recommended topics. However, this work does not take the correlation between tags into consideration, and future work could extend on this.

Kavuk and Tosun [7] set out to predict suitable tags for posts on Stack Overflow. First they scrape the raw data, which they then pre-process and transform into features using Latent Dirichlet Allocation. These data are further used to construct two sorts of classifiers: 1) one-against-all classifiers for each of the top 15 most used tags, in order to predict if the tag belongs to a given post, and 2) a multi-tag predictor which suggests top 5 tags for a given post. The individual one-against-all classifiers reach up to 90% recall rates and obtain 75% on average to predict one tag per post. The multi-tag predictor has a recall of 55% and an F1 score of 39%. The inaccuracy in their results are partly attributed to the problems of collaborative tagging: redundancies, inconsistent levels of abstraction, inconsistent spelling, lack of understanding from the user, etc. For this reason their model performs better for predicting more specific tags that were less frequently used. The model described in this study could potentially be improved by mapping the collaboratively generated tags to a set of featured topics as described in the study performed by Izadi et al. [5].

Ali et al. [1] propose a machine learning model to classify the architectural knowledge related posts found on Stack Overflow into predefined categories (i.e. analysis, synthesis, evaluation, and implementation). To find the most suitable model, they apply various combinations of classic machine learning algorithms. They then perform comparative analysis by defining a suitability method which considers both the classifier's accuracy and execution time to determine the most suitable model. They find that while the Support-Vector-Machine (SVM) model obtains the highest accuracy, it is actually the Naive-Bayes (NB) model that obtains the highest suitability due to its overall lower computational cost. This study shows that it's feasible to capture and classify architectural knowledge from sites like Stack Overflow with high accuracy. This in turn could lead to developers being able to find important information significantly faster.

TagMulRec [12] is a tool that can recommend tags for software information sites. In particular, it has been evaluated with Stack Overflow, AskUbuntu, AskDifferent and Freecode. TagMulRec first preprocesses the raw data (remove tags that are not used frequently, remove stop words, etc.), then assigns a unique index to each software object description, and defines a subroutine for computing similarity scores to construct target candidate sets for software objects that are semantically similar. According to empirical analysis, TagMulRec is both accurate and scalable for use with large-scale software information sites with millions of software objects and thousands of tags.

Liu et al. [8] extends TagMulRec, in their paper called regarding their new tool: FastTagRec. In this paper, Liu et al. propose a tag recommendation architecture similar to the continuous bag of words model (CBOW). FastTagRec is able to very accurately infer tags for new postings. From empirical evaluation, FastTagRec showed to be both more accurate and three orders of magnitude faster than the comparable tool: TagMulRec [12]. This study contributes largely to the understanding of how to improve the performance of a recommender system.

In their paper, Chen et al. [2] study the challenges that come with creating a hierarchical multi-label classifier (HMLC). While traditional classifiers assume two labels are distinct, this is not necessarily the case for an HMLC (where one label might be a more concrete or abstract version of a different label). They then design a new Hyperbolic Interaction Model (HyperIM), which is designed to learn label-aware document representations. This hierarchy is modeled in hyperbolic space. The results demonstrate that the new model can realistically capture the complex data structures.

In their study, Xiao et al. [10] design and evaluate a multi-label text classification system (MLTC), which aims to find the most relevant labels for a given text document. This paper proposes a Label Specific Attention Network (LSAN) to learn new document representations. LSAN takes advantage of label semantic information to determine the semantic connection between labels and documents. In their evaluation, they compare it to various baselines: XML-CNN, SGM, DXML and AttentionXML. They find that compared

to these models, LSAN performs better in terms of accuracy, especially on the prediction of low-frequency labels. This study is performed with only label texts as data, but further studies could extend on this by making use of more information such as the description for a label, or its position in a label hierarchy. HyperIM consistently outperforms all of the baselines (EXAM, SLEEC and DXML). One of the causes this is attributed to is that HyperIM benefits from the retention of the hierarchical label relations.

Xu et al. wrote a paper [11] in which they propose a specialized deep learning architecture called Post2Vec, which can extract distributed representations of Stack Overflow posts (i.e. title, description and code snippets). To evaluate the system. Xu et al. compared the results to those of state-of-the-art tag recommendation systems that also employ deep neural networks. As it turns out, Post2Vec achieves 15-25% improvement on its F1-score@5 and at a lower computational cost.

See Table 1 for a brief overview of how each related work addresses key topics.

5 Approach

The first step in the building and evaluating the hierarchy-informed question categorizer is to build the hierarchy itself. The next step is to obtain and pre-process Stack Overflow question data that such that it can be used for training the machine learning model. Then, a baseline model is evaluated in order to compare any effects the additional heuristic of the tag hierarchy might have. Furthermore, the baseline is extended with the additional effect of the tag hierarchy. Finally, the resulting model and the effects of the tag hierarchy are evaluated on their accuracy and usefulness.

To build the hierarchy, the set of tags and their relations obtained from [5] are clustered based on the topics they represent. These clusters are then connected to newly introduced label nodes that represent the collection of both clusters (i.e. the "front-end" and "back-end" clusters are joined under a new label "full-stack"). This process is repeated until a complete hierarchy is formed.

The Stack Overflow data is obtained directly with SQL from the Stack Exchange Data Explorer[3]. The obtained textual data is first pre-processed to reduce irregularities, and then one-hot-encoded so that it can be used for training the model. One-hot encoding is an encoding where a set of textual labels are converted into a vector of binary values, indicating for each position whether the label is present or not.

The baseline uses HuggingFace's DistilBERT, an extension of the state-of-the-art transformer model BERT (Bidirectional Encoder Representations from Transformers). DistilBERT can reduce the size of a BERT model by 40%, while still retaining up to 97% of its language understanding capabilities and being 60% faster. [9]

For any given Stack Overflow post, the baseline will be able to predict a score for each possible tag, indicating the probability of that tag belonging to the given post. To extend the baseline with the tag hierarchy, a new score for each tag is calculated. This is accomplished by using the baseline's predictions to construct a clique: a graph where each node is connected to every other node by an edge. To construct the graph, each tag is a node, and the distance to any other node is the distance between those two nodes in the tag hierarchy. The new predictions are then the top N nodes and with their combined sum of edges and tag probabilities yield the highest possible score within the clique.

Finally, to evaluate the accuracy of the model, the F1- and LRAP scores of the hierarchy model are compared to those of the baseline. The usefulness of the model is evaluated by performing a survey with expert-users in the domain of Computer Science.

See Figure 2 for an overview of the approach.

6 Experiment Design

This section briefly describes the concrete details of the experiment configuration, such that when the same configuration is used with the aforementioned approach, the same results should follow.

6.1 Research Questions

To provide an answer to how accurate and useful the categorizer developed in this paper is, the following sub-questions will be investigated:

RQ1: How accurate is the tag predictor for SO questions?

RQ2: How should the tag hierarchy be organized?

RQ3: How useful is the tag predictor for SO questions?

RQ1 is answered by evaluating the hierarchy model. Given the optimized hierarchy and trained model, the accuracy of the tag predictor is determined by calculating its LRAP and F1-scores.

RQ2 regards the organization of the tag hierarchy. There are different ways of structuring the hierarchy (E.g. "windows" as a tag could be placed under a "microsoft" cluster or alternatively under a "operating systems" cluster). To analyze these effects and find out what hierarchy leads to the best results, the model is evaluated with three different tag hierarchies.

[3] https://data.stackexchange.com/stackoverflow/query/new

Table 1: Related work overview

Reference	Q&A Platforms	Multi-label classification	Label hierarchy	Transformer-based model
Xu et al. 2021 [11]	x	x		
Kavuk et Tosun 2020 [7]	x	x		
Liu et al. 2018 [8]	x	x		
Izadi et al. 2021 [5]	x	x		x
Ali et al. 2021 [1]	x	x		
Zhou et al. 2017 [12]	x			
Chen et al. 2020 [2]		x	x	
Xiao et al. 2019 [10]		x		

RQ3 aims to answer how useful the question tag predictor is. That is, to what extent is it helpful to the average user. To address this question, a survey is performed with expert-users in the domain of Computer Science.

6.2 Evaluation Metrics

Standard evaluation metrics for multi-label classifications are used to evaluate the tag predictor's accuracy. This work uses LRAP, Precision, Recall and the F1 score.

Label-Ranking-Average-Precision (LRAP) is a metric that is often used in multi-label classification problems. This metric considers the rank of the predicted labels in its evaluation rather than just the (lack of) presence of a given label. Given a binary indicator matrix of the ground truth topics and the score associated with each topic, LRAP is defined as:

$$LRAP(y, \hat{f}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} \frac{1}{||y_i||_0} \sum_{j:y_{ij}=1} \frac{|\mathcal{L}_{ij}|}{\text{rank}_{ij}} \quad (1)$$

Where $\mathcal{L}_{ij} = \{k : y_{ik} = 1, \hat{f_{ik}} \geq \hat{f_{ij}}\}$
and $rank_{ij} = |\{k : \hat{f_{ik}} \geq \hat{f_{ij}}\}|$
where $|.|$ computes the number of elements in the set
and $||.||_0$ is the l_0 "norm".

The F1 score is defined as the harmonic mean between the model's Precision $\frac{tp}{tp+fp}$ and Recall $\frac{tp}{tp+fn}$ where tp, fp and fn are respectively true positives, false positives and false negatives. The F1 score can then be computed as: $F_1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$

The usefulness of the model is evaluated through performing the aforementioned survey. The survey's results indicate for each question, for each predicted tag, whether that tag was correctly or incorrectly added, omitted or given a different priority over other tags.

6.3 Hierarchy

To answer RQ2, three different hierarchies are evaluated. The first hierarchy is the original hand-crafted hierarchy. This hierarchy consists of 76 cluster nodes, with a maximum depth of 7 and a maximum cluster size of 239. The second hierarchy is a flattened version of the first with a total of 19 nodes, a maximum depth of 3 and a maximum cluster size of 169. The third and final hierarchy is a refined version of the first one, where the max cluster size is vastly reduced. This hierarchy consists of 86 nodes, with a maximum depth of 7 and a maximum cluster size of 59.

The tags used in the training of the model, are first organized in a hierarchy. The tags used to build the hierarchy come directly from the work of Izadi et al. [5]. This means that the Stack Overflow tags first need to be mapped to these shared tags. Fortunately, 525 of the Stack Overflow tags appear directly in the 864 tags listed in the works of [5]. The remaining tags will not be used for the experiment.

6.4 Dataset

To limit the amount of training data, and to train the model on recent trends, only questions posted after 2016 are used for training. Furthermore, only questions that feature at least one of the tags featured in the tag hierarchy are selected.

A total of 62'987 (including synonymous) tags are collected using the Stack Exchange Data Explorer, and a total of 20'000 questions are gathered. The question data includes the question title, the markup annotated body as displayed on the Stack Overflow website, and the tags assigned to them.

The question data is pre-processed before it is given to the machine learning model. This pre-processing consists of the following steps:

- Strip out all the code-blocks, XML-tags, URLs and paths.
- Convert all the remaining text to lower-case.
- Replace common abbreviations.
- Lemmatize the remaining text. This means that various inflected forms of a word are grouped together so they can be analysed as a single item.
- Strip out consecutive whitespace.

6.5 Baseline

As mentioned in the Approach section, DistilBERT is used for the baseline. This model is set up using the simpletransformers library [4] and is trained at 5 epochs, with a learning

[4] https://simpletransformers.ai/

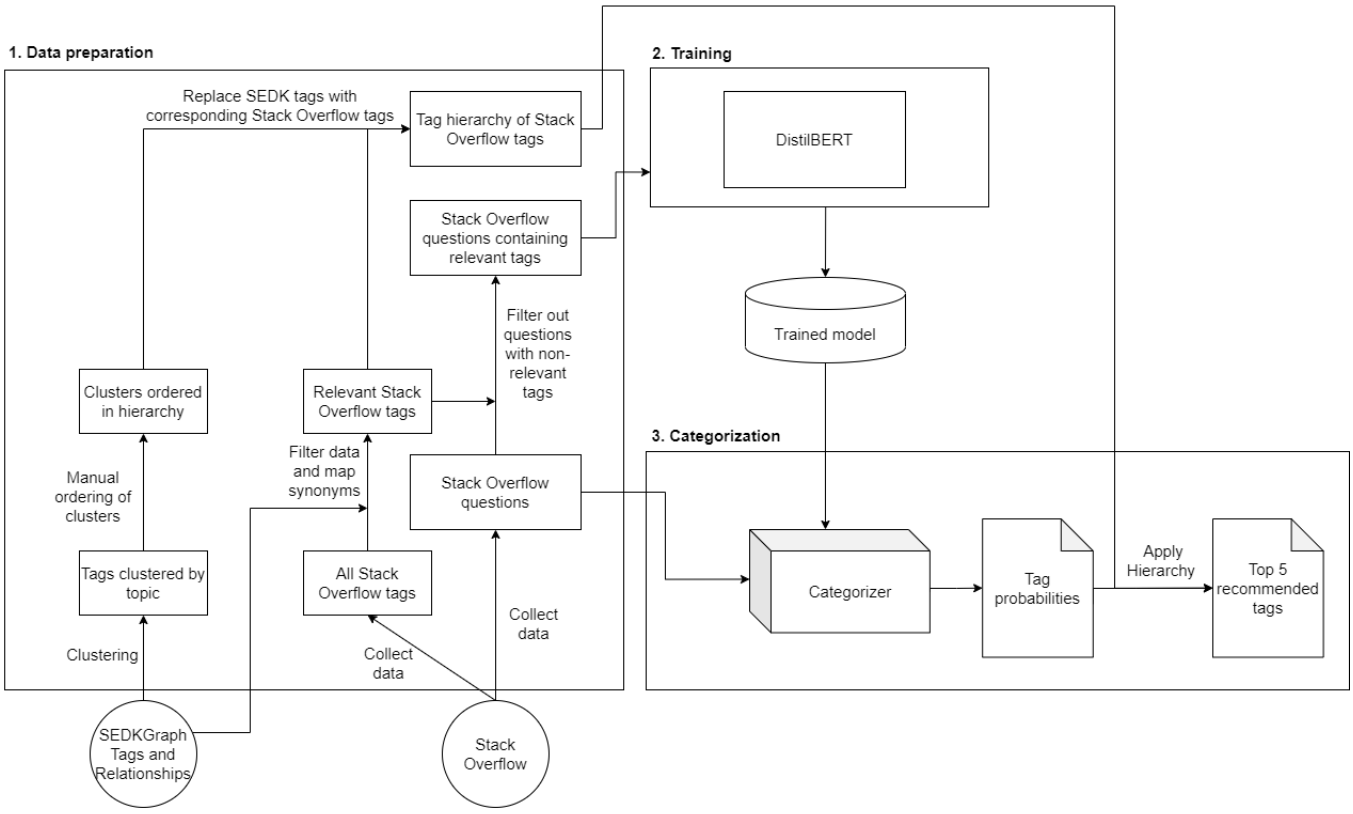


Figure 2: Approach Overview

rate of 1e-4 and tested using a 80%/20% split of the 20'000 gathered questions. The resulting scores of this model are an LRAP of 52%, with a precision of 78%, recall of 53% and an F1-score of 63%.

6.6 Survey

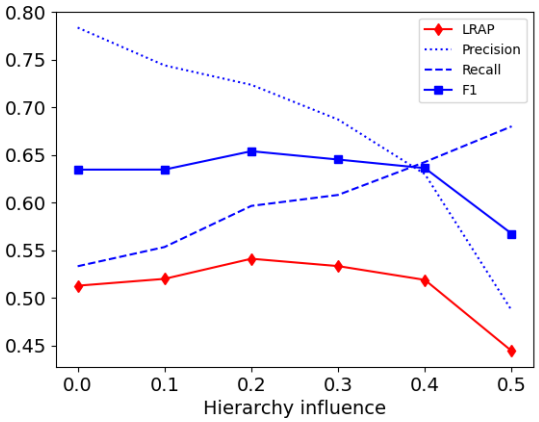
The usefulness of the model is evaluated through performing a survey with expert-users in the domain of Computer Science. The survey starts with a short series of questions regarding the participant demographics, and is followed by questions about predictions made for ten different Stack Overflow posts. For half of the questions, the participant is shown the Stack Overflow post, followed by the set tags predicted by the model. They are then asked to, for each tag individually, rate its usefulness and relatedness on a Likert [6] scale of 0-5. For the other half of the questions, the participant is asked to select tags they deem fitting out of a pool of ten tags consisting of the post's assigned tags, the baseline predicted tags, and the new model's predicted tags.

7 Results

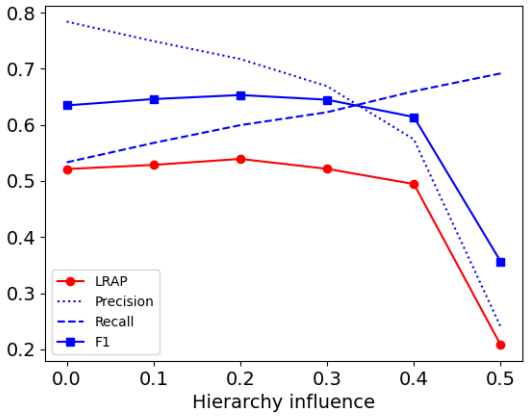
Figure 3a, 3b and 3c show respectively for each hierarchy, how the LRAP, Precision, Recall and F1-scores change as the hierarchy is applied with a different weight. The highest scores are achieved with hierarchy 3 when applied with a weight of 0.2, achieving an LRAP score of 54% and an F1 score of 65%, improving over the baseline with 2% and 2% respectively **RQ1, RQ2**.

Fifteen individuals participated in the survey. Slightly more than half of all participants indicated they were around Master level or higher in the field of Computer Science. The remaining participants indicated to be around Bachelor level. The survey results show that for some questions the baseline's predictions were more useful, for some it was the same, and for five out of the ten main questions, the hierarchy model's predictions either added a somewhat useful tag (average rating above 2 out of 5), or correctly gave a higher ranking to one or more tags compared to the baseline.
Overall, compared to the baseline, 7 tags got their rank correctly increased, 1 got it incorrectly decreased, 6 tags that were deemed useful were added, and 1 tag that was deemed useful was removed. **RQ3**.

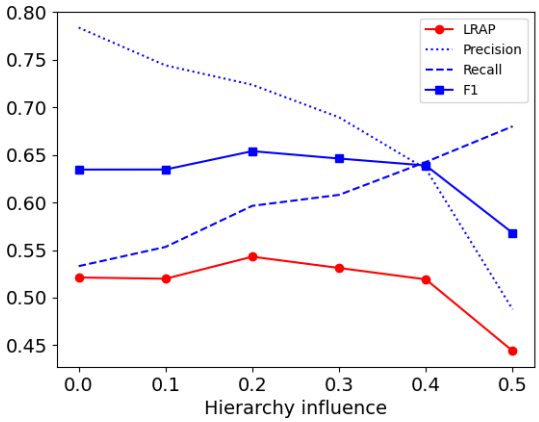
For a complete overview of the survey results, see Table 2.



(a) Score against hierarchy weight (Hierarchy 1)



(b) Score against hierarchy weight (Hierarchy 2)



(c) Score against hierarchy weight (Hierarchy 3)

Figure 3: Different hierarchy influences

8 Discussion and threats

8.1 Discussion

The resulting scores have shown that the third hierarchy when applied with a weight of 0.2, results in the LRAP and F1 scores that exceed that of the baseline by 3% and 2% respectively. It seems the results are highly sensitive to the configuration of the hierarchy. The initial hierarchy has small improvements over the baseline, the second hierarchy is more coarse-grained and seems to have worse results, and the third hierarchy has the best results. Therefore it seems plausible that the results could be improved even more with an even better hierarchy. It seems the ideal hierarchy has clusters that are not very big, and is structured as a deep tree, rather than a broad tree.

The survey results have indicated mixed results. On the negative side, in two cases, the correctly predicted tags by the baseline were lost after applying the hierarchy bias. However for five of the questions a correct tag was either added or given a better ranking. This seems to indicate that the net usefulness has increased over the baseline.

8.2 Threats to the Validity

Internal validity

The hierarchy is largely constructed manually and is therefore subject to subjective decision-making. In some cases this could lead to a tag being placed in the wrong order in the hierarchy, or even omitted entirely, causing the resulting scores to be lower than ideally possible. To deal with this threat, several distinct hierarchies are evaluated, although there is no way to say for certain that the best scoring hierarchy is optimal.

External validity

Because the hierarchy was constructed largely manually it can be challenging to reproduce the setup for a different dataset with different tags. Furthermore, because some tags used in the hierarchy are not available on Stack Overflow, these tags are not used in the experiment. In future work, a mapping could made from the hierarchy tags to the Stack Overflow tags, but since for most tags there does not exist a one-to-one mapping, this could alter the intended meaning behind the assigned tags, and so this was left out of this experiment. For example the ”kerbal-space-program” tag does not exist on Stack Overflow, the closest mapping would be to the tag ”games”, but this would lose most of its meaning in the process.

Construct validity

Standard theoretical metrics and techniques are used to evaluate the accuracy of the classifier, through the LRAP- and F1 scores. The user survey is conducted in an empirical fashion and is therefore susceptible to a degree of influence of domain-expertise and common practice that is not based on an established theoretical approach.

8.3 Responsible Research

This work evaluates the usefulness of the multi-label classifier by means of expert-user evaluations. Because this involves human participants, it is important to ensure that they are sufficiently informed on the ways in which their feedback

Table 2: Survey results

	#Rank improved	#Rank deteriorated	#Correct tags added	#Correct tags removed
Q1	0	0	1	1
Q2	0	0	0	0
Q3	2	0	0	0
Q4	2	0	1	0
Q5	0	0	0	0
Q6	0	0	0	0
Q7	1	1	1	0
Q8	1	0	2	0
Q9	1	0	0	0
Q10	0	0	1	0
Total	**7**	**1**	**6**	**1**

is processed and used, and to guarantee their anonymity. In the same sense, it is important to make sure all data obtained from the survey does not contain any information that can be traced back to any specific participant.

9 Conclusion and future work

This work sets out to answer how accurate and useful a Stack Overflow post tag predictor that makes use of a tag hierarchy is. This problem was then further subdivided into three research questions: How accurate is the tag predictor? How should the tag hierarchy be organized? And how useful is the tag predictor?

To answer these questions, three different hierarchies were constructed from the set of available tags, and a baseline was set-up using the state-of-the-art machine-learning classifier model DistilBERT. The results of the baseline's predictions were then processed to favour tags that are close together in the hierarchy.

The results show that using this method of post-processing, the resulting predictions does improve the LRAP- and F1-scores compared to the baseline, and thus a higher accuracy is obtained. They also show that the hierarchy that has the most amount of relatively-small clusters performs the best. Finally the survey shows that while the hierarchy-extended model does introduce new errors, overall it seems that the usefulness of the model has been improved.

This work could be extended and improved by refining the hierarchy even further and investigating exactly how to optimize the hierarchy to give the highest possible accuracy. If an ideal hierarchy can be determined, and even automatically constructed, it could prove very useful for all sorts of classification tasks.

10 Appendix

The full overview of the code used in this work, as well as the raw data and results, are openly available online. [5]

Table 3: Hierarchy 1 Scores

Hierarchy Weight	LRAP	Precision	Recall	F1
0.0 (baseline)	0.521	0.784	0.533	0.635
0.1	0.519	0.744	0.553	0.635
0.2	**0.541**	**0.722**	**0.596**	**0.654**
0.3	0.533	0.687	0.608	0.645
0.4	0.519	0.629	0.643	0.636
0.5	0.444	0.487	0.680	0.567

Table 4: Hierarchy 2 Scores

Hierarchy Weight	LRAP	Precision	Recall	F1
0.0 (baseline)	0.521	0.784	0.533	0.635
0.1	0.529	0.749	0.568	0.646
0.2	**0.539**	**0.717**	**0.599**	**0.653**
0.3	0.522	0.669	0.624	0.645
0.4	0.495	0.574	0.659	0.614
0.5	0.208	0.240	0.692	0.356

Table 5: Hierarchy 3 Scores

Hierarchy Weight	LRAP	Precision	Recall	F1
0.0 (baseline)	0.521	0.784	0.533	0.635
0.1	0.520	0.744	0.553	0.635
0.2	**0.543**	**0.724**	**0.597**	**0.654**
0.3	0.532	0.689	0.608	0.646
0.4	0.519	0.635	0.643	0.639
0.5	0.444	0.487	0.680	0.567

[5] https://github.com/PhilipMR/RPExperiments

References

[1] Mubashir Ali, Husnain Mushtaq, Muhammad B Rasheed, Anees Baqir, and Thamer Alquthami. Mining software architecture knowledge: Classifying stack overflow posts using machine learning. *Concurrency and Computation: Practice and Experience*, 33(16):e6277, 2021.

[2] Boli Chen, Xin Huang, Lin Xiao, Zixin Cai, and Liping Jing. Hyperbolic interaction model for hierarchical multi-label classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7496–7503, 2020.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[4] Scott A Golder and Bernardo A Huberman. Usage patterns of collaborative tagging systems. *Journal of information science*, 32(2):198–208, 2006.

[5] Maliheh Izadi, Abbas Heydarnoori, and Georgios Gousios. Topic recommendation for software repositories using multi-label classification algorithms. *Empirical Software Engineering*, 26(5):1–33, 2021.

[6] Ankur Joshi, Saket Kale, Satish Chandel, and D Kumar Pal. Likert scale: Explored and explained. *British journal of applied science & technology*, 7(4):396, 2015.

[7] Eray Mert Kavuk and Ayse Tosun. Predicting stack overflow question tags: a multi-class, multi-label classification. In *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, pages 489–493, 2020.

[8] Jin Liu, Pingyi Zhou, Zijiang Yang, Xiao Liu, and John Grundy. Fasttagrec: fast tag recommendation for software information sites. *Automated Software Engineering*, 25(4):675–701, 2018.

[9] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[10] Lin Xiao, Xin Huang, Boli Chen, and Liping Jing. Label-specific document representation for multi-label text classification. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 466–475, 2019.

[11] Bowen Xu, Thong Hoang, Abhishek Sharma, Chengran Yang, Xin Xia, and David Lo. Post2vec: Learning distributed representations of stack overflow posts. *IEEE Transactions on Software Engineering*, 2021.

[12] Pingyi Zhou, Jin Liu, Zijiang Yang, and Guangyou Zhou. Scalable tag recommendation for software information sites. In *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 272–282. IEEE, 2017.